



**HAL**  
open science

# Méthodes algébriques pour la décomposition de modèles comportementaux : Application à la détection et à la localisation de défaillances

Denis Berdjag

► **To cite this version:**

Denis Berdjag. Méthodes algébriques pour la décomposition de modèles comportementaux : Application à la détection et à la localisation de défaillances. Automatique / Robotique. Université des Sciences et Technologie de Lille - Lille I, 2007. Français. NNT : . tel-00385550

**HAL Id: tel-00385550**

**<https://theses.hal.science/tel-00385550>**

Submitted on 19 May 2009

**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.

## Thèse

présentée et soutenue publiquement le 26 octobre 2007  
pour l'obtention du

Doctorat de l'Université des Sciences et Technologies de Lille  
(Spécialité Automatique et Informatique Industrielle)

par  
Denis BERDJAG

# Méthodes algébriques pour la décomposition de modèles comportementaux : Application à la détection et à la localisation de défaillances

### Co-directeurs

Pr. Vincent COCQUEMPOT (LAGIS, Lille)  
Dr. Cyrille CHRISTOPHE (LAGIS, Lille)

### Rapporteurs

Pr. Jean-Pierre BARBOT (ECS, Cergy-Pontoise)  
Pr. Didier MAQUIN (CRAN, Nancy)

### Examineurs

Pr. Thierry-Marie GUERRA (LAMIH, Valenciennes)  
Pr. Dimitri LEFEBVRE (GREAH, Le Havre)



## Remerciements

*Cette thèse a été effectuée au sein de l'équipe Sûreté de Fonctionnement des Systèmes Dynamiques (SFSD) du Laboratoire d'Automatique, Génie Informatique et Signal de Lille (LAGIS, CNRS - UMR 8146) sous la direction de Messieurs Vincent Cocquempot et Cyrille Christophe, respectivement Professeur et Maître de Conférences à l'Université des Sciences et Technologies de Lille. Je les remercie pour leur conseils, pour leur suivi et pour leur aide tout au long de cette thèse.*

*Je remercie Messieurs Jean-Pierre Barbot, Professeur à l'ECS (Cergy-Pontoise) et Didier Maquin, Professeur au CRAN (Nancy) d'avoir accepté d'examiner ce travail et d'en être les rapporteurs, ainsi que pour leurs précieux conseils.*

*J'adresse également mes remerciements à Messieurs Thierry-Marie Guerra, Professeur au LAMIH (Valenciennes) et Dimitri Lefebvre, Professeur au GREAH (Le Havre) pour avoir accepté d'examiner ce travail et de participer au jury.*

*Je tiens également à remercier Monsieur Denis Pomorski, Professeur à l'USTL pour ses conseils pendant les premières années de ma thèse.*

*Je remercie tous les membres de l'équipe SFSD, permanents ou doctorants pour leur accueil et pour leur sympathie.*

*Enfin, j'adresse mes remerciements à tout ceux qui de près ou de loin ont contribué à cette thèse.*



# Table des matières

<b>Table des figures</b>	<b>v</b>
<b>1 Introduction générale</b>	<b>3</b>
<b>2 Surveillance et décomposition de modèles</b>	<b>7</b>
2.1 Introduction . . . . .	7
2.2 Généralités sur la surveillance à base de modèles . . . . .	8
2.2.1 Lexique et définitions . . . . .	8
2.2.2 Notions de surveillance . . . . .	9
2.2.3 Modélisation d'un système dynamique . . . . .	10
2.2.4 Notion de sous-modèle . . . . .	12
2.3 Surveillance à base de modèles temporels . . . . .	13
2.3.1 Introduction . . . . .	13
2.3.2 Modèle de bon fonctionnement . . . . .	14
2.3.3 Modèle de fonctionnement défaillant . . . . .	15
2.3.4 Utilisation du modèle pour la surveillance . . . . .	15
2.3.5 Génération de résidus pour la détection de défauts . . . . .	17
2.3.6 Résidus structurés pour la localisation de défauts . . . . .	19
2.4 Surveillance à base de modèles événementiels . . . . .	22
2.4.1 Introduction . . . . .	22
2.4.2 Représentation événementielle : principe et rappels . . . . .	23
2.4.3 Modèle de bon fonctionnement . . . . .	25
2.4.4 Modèle de fonctionnement défaillant . . . . .	25
2.4.5 Techniques de surveillance à base de modèles événementiels . .	28
2.4.6 Localisation d'événements inconnus par structuration des indi- cateurs de défaillances . . . . .	30
2.5 Conclusion . . . . .	35

2.5.1	Synthèse des problèmes à traiter . . . . .	35
2.5.2	Outils algébriques pour la surveillance à base de modèles temporels et événementiels . . . . .	35
2.5.3	La décomposition de modèles pour la surveillance . . . . .	36
<b>3</b>	<b>Rappels mathématiques</b>	<b>37</b>
3.1	Introduction . . . . .	37
3.2	Terminologie de la théorie des ensembles . . . . .	37
3.2.1	Ensembles . . . . .	37
3.2.2	Applications . . . . .	38
3.2.3	Relations . . . . .	39
3.2.4	Relations d'ordre . . . . .	40
3.2.5	Structures algébriques . . . . .	41
3.2.6	Treillis et ensembles ordonnés . . . . .	43
3.2.7	Partitions . . . . .	44
3.3	L'algèbre des paires . . . . .	48
3.3.1	Introduction . . . . .	48
3.3.2	Propriété de substitution . . . . .	48
3.3.3	Paires algébriques et opérateurs de l'algèbre des paires . . . . .	49
3.3.4	Structure algébrique et définitions de l'algèbre des paires . . . . .	51
3.3.5	Calcul des opérateurs $m$ et $M$ . . . . .	53
3.3.6	Application aux machines séquentielles . . . . .	55
3.3.7	Conclusion . . . . .	56
3.4	L'algèbre des fonctions . . . . .	57
3.4.1	Introduction . . . . .	57
3.4.2	Principe . . . . .	57
3.4.3	Définitions . . . . .	58
3.4.4	Paire algébrique en continu . . . . .	60
3.4.5	Parallèle entre les méthodes géométriques et les méthodes algébriques . . . . .	62
3.5	Conclusion . . . . .	64
<b>4</b>	<b>Décomposition de modèles comportementaux pour la surveillance</b>	<b>65</b>
4.1	Introduction . . . . .	65
4.2	Principe général de la décomposition . . . . .	66

4.2.1	Formulation du problème de décomposition . . . . .	66
4.2.2	Critère de couplage et de découplage pour la surveillance . . . . .	68
4.2.3	Condition d'existence d'un sous-modèle découplé . . . . .	69
4.2.4	Algorithme de décomposition avec critère de découplage . . . . .	72
4.2.5	Extension de l'algorithme de décomposition par injection de sorties . . . . .	74
4.3	Décomposition des modèles comportementaux temporels . . . . .	75
4.3.1	Introduction . . . . .	75
4.3.2	Objectif de la décomposition . . . . .	76
4.3.3	Méthodologie de décomposition d'un modèle temporel . . . . .	78
4.3.4	Exemple de décomposition de modèles temporels . . . . .	88
4.3.5	Aspects calculatoires . . . . .	92
4.3.6	Conclusion . . . . .	103
4.4	Décomposition des modèles comportementaux événementiels . . . . .	103
4.4.1	Introduction et objectifs . . . . .	103
4.4.2	Méthodologie de décomposition d'un modèle événementiel . . . . .	104
4.4.3	Conclusion . . . . .	120
4.5	Conclusion générale . . . . .	120
<b>5</b>	<b>Illustration sur le système à trois cuves</b>	<b>123</b>
5.1	Introduction . . . . .	123
5.2	Modélisation temporelle . . . . .	124
5.2.1	Modèle de bon fonctionnement . . . . .	124
5.2.2	Modèle de fonctionnement défaillant . . . . .	125
5.2.3	Décomposition en sous-modèles découplés . . . . .	126
5.2.4	Synthèse de sous-modèles découplés . . . . .	127
5.2.5	Génération de résidus . . . . .	130
5.2.6	Simulation . . . . .	131
5.3	Modélisation événementielle . . . . .	135
5.3.1	Description du modèle événementiel de bon fonctionnement . . . . .	135
5.3.2	Description du modèle de fonctionnement défaillant . . . . .	138
5.3.3	Décomposition du modèle événementiel . . . . .	141
5.3.4	Génération d'indicateurs de défaillances . . . . .	143
5.3.5	Simulations . . . . .	145
5.4	Conclusion . . . . .	154



<b>6 Conclusion générale</b>	<b>155</b>
6.1 Contribution . . . . .	155
6.2 Perspectives . . . . .	156
<b>A Rappels sur les propriétés des machines séquentielles</b>	<b>159</b>
A.1 Introduction . . . . .	159
A.2 Rappels sur les machines séquentielles . . . . .	159
<b>B Rappels sur la géométrie différentielle</b>	<b>163</b>
B.1 Introduction . . . . .	163
B.2 Variété et difféomorphisme . . . . .	163
B.3 Algèbre de Lie . . . . .	165
B.4 Distribution de champs de vecteurs et intégrabilité . . . . .	166
B.5 Équations aux dérivées partielles du premier ordre . . . . .	167
B.6 Formes différentielles et intégrabilité . . . . .	168
B.7 Conclusion . . . . .	170
<b>C Simulations</b>	<b>171</b>
C.1 Fuite dans la 1 <sup>re</sup> cuve $f_1$ . . . . .	171
C.2 Fuite dans la 2 <sup>e</sup> cuve $f_2$ . . . . .	172
C.3 Fuite dans la 3 <sup>e</sup> cuve $f_3$ . . . . .	173
C.4 Fuite dans la 2 <sup>e</sup> conduite $b_{32}$ . . . . .	174
C.5 Fuite dans la conduite de sortie $b_{20}$ . . . . .	175
<b>Bibliographie</b>	<b>177</b>

# Table des figures

2.1	Système élémentaire . . . . .	8
2.2	Modèle hybride . . . . .	12
2.3	Illustration de la surveillance à base de modèles . . . . .	16
2.4	Banc de générateurs de résidus robustes . . . . .	20
2.5	Utilisation d'un banc de sous-systèmes découplés pour la génération de résidus structurés robustes . . . . .	21
2.6	Graphe d'état d'une machine de Moore . . . . .	24
2.7	Graphe d'état de $\Sigma$ . . . . .	24
2.8	Représentation de la machine séquentielle $\Sigma_1$ . . . . .	26
2.9	Représentation de la machine séquentielle $\Sigma_2$ . . . . .	27
2.10	Représentation de la machine séquentielle $\Sigma_3$ . . . . .	27
2.11	Méthode de surveillance des systèmes à dynamique événementielle . . . . .	29
2.12	Diagramme représentant une machine séquentielle partielle . . . . .	31
2.13	Banc de machines séquentielles partielles . . . . .	32
2.14	Graphe d'évolution de la machine séquentielle $\Sigma$ . . . . .	33
2.15	Représentation de la machine séquentielle partielle $\Sigma'$ . . . . .	34
2.16	Schéma représentant la machine séquentielle partielle $\Sigma''$ . . . . .	34
3.1	Treillis formé par les sous-ensembles de $\{1, 2, 3\}$ associés à « $\subseteq$ » . . . . .	44
3.2	Graphe représentant les partitions $\pi_1$ (gauche) et $\pi_2$ (droite) . . . . .	45
3.3	Diagramme représentant la relation d'ordre $\pi_1 \leq \pi_2$ . . . . .	47
4.1	Diagramme représentant la décomposition d'un modèle $\Sigma$ . . . . .	67
4.2	Décomposition parallèle du modèle $\Sigma$ . . . . .	77
4.3	Graphe représentant la machine séquentielle $\Sigma$ . . . . .	108
4.4	Graphe représentant la machine séquentielle partielle $\Sigma_\pi$ . . . . .	108
4.5	Graphe représentant $\Sigma_{\pi'}$ . . . . .	109
4.6	Graphe représentant la machine séquentielle partielle $\Sigma_{\pi'}$ . . . . .	111
4.7	Graphe illustrant la machine séquentielle $\Sigma$ . . . . .	112
4.8	Représentation graphique de modèle $\Sigma$ . . . . .	116
4.9	Graphe d'état de la machine séquentielle partielle $\Sigma_{f_1}$ . . . . .	118
4.10	Graphe d'état de la machine séquentielle partielle $\Sigma_{f_2}$ . . . . .	119
5.1	Représentation du processus à trois cuves . . . . .	123
5.2	Diagramme fonctionnel du processus 3 cuves . . . . .	124

5.4	Résidus dans le cas normal . . . . .	132
5.3	Schéma de simulation Simulink . . . . .	133
5.5	Résidus dans le cas d'une défaillance $b_{13}$ permanente . . . . .	134
5.6	Résidus dans le cas d'une défaillance $b_{13}$ temporaire . . . . .	134
5.7	Représentation du modèle de fonctionnement normal du processus 3 cuves . . . . .	137
5.8	Modèle événementiel d'une vanne (approche <i>transitions</i> ) . . . . .	139
5.9	Modèle événementiel d'une vanne (approche <i>état</i> ) . . . . .	139
5.10	Représentation du modèle de fonctionnement défaillant du processus à 3 cuves . . . . .	140
5.11	Représentation graphique de la machine séquentielle partielle $\Sigma_{f_{V_{13}}}$ et de $\Sigma_{f_{V_{32}}}$ . . . . .	144
5.12	Séquences d'événements et défaillances . . . . .	146
5.13	Graphes représentant les réponses du système complet $\Sigma$ et des machines partielles $\Sigma_1$ et $\Sigma_2$ . . . . .	148
5.14	Graphe représentant l'évolution des indicateurs de défaillances en fonction de l'évolution de $\Sigma$ . . . . .	148
5.15	Blocs représentant les machines séquentielles partielles $\Sigma_{V_{13}}$ et $\Sigma_{V_{32}}$ . . . . .	150
5.16	Évolution des niveaux de liquide dans les trois cuves . . . . .	150
5.17	Entrées et défaillances du modèle continu . . . . .	150
5.18	Évolution des débits de liquide dans les conduites $c_{13}$ et $c_{32}$ . . . . .	151
5.19	Événements d'entrée du modèle événementiel . . . . .	151
5.20	Évolution discrétisée du système et du modèle . . . . .	152
5.21	Évolution des machines séquentielles partielles découplées . . . . .	152
5.22	Évolution des indicateurs de défaillances . . . . .	153
5.23	Résidus continus . . . . .	153
A.1	Représentation schématique des machines séquentielles isomorphes $\Sigma'$ et $\Sigma$ . . . . .	160
A.2	Représentation schématique $\Sigma$ et de son image homomorphe $\Sigma'$ . . . . .	160
A.3	Schéma représentant les machines équivalentes $\Sigma_1$ , $\Sigma_2$ et $\Sigma'$ . . . . .	161
C.1	Résidus dans le cas d'une défaillance $f_1$ permanente . . . . .	171
C.2	Résidus dans le cas d'une défaillance $f_2$ temporaire . . . . .	172
C.3	Résidus dans le cas d'une défaillance $f_2$ permanente . . . . .	172
C.4	Résidus dans le cas d'une défaillance $f_2$ temporaire . . . . .	173
C.5	Résidus dans le cas d'une défaillance $f_3$ permanente . . . . .	173
C.6	Résidus dans le cas d'une défaillance $f_3$ temporaire . . . . .	174
C.7	Résidus dans le cas d'une défaillance $b_{32}$ permanente . . . . .	174
C.8	Résidus dans le cas d'une défaillance $b_{32}$ temporaire . . . . .	175
C.9	Résidus dans le cas d'une défaillance $b_{20}$ permanente . . . . .	175
C.10	Résidus dans le cas d'une défaillance $b_{20}$ temporaire . . . . .	176

## Résumé

Cette thèse traite de la détection et de la localisation des défaillances à base de modèles comportementaux. L'approche suggérée consiste à extraire du modèle comportemental un sous-modèle découplé par rapport aux perturbations et par rapport à un sous-ensemble de défaillances et couplé par rapport au sous-ensemble complémentaire de défaillances. À l'aide de ce sous-modèle, des indicateurs de défaillances structurés sont générés. Ces indicateurs possèdent intrinsèquement des propriétés de robustesse aux perturbations et à certaines défaillances et sont sensibles au reste des défaillances. Cette approche repose donc sur la décomposition du modèle comportemental du système en sous-modèles.

Une méthodologie de décomposition est présentée pour traiter les modèles continus et les modèles événementiels de la même manière. Cette méthodologie de décomposition est formulée en utilisant des outils algébriques connus sous le nom d'*algèbre des paires* et d'*algèbre des fonctions*. Les deux algèbres sont présentées et leurs opérateurs sont détaillés et illustrés. Il est montré que la procédure de décomposition décrite pour les modèles continus en utilisant l'algèbre des fonctions est similaire à celle décrite pour les modèles événementiels en utilisant l'algèbre des paires. Une procédure de décomposition générale est ensuite proposée pour tous les modèles comportementaux déterministes.

Les résultats théoriques sont illustrés par la décomposition des modèles comportementaux (continu et événementiel) du système à trois cuves, et par la synthèse d'un module de détection et de localisation de défaillances pour chaque cas traité.

## Abstract

This thesis concerns fault detection and isolation using behavioral models. The suggested approach consists in the determination of a sub-system which is decoupled from the perturbation and from a subset of faults and sensitive to the remaining faults. Using this sub-system, structured fault indicators are generated. These indicators are intrinsically robust to the perturbations and to some faults and sensitive to the remaining faults. Thus, this approach is based on the decomposition of the behavioral model of the process to monitor into sub-systems.

A decomposition methodology is proposed to handle continuous and discrete models in a similar way. Algebraic tools, known as *pair algebra* and *algebra of functions*, are used to formulate the decomposition methodology. Both algebras are presented, and their operators detailed and illustrated. It is shown that the decomposition procedure for the continuous case based on algebra of functions is similar to the discrete case based on pair algebra. Finally, a decomposition methodology for all types of deterministic behavioral models is proposed.

The theoretical results are illustrated by the decomposition of the behavioral models (continuous and discrete) of the three tank process, and the synthesis of fault detection and isolation module for both cases.

# Chapitre 1

## Introduction générale

La surveillance des systèmes automatiques est devenue, depuis les années soixante dix, une thématique à part entière de la recherche scientifique. Les progrès technologiques ont entraîné une complexité croissante des systèmes, et les problèmes de sécurité, de fiabilité et de sûreté ont pris de l'importance. Ceci a créé le besoin de développer des méthodes fiables et performantes de surveillance.

La surveillance consiste en la détection précoce des défauts dont l'apparition perturbe le fonctionnement normal d'un système. Une fois que l'anomalie est détectée, le composant en défaut est localisé et la gravité du défaut est estimée.

Nous nous intéressons dans cette thèse aux méthodes de surveillance à base de modèles. Le modèle représente une abstraction mathématique du comportement du système physique. Les modèles peuvent être représentés de plusieurs manières suivant le degré d'abstraction retenu : équations différentielles ou aux différences, linéaires ou non linéaires, graphes d'évolution (réseaux de Petri, automates à états finis ...), relations qualitatives, lois empiriques. Pour chaque cas, des outils de traitement adéquats sont utilisés pour développer des algorithmes de surveillance efficaces. Ces outils ont été développés par des communautés différentes (systèmes continus, systèmes événementiels) en utilisant une formalisation du problème spécifique.

Quelle que soit la représentation choisie, un modèle permet d'estimer la sortie du système, connaissant son évolution passée et les sollicitations actuelles. La surveillance consiste en premier lieu à vérifier la cohérence des sorties estimées avec les sorties connues (mesurées). Cette cohérence est quantifiée par un indicateur. La valeur de l'indicateur dépend *a priori* des erreurs de modélisation, des sollicitations inconnues et des défauts affectant les caractéristiques nominales du système. Pour permettre une détection efficace, l'indicateur doit être sensible uniquement aux défauts et robuste aux autres influences. La détection robuste d'un défaut nécessite la robustesse de l'estimation des sorties.

Si l'occurrence de défauts multiples est considérée, l'utilisation de plusieurs indicateurs sensibles à des sous-ensembles de défauts différents permet, sous certaines conditions, de localiser l'élément défaillant. Le problème de détection et de localisation de défauts est alors ramené à un problème général de sensibilité et de robustesse des estimations fournies par le modèle.

Les méthodes classiques de surveillance se focalisent sur les techniques de géné-

ration des indicateurs ce qui rend ces méthodes tributaires des modèles utilisés. En fonction de la dynamique décrite par le modèle, différentes techniques doivent être employées pour la détection des défauts. Les approches employées pour surveiller des dynamiques continues dans le temps diffèrent considérablement des méthodes utilisées pour la surveillance de dynamiques événementielles discrètes. Par ailleurs, la génération d'indicateurs de défauts est traitée comme un problème multi-contraintes difficile à résoudre dans les cas complexes.

Nous proposons dans cette thèse une résolution du problème de génération d'indicateurs robustes en deux étapes. La méthodologie est générale et peut s'appliquer à des systèmes à dynamique continue ou événementielle. Les propriétés de robustesse et de sensibilité par rapport aux erreurs de modélisation, aux sollicitations et aux défauts peuvent être ramenées à des propriétés de couplage et de découplage au sein même du modèle. La première étape consiste à déterminer *les parties* du modèle qui possèdent les propriétés de couplage/découplage souhaitées. Les *sous-modèles* obtenus permettent une génération facilitée d'indicateurs de défauts. Les indicateurs possèdent intrinsèquement les propriétés de robustesse ou de sensibilité souhaitées *héritées* des sous-modèles.

Le deuxième point abordé est l'utilisation d'outils mathématiques particuliers, appelés *algèbre des paires* et *algèbre des fonctions*, pour formuler la méthode de décomposition. L'algèbre des paires est utilisée pour la décomposition de modèles événementiels, et l'algèbre des fonctions est utilisée pour la décomposition de modèles temporels. Les deux algèbres sont présentées, et leur utilisation pour la décomposition est développée, notamment sur des aspects calculatoires. Des analogies avec des outils de découplage classiques sont montrées pour illustration.

## Plan du mémoire

La suite de ce mémoire est structurée autour des chapitres suivants :

### Chapitre 2

Le chapitre présente des rappels fondamentaux sur la représentation des systèmes et des défauts par des modèles comportementaux et sur les notions de la surveillance à base de modèles. Deux types de modèles comportementaux sont étudiés :

- Le modèle comportemental temporel qui représente l'évolution des systèmes à événements continus.
- Le modèle comportemental événementiel qui représente l'évolution des systèmes à événements discrets.

Une section entière est consacrée à chaque type de modèle, comportant :

- des rappels sur la modélisation du processus en fonctionnement normal ou en fonctionnement défaillant,
- un rappel des principales notions de surveillance à base de modèles,
- un état de l'art sur les techniques les plus courantes de la surveillance à base de modèle.

Dans la dernière partie de chaque section, une méthode de détection et de localisation de défaillances basée sur la génération d'indicateurs de défauts structurés est proposée. Un indicateur de défauts structuré est un signal qui réagit à certains défauts en restant robuste aux autres. Le problème classique de détection et de localisation de défauts, bien connu dans le cadre des systèmes à états continus est formulé comme un problème de décomposition avec des critères de découplage par rapport à un sous-ensemble d'entrées. Les sous-modèles obtenus permettent ensuite de générer des indicateurs de défauts structurés. La formulation du problème est présentée dans le cadre des modèles temporels et étendue au cadre des modèles événementiels.

### Chapitre 3

Ce chapitre présente les outils mathématiques utilisés. Des rappels généraux sur la théorie des ensembles et sur les structures d'*algèbre* et de *treillis* sont donnés. La notion de *partition* sur un ensemble est rappelée et les opérations de manipulation de partitions sont détaillées. Suit une présentation de l'*algèbre des paires*, qui est utilisée pour le traitement des systèmes à dynamique événementielle. Des notions clés liées aux partitions sont introduites, telles la propriété de substitution et la paire algébrique. La propriété de substitution est une relation entre partitions qui permet de décrire le modèle sous la forme d'un treillis de partitions. Ce treillis ordonne les partitions suivant une hiérarchie en fonction d'une relation d'ordre, ce qui permet de choisir une partition de décomposition. L'*algèbre des paires* est ensuite étendue aux ensembles infinis sous le nom d'*algèbre des fonctions*. Cette extension est utilisée pour le traitement des systèmes à dynamique temporelle. Une comparaison avec des outils mathématiques similaires est faite, en montrant des analogies entre l'*algèbre des fonctions* et l'*algèbre de Lie*.

### Chapitre 4

Le quatrième chapitre présente la méthodologie de décomposition en sous-modèles découplés. Le principe de l'approche est présenté tout d'abord d'un point de vue général. Les étapes qui mènent à l'obtention d'un sous-modèle découplé sont décrites sous la forme d'un algorithme. Les sorties du sous-modèle possèdent des propriétés de robustesse par rapport aux entrées qui sont découplées. Cette propriété permet d'appliquer la décomposition en sous-systèmes découplés pour synthétiser des générateurs d'indicateurs de défauts structurés. Le chapitre est ensuite divisé en deux sous-parties :

1. La première section est consacrée au détail de la décomposition des systèmes à dynamique temporelle et à sa réalisation en utilisant l'*algèbre des fonctions*. Des illustrations sont proposées pour des systèmes représentés par des équations différentielles linéaires et non linéaires. Des améliorations de l'approche existante [Zhi97] sont proposées, notamment sur l'initialisation de l'algorithme et sur les aspects calculatoires.
2. La seconde section traite de la décomposition des systèmes à dynamique événementielle. Le processus est décrit sous la forme d'une machine séquentielle et les



étapes de la décomposition sont réécrites pour en tenir compte. L'algorithme de décomposition est alors réalisé en utilisant l'algèbre des paires.

## Chapitre 5

Le cinquième chapitre illustre les techniques développées tout au long de ce mémoire, en réalisant la détection de défaillances sur un système à trois cuves. Deux représentations sont utilisées, l'une temporelle et l'autre événementielle.

1. *Dans le cadre temporel* : le système est représenté par un ensemble d'équations différentielles qui présente des non-linéarités non différentiables. La méthodologie de décomposition est réalisée en utilisant les opérateurs de l'algèbre des fonctions. Les sous-modèles obtenus permettent de synthétiser des générateurs de résidus structurés en utilisant des observateurs discontinus. Des simulations sont réalisées à l'aide du logiciel MATLAB/SIMULINK©.
2. *Dans le cadre événementiel* : une partie de l'évolution du système est représentée sous la forme d'une machine séquentielle à seize états. La méthodologie de décomposition est réalisée en utilisant les opérateurs de l'algèbre des paires. Les sous-modèles générés permettent d'obtenir des séquences indicatrices de défaillances structurées. Des simulations sont réalisées pour deux cas :
  - Les sorties du système à surveiller sont obtenues directement à partir d'un modèle événementiel. L'objectif est de valider l'approche de surveillance *théoriquement*.
  - Les sorties du système à surveiller sont obtenues par discrétisation des sorties d'un modèle temporel. L'objectif est de valider l'approche de surveillance dans un cas plus pratique sur un modèle proche du système réel.

Les simulations (et les calculs des opérateurs) sont réalisées à l'aide du logiciel de calcul symbolique MAXIMA.

**Remarque 1.0.1** *Il est recommandé au lecteur qui n'est pas habitué aux techniques algébriques de lire le chapitre 5 avant d'aborder les chapitres 3 et 4.*

# Chapitre 2

## Surveillance et décomposition de modèles

### 2.1 Introduction

Ce chapitre rappelle les notions principales liées à la surveillance (détection, localisation, identification de défauts) de systèmes automatisés. L'objectif ici n'est pas de réaliser un exposé exhaustif mais bien de donner les éléments essentiels qui permettront de situer et de comprendre toute la problématique traitée dans cette thèse ainsi que les solutions apportées. Nous nous intéressons aux méthodes de surveillance utilisant un modèle comportemental de l'installation considérée. D'un point de vue très général, un tel modèle est constitué d'un ensemble de contraintes qui *définissent* ou *contraignent* les évolutions des variables internes. Quelle que soit la technique de surveillance utilisée, le principe est toujours le même : comparer le comportement du système réel tel qu'il est vu au travers de son instrumentation avec le comportement théorique donné par son modèle. Les automaticiens manipulent trois types de modèles comportementaux : les modèles que l'on qualifiera de *temporels*, pour lesquels les évolutions des variables dépendent explicitement du temps (ces modèles peuvent être à temps continu ou à temps discret), les modèles *événementiels* lorsque les variables réagissent exclusivement à des événements internes ou externes, enfin les modèles dits *hybrides* qui combinent les deux évolutions. Suivant le type de modèle considéré, les techniques employées sont différentes.

- Les méthodes utilisant un modèle temporel reposent sur des indicateurs de défauts (on parle aussi de *résidus*). Ceux-ci sont générés par trois types de méthodes : les méthodes basées sur les observateurs ou filtres, les méthodes à base de relations de redondance analytique et les techniques utilisant des algorithmes d'estimation paramétrique (techniques d'identification). Afin de localiser précisément les défauts, plusieurs résidus sont utilisés. Chaque résidu doit avoir des propriétés spécifiques de sensibilité et de robustesse vis à vis des perturbations et d'un sous-ensemble donné de défauts.
- Les méthodes utilisant un modèle événementiel ont pour principe général la simulation parallèle. Un *diagnostiqueur* est associé avec pour objectif de détermi-

ner si le système réel est dans un état normal ou dans un état anormal. Dans le deuxième cas, le diagnostiqueur recherchera la(les) cause(s) *première(s)* du dysfonctionnement. Dans le cas de systèmes complexes, lorsque la surveillance doit être réalisée en-ligne, il est impossible de simuler tous les scénarii et d'explorer systématiquement toutes les combinaisons ayant amené le système dans un état donné. Ceci serait d'ailleurs inutile. En effet, certaines situations sont physiquement impossibles, d'autres n'apportent aucune information sur le défaut, enfin d'autres sont imprécises du fait de l'existence de perturbations. Seuls les scénarii ayant un sens et une utilité doivent donc être considérés. On retrouve ainsi pour la procédure utilisée, comme précédemment pour les résidus, une certaine propriété de robustesse et de sensibilité aux perturbations et aux défauts.

Les démarches classiques de surveillance partent du modèle global de bon ou de mauvais fonctionnement et cherchent à générer des résidus (dans le cas continu) ou des simulateurs/diagnostiqueurs possédant les propriétés de robustesse et de sensibilité souhaitées. Une autre démarche est présentée dans cette thèse. Nous proposons de travailler dans un premier temps sur les modèles avant de s'intéresser à la technique de surveillance à proprement parler. Ainsi, le modèle global peut être décomposé en *sous-modèles* possédant chacun des propriétés de couplage/découplage par rapport aux perturbations et/ou aux défauts. Nous explicitons à la fin de ce chapitre cette démarche et dégageons les avantages d'une telle approche.

## 2.2 Généralités sur la surveillance à base de modèles

### 2.2.1 Lexique et définitions

Un processus physique est un assemblage de composants en interaction. Les composants ont été choisis pour remplir une (parfois plusieurs) fonction(s) permettant au système complet d'assurer ses missions [BKLS03]. Les composants sont classés en trois catégories : les actionneurs qui permettent d'agir sur le dispositif, les composants du dispositif en tant que tel et les capteurs qui fournissent une image de certaines quantités produites par le dispositif. Le schéma classique est donc le suivant :

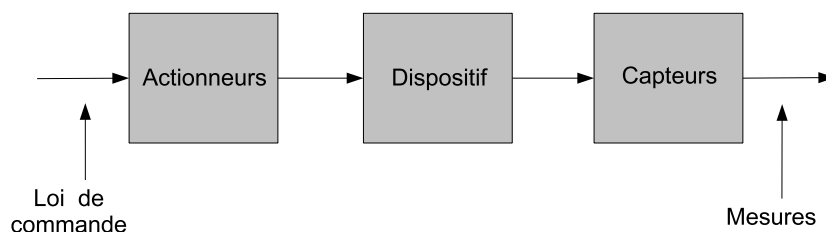


FIG. 2.1 – Système élémentaire

Chaque composant réagit à des sollicitations connues telles les commandes manuelles ou automatiques, les signaux ou données d'entrées et/ou à des sollicitations inconnues appelées entrées inconnues ou perturbations.

Des dysfonctionnements sont susceptibles de perturber le comportement *normal* du composant. Ces dysfonctionnements sont soit des défauts, soit des pannes.

**Défaut** On appelle *défaut* toute modification des caractéristiques physiques de l'élément concerné qui entraîne un comportement dégradé, non nominal, de l'élément.

**Panne** On appelle *panne* toute modification des caractéristiques physiques de l'élément concerné qui entraîne son incapacité totale à accomplir sa fonction première. Une panne est donc en quelque sorte un défaut majeur. Dans la suite du mémoire, seul le terme *défaut*, plus général, sera utilisé.

La distinction entre *défaut* et *perturbation* n'est pas toujours très claire. Une perturbation traduit l'influence de l'environnement dans lequel évolue un élément. Ceci entraîne une modification non souhaitée du comportement de l'élément *sans* qu'il y ait modification de ses caractéristiques physiques intrinsèques. La difficulté de la surveillance est de pouvoir distinguer les perturbations des défauts réels.

Les défauts sont classés en fonction de leur persistance. Ils sont de deux types :

**intermittents** s'ils sont présents uniquement pendant un laps de temps déterminé. Le dispositif touché récupère son mode de fonctionnement normal sans intervention extérieure. Ces défauts sont très gênants et très difficiles à détecter et à diagnostiquer.

**permanents** s'ils persistent au cours du temps jusqu'à une intervention de maintenance.

### 2.2.2 Notions de surveillance

La surveillance d'un processus consiste en la détection, la localisation et l'identification des défauts du système à un stade précoce. La surveillance a pour objectif de donner une image "réelle" de l'état des composants du système. Elle permet de faire intervenir la maintenance de manière appropriée ou de modifier automatiquement la commande pour permettre un fonctionnement proche du fonctionnement nominal et en tout cas garantissant certaines propriétés essentielles. On parle alors de commande active tolérante aux défauts.

Un algorithme de surveillance [BKLS03] comporte 3 étapes :

**La détection** L'objectif est de détecter les défauts le plus précocement possible et sans erreur (fausse alarme ou non détection). Un écart entre le comportement observé et le comportement normal de référence donné par un modèle (qui peut être analytique ou numérique) peut être dû à un défaut mais aussi à des perturbations ou des erreurs de modélisation. Une décision doit être prise pour déclencher une alarme si le dysfonctionnement est réellement causé par un défaut et ne pas la déclencher s'il s'agit d'une perturbation.

**La localisation** Il s'agit de dire quel élément physique du système est en défaut. Parfois la localisation très précise est impossible du fait des interactions entre composants et de l'instrumentation limitée. Le résultat se limitera donc à isoler un

sous-système comportant un ensemble de composants susceptibles d'être en défaut.

**L'identification** L'objectif est de caractériser plus finement le défaut. D'un point de vue physique, il s'agit de déterminer la nature du défaut (nombre de spires court-circuitées dans une machine électrique, pourcentage d'ouverture d'une vanne lorsqu'elle reste partiellement ouverte, ...). Étant donné que les paramètres du modèle sont liés aux éléments physiques, l'identification du défaut consiste souvent dans l'identification des paramètres du modèle.

Dans la littérature, le terme *diagnostic* est aussi souvent utilisé. Il correspond à la caractérisation du défaut lorsque celui-ci a été détecté. Le diagnostic peut donc être vu comme la réunion des fonctions de localisation et d'identification du défaut.

Afin de réaliser les fonctions précédentes, il est nécessaire de disposer de comportements de référence décrivant les fonctionnements *normaux* et *défaillants* du processus. Ces comportements peuvent se présenter sous différentes formes et sont groupés en deux catégories : les relations qui décrivent l'évolution du processus et les données relevées durant des tests de *scenarii* de fonctionnement possibles. Les techniques de surveillance existantes sont classées traditionnellement en deux familles :

**Techniques de surveillance dites sans modèle** Ces techniques se basent sur l'analyse des données issues des capteurs. Un apprentissage des différents *scenarii* de fonctionnement possibles est effectué et une classification des données obtenues en sortie permet de statuer sur l'état du processus. L'apprentissage peut être fait en ligne ou bien pendant une phase de test préalable. Selon le degré de sophistication voulu, les classes décrivant les *scenarii* peuvent être fixes ou bien varier en fonction de l'apprentissage en ligne.

**Techniques de surveillance avec modèles** Le fonctionnement normal du système est décrit par un modèle comportemental constitué de relations de contraintes entre des variables choisies du système. Les trajectoires (ou évolutions) des variables, estimées à l'aide de ce modèle, sont comparées avec les données prélevées sur le système réel et provenant des capteurs. Un écart entre ces trajectoires provient soit d'erreurs de modélisation, soit de perturbations (entrées inconnues) non prises en compte dans le modèle, soit enfin de défauts de composants. Afin de détecter, localiser voire identifier un défaut, il est nécessaire de différencier les causes des dysfonctionnements. Ainsi, le modèle est complété afin de tenir compte des perturbations, incertitudes et défauts.

Dans la suite de ce mémoire nous nous intéressons uniquement aux techniques de surveillance utilisant un modèle comportemental du système considéré.

### 2.2.3 Modélisation d'un système dynamique

Un modèle est une représentation abstraite des phénomènes physiques. Cette représentation d'un système dynamique permet de simuler les comportements à partir de conditions initiales données et en fonction de sollicitations (entrées) connues. Un modèle comportemental, ou analytique, regroupe les variables de description du processus ainsi que les relations qui les lient.

**Définition 2.1** *Un modèle comportemental (analytique) est défini par le quintuplet*

$$\Sigma(\mathcal{X}, \mathcal{W}, \mathcal{Y}, \mathcal{F}, \mathcal{H}) \quad (2.1)$$

où  $\mathcal{W}$  représente l'ensemble des excitations,  $\mathcal{Y}$  représente l'ensemble des sorties et  $\mathcal{X}$  représente l'ensemble des états internes.  $\mathcal{F}$  et  $\mathcal{H}$  sont des relations décrites respectivement par

$$\mathcal{F} : \mathcal{W} \times \mathcal{X} \longrightarrow \mathcal{X} \quad (2.2)$$

$$\mathcal{H} : \mathcal{W} \times \mathcal{X} \longrightarrow \mathcal{Y} \quad (2.3)$$

Les relations qui sont décrites par l'application  $\mathcal{F}$  caractérisent les dynamiques contenues dans le processus représenté. Deux types de dynamiques peuvent être distingués, suivant le degré d'abstraction retenu pour l'élaboration du modèle :

**La dynamique temporelle** est caractérisée par des variables de description qui ont une évolution continue dans le temps. Les contraintes sont exprimées par des relations mathématiques, du type *équations différentielles* pour le cas des systèmes temporels continus ou du type *équations aux différences* (récurrences) pour le cas de systèmes temporels échantillonnés (ou discrétisés). Les variables de description peuvent prendre un nombre infini de valeurs.

**La dynamique événementielle :** est caractérisée par des variables de description qui subissent des variations brusques. Ces évolutions ne dépendent pas directement du temps, mais de l'occurrence d'événements particuliers. Les variables de description ne peuvent prendre qu'un nombre fini de valeurs.

Les modèles de comportement temporels et événementiels peuvent maintenant être caractérisés de manière plus précise.

**Définition 2.2** *Un modèle comportemental  $\Sigma(\mathcal{X}, \mathcal{W}, \mathcal{Y}, \mathcal{F}, \mathcal{H})$  (2.1) est dit temporel et est noté*

$$\Sigma(X, W, Y, f, h)$$

si et seulement si ses variables dépendent explicitement du temps et sont définies sur des ensembles infinis. On a alors :

$$\begin{aligned} X &\subseteq \mathbb{R}^n & f &: \mathbb{R}^n \times \mathbb{R}^l \longrightarrow \mathbb{R}^n \\ W &\subseteq \mathbb{R}^l & h &: \mathbb{R}^n \times \mathbb{R}^l \longrightarrow \mathbb{R}^m \\ Y &\subseteq \mathbb{R}^m & & \end{aligned} \quad (2.4)$$

La fonction  $f$  est appelée fonction d'état et décrit les évolutions des variables d'état en fonction des entrées et des paramètres. C'est une fonction qui peut être linéaire ou non linéaire. La nature de cette fonction caractérise le type de modèle temporel, il est alors soit linéaire, soit non linéaire. La fonction  $h$  est appelée fonction de sortie ; elle lie les états et les entrées du système aux mesures relevées par les capteurs installés.

**Définition 2.3** Un modèle comportemental  $\Sigma(\mathcal{X}, \mathcal{W}, \mathcal{Y}, \mathcal{F}, \mathcal{H})$  (2.1) est dit événementiel et est noté

$$\Sigma(S, I, O, \delta, \lambda)$$

si et seulement si les variables qui le composent sont de nature événementielle et sont définies sur des ensembles finis. On a alors

$$\begin{aligned} S &= \{s_1, \dots, s_{n'}\} & \delta : S \times I &\longrightarrow S \\ I &= \{i_1, \dots, i_{l'}\} & \lambda : S \times I &\longrightarrow O \\ O &= \{o_1, \dots, o_{m'}\} \end{aligned} \quad (2.5)$$

où les  $s_i, i_i, o_i$  sont les éléments respectifs des ensembles  $S, I$  et  $O$  aux cardinalités  $n', l'$  et  $m'$ .

Un modèle qui contient les deux dynamiques, temporelle et événementielle, est appelé modèle *hybride* [Pet99]. L'évolution d'un modèle hybride est représentée par deux modèles (figure 2.2) :

- un modèle événementiel décrit par  $(S, I, O, \delta, \lambda)$ ,
- un modèle temporel décrit par  $(\check{X}, \check{W}, \check{Y}, \check{f}, \check{h})$ .

Chaque état  $s \in S$  du modèle événementiel correspond à un mode de fonctionnement du modèle hybride et définit un sous-ensemble d'états atteignables du modèle temporel  $X_i \subseteq \check{X}$ . Les changements de modes sont déclenchés par des événements commandés ou par le changement d'état de contraintes inégalité  $g_i(x_i) \leq 0$ . Le binôme  $(s, x_i)$  à un instant donné représente l'état complet du modèle hybride.

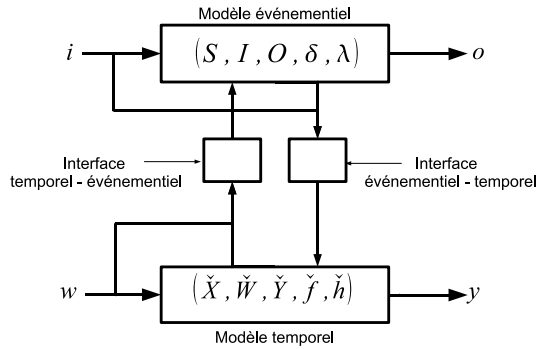


FIG. 2.2 – Modèle hybride

## 2.2.4 Notion de sous-modèle

Un modèle  $\Sigma$  représente une abstraction mathématique du comportement d'un système physique. Il permet, à partir d'un ensemble de sollicitations externes  $\mathcal{W}$ , d'obtenir, via un ensemble de variables internes  $\mathcal{X}$ , une image des sorties instrumentées du système  $\mathcal{Y}$ . Pour diverses raisons, il est parfois intéressant de décomposer le modèle complet du système pour déterminer une partie qui ne dépend que d'un sous-ensemble de sollicitations externes  $\mathcal{W}' \subseteq \mathcal{W}$  et dont les sorties  $\mathcal{Y}'$  peuvent être exprimées en fonction des éléments de l'ensemble de sorties  $\mathcal{Y}$ .

**Définition 2.4** Soit un modèle  $\Sigma : (\mathcal{X}, \mathcal{W}, \mathcal{Y}, \mathcal{F}, \mathcal{H})$ . Le modèle  $\Sigma' : (\mathcal{X}', \mathcal{W}', \mathcal{Y}', \mathcal{F}', \mathcal{H}')$  est un sous-modèle de  $\Sigma$  si pour une relation  $\Psi_{\mathcal{W}}$  définie par :

$$\begin{aligned} \Psi_{\mathcal{W}} : \mathcal{W} &\rightarrow \mathcal{W}' \\ w &\mapsto w' = \Psi_{\mathcal{W}}(w) \end{aligned}$$

il existe une relation  $\Psi_{\mathcal{X}}$  définie par

$$\begin{aligned} \Psi_{\mathcal{X}} : \mathcal{X} &\rightarrow \mathcal{X}' \\ x &\mapsto x' = \Psi_{\mathcal{X}}(x) \end{aligned}$$

qui satisfait la relation suivante

$$\forall w \in \mathcal{W} : x' = \Psi_{\mathcal{X}}(x) \Leftrightarrow \mathcal{F}'(x', \Psi_{\mathcal{W}}(w)) = \Psi_{\mathcal{X}}(\mathcal{F}(x, w))$$

et une relation  $\Psi_{\mathcal{Y}}$  définie par

$$\begin{aligned} \Psi_{\mathcal{Y}} : \mathcal{Y} &\rightarrow \mathcal{Y}' \\ y &\mapsto y' = \Psi_{\mathcal{Y}}(y) \end{aligned}$$

qui satisfait

$$\forall w \in \mathcal{W} \forall x \in \mathcal{X} \forall x' \in \mathcal{X}' : x' = \Psi_{\mathcal{X}}(x) \Leftrightarrow \mathcal{H}'(x', \Psi_{\mathcal{W}}(w)) = \Psi_{\mathcal{Y}}(\mathcal{H}(x, w))$$

Un sous-modèle permet de simuler une partie du système. Il évolue suivant les mêmes trajectoires que la partie du modèle complet dépendante de  $\mathcal{W}'$  et qui est identifiée par le triplet de sous-ensembles  $(\mathcal{X}', \mathcal{W}', \mathcal{Y}')$ . La partie de  $\Sigma$  dépendante de  $\mathcal{W}'$  est dite réalisable sous forme d'un sous-modèle. De plus, si l'ensemble  $\mathcal{Y}'$  n'est pas nul, alors il est possible de comparer les sorties de  $\Sigma$  et de  $\Sigma'$ . Ceci implique que  $\Sigma$  possède des sorties dépendantes des entrées  $\mathcal{W}'$ .

Si le sous-modèle  $\Sigma'$  est de dimension strictement inférieure au modèle  $\Sigma$  alors le triplet d'applications  $(\Psi_{\mathcal{W}}, \Psi_{\mathcal{X}}, \Psi_{\mathcal{Y}})$  est appelé *homomorphisme* de  $\Sigma$  vers  $\Sigma'$  et  $\Sigma'$  est obtenu par décomposition de  $\Sigma$ . Si les dimensions de  $\Sigma'$  et de  $\Sigma$  sont égales, alors le triplet d'applications  $(\Psi_{\mathcal{W}}, \Psi_{\mathcal{X}}, \Psi_{\mathcal{Y}})$  est appelé *isomorphisme* [HS66] et  $\Sigma'$  est obtenu par transformation de  $\Sigma$ .

Si  $\Sigma'$  est obtenu par décomposition, il est possible de déterminer le sous-modèle complémentaire  $\Sigma_*$  de sorte que la connaissance de  $\Sigma'$  et de  $\Sigma_*$  permet de retrouver le modèle complet  $\Sigma$ .

## 2.3 Surveillance à base de modèles temporels

### 2.3.1 Introduction

Selon les objectifs visés, deux types de modèles comportementaux temporels sont utilisés pour la surveillance :

**Le modèle de bon fonctionnement** représente le système en fonctionnement normal sans tenir compte des défauts potentiels.

**Le modèle de fonctionnement défaillant** décrit le système en tenant compte des défauts. L'influence des défauts est intégrée au modèle du processus.



### 2.3.2 Modèle de bon fonctionnement

Les fonctions d'état  $f$  et de sortie  $h$  de la définition (2.2) décrivent des évolutions temporelles des variables associées au modèle.

Si la variable temps est continue, des équations différentielles du premier ordre sont en général utilisées, on parle alors de *modèle d'état continu*. La forme classique est la suivante :

$$\begin{cases} \dot{x}(t) = f(x(t), w(t)) \\ y(t) = h(x(t), w(t)) \end{cases} \quad (2.6)$$

où  $x \in X \subseteq \mathbb{R}^n$  est le vecteur qui contient les variables internes du système ou *vecteur d'état*,  $w \in W \subseteq \mathbb{R}^l$  le vecteur des excitations qui regroupe les signaux d'excitation connus (commandes) et inconnus (perturbations ou entrées inconnues).  $y \in Y \subseteq \mathbb{R}^m$  est le vecteur des signaux de sortie ou *vecteur de sortie*.

Le vecteur d'excitation s'écrit comme suit :

$$w = \begin{bmatrix} u \\ \gamma \end{bmatrix}$$

où  $u \in U \subseteq \mathbb{R}^q$  est le vecteur des commandes et  $\gamma$  le vecteur des perturbations.

Le signal de perturbation  $\gamma$  correspond à l'influence de l'environnement et à l'influence des erreurs de modélisation.

Si la variable temps est échantillonnée avec une période «  $Te$  », le modèle est dit *modèle d'état échantillonné ou discrétisé* et prend la forme suivante :

$$\begin{cases} x((k+1)Te) = f^*(x(kTe), w(kTe)) \\ y(kTe) = h^*(x(kTe), w(kTe)) \end{cases} \quad (2.7)$$

où  $k$  représente le numéro de l'échantillon qui correspond à l'instant d'échantillonnage «  $kTe$  ». Pour simplifier l'écriture, les variables sont indicées par  $k$  :

$$\begin{cases} x_{k+1} = f^*(x_k, w_k) \\ y_k = h^*(x_k, w_k) \end{cases} \quad (2.8)$$

Les fonctions  $f$  et  $h$  (cas continu),  $f^*$  et  $h^*$  (cas discrétisé) peuvent être linéaires ou non linéaires :

**Définition 2.5** Soit une fonction  $f : X \rightarrow X$ .  $f$  est une fonction linéaire si et seulement si

$$\begin{aligned} \forall x \in X, \forall x' \in X : f(x+x') &= f(x) + f(x') \\ \forall \lambda \in \mathbb{K}, \forall x \in E : f(\lambda \cdot x) &= \lambda \cdot f(x) \end{aligned}$$

La première relation décrit la propriété d'additivité, et la seconde, l'homogénéité.

Dans le cas linéaire continu, le système (2.6) s'écrit

$$\begin{cases} \dot{x}(t) = Ax(t) + Bw(t) \\ y(t) = Cx(t) + Dw(t) \end{cases} \quad (2.9)$$

où A,B,C et D sont des matrices constantes de dimensions appropriées.

Dans le cas discrétisé, le système (2.8) s'écrit

$$\begin{cases} x_{k+1} = Ax_k + Bw_k \\ y_k = Cx_k + Dw_k \end{cases} \quad (2.10)$$

où A,B,C et D sont des matrices constantes de dimensions appropriées.

### 2.3.3 Modèle de fonctionnement défaillant

Le modèle de fonctionnement défaillant (MFD) permet de représenter le comportement du système réel en présence de défauts. Le défaut d'un composant physique a pour effet de modifier la valeur de certains paramètres du modèle, voire même de modifier la structure du modèle. Nous ne considérerons pas ce deuxième cas.

L'influence du défaut sur le modèle peut être représentée à l'aide d'un signal d'excitation supplémentaire. Ce signal, pour le différencier du défaut physique, sera appelé *défaillance*.

Le vecteur d'excitation s'écrit alors

$$w = \begin{bmatrix} u \\ \rho \\ \gamma \end{bmatrix}$$

où  $\rho$  est le vecteur de défaillances.

Le modèle de fonctionnement défaillant peut alors s'écrire sous la forme :

$$\begin{cases} \dot{x}(t) = f'(x(t), w(t)) = f(x(t), u, \gamma) + f_d(x(t), u, \gamma, \rho) \\ y(t) = h'(x(t), w(t)) = h(x(t), u, \gamma) + h_d(x(t), u, \gamma, \rho) \end{cases} \quad (2.11)$$

Le signal de défaillance  $\rho$  est nul en fonctionnement normal. On a dans ce cas :

$$\begin{cases} f_d(x(t), u, \gamma, 0) = 0 \\ h_d(x(t), u, \gamma, 0) = 0 \end{cases} \quad (2.12)$$

Le modèle de fonctionnement défaillant discrétisé s'obtient de manière similaire.

### 2.3.4 Utilisation du modèle pour la surveillance

L'état du système est connu par l'intermédiaire des mesures réalisées par les capteurs. Cette connaissance de l'état n'est cependant que partielle car il est impossible de mesurer toutes les caractéristiques du système. De plus, les capteurs peuvent donner des informations erronées s'ils sont en défaut. La première solution permettant de fiabiliser l'information est de sur-instrumenter le système, on parle alors de *redondance matérielle*. Le principe est de mesurer une même quantité à l'aide de plusieurs capteurs. Les incohérences détectées mettent en évidence les défauts ou pannes des capteurs et permettent de fournir une information validée au système de contrôle/commande. Cette solution largement utilisée dans les domaines sensibles tels l'aéronautique ou

l'aéronautique, est cependant très coûteuse. De plus, elle ne permet pas de détecter les défauts internes du système. La surveillance à base de modèle utilise le modèle comportemental afin d'obtenir une redondance d'informations. Les relations liant les variables de description, permettent de vérifier la cohérence (ou consistance) des mesures prélevées par les capteurs. On parle alors de *redondance analytique* [BKLS03, PFC00].

Cette cohérence (ou non cohérence) est quantifiée par un signal  $r(t)$  appelé *résidu*. En fonctionnement normal, non perturbé et lorsque le modèle correspond exactement au système réel (cas idéal)  $r(t)$  est nul. Si un défaut apparaît, si le modèle présente des incertitudes ou si le système est perturbé, on aura  $r(t) \neq 0$ .

La surveillance nécessite donc deux étapes : la génération des résidus (voir section suivante) et une procédure de décision permettant d'interpréter la valeur des résidus et de décider de l'occurrence ou non d'un défaut. L'utilisation du résidu pour détecter et localiser les défaillances est décrite par le diagramme 2.3 [Ise05].

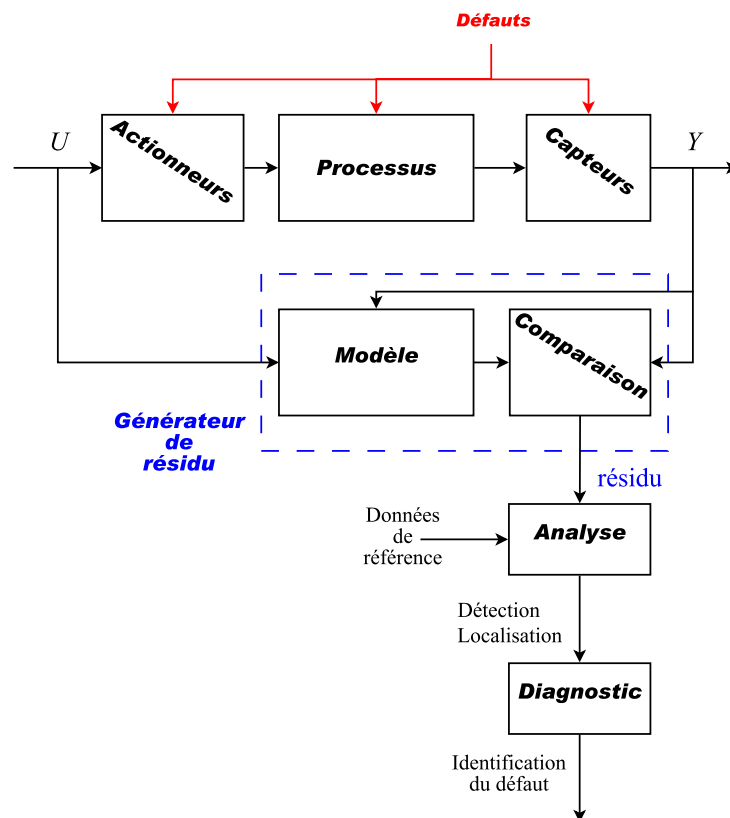


FIG. 2.3 – Illustration de la surveillance à base de modèles

Nous nous intéressons dans cette thèse uniquement à la synthèse de générateurs de résidus. Le lecteur intéressé par les procédures de décisions, pourra consulter les ouvrages [Ger98, BN93]. La procédure de décision sera grandement facilitée si les résidus comportent de *bonnes* propriétés de robustesse et de sensibilité. Dans la section suivante, nous détaillons ces propriétés et décrivons brièvement les méthodes de génération de résidus.

### 2.3.5 Génération de résidus pour la détection de défauts

#### Propriétés des résidus

Comme nous l'avons vu, un résidu devient différent de zéro lorsque le comportement du système réel s'écarte de son comportement nominal donné par un modèle de bon fonctionnement. Ainsi un résidu est une fonction des signaux  $\gamma$  et  $\rho$ . Afin de faciliter la décision, il est intéressant d'obtenir des résidus qui réagissent uniquement à un sous-ensemble de défaillances et qui ne réagissent pas, ou réagissent faiblement, aux perturbations et aux autres défaillances. Un résidu possède donc des propriétés de *robustesse* à certains signaux et de *sensibilité* à d'autres.

**Définition 2.6** Un résidu  $r(t)$  est dit robuste par rapport à un signal  $\rho$  si  $r(t)$  ne dépend pas de  $\rho$ . En terme mathématique, cela signifie que

$$r(t) \text{ robuste par rapport à } \rho \Leftrightarrow \frac{\partial r(t)}{\partial \rho} = 0 \quad (2.13)$$

La sensibilité est définie de manière analogue.

**Définition 2.7** Un résidu  $r(t)$  est dit sensible à un signal  $\rho$  si une variation donnée de  $\rho$  engendre une variation de  $r(t)$ . En terme mathématique, cela signifie que :

$$r(t) \text{ sensible à } \rho \Leftrightarrow s_\rho(r(t)) = \frac{\partial r(t)}{\partial \rho} \neq 0 \quad (2.14)$$

La sensibilité asymptotique par rapport à  $\rho$  est définie comme la limite quand  $t$  tend vers l'infini de  $s_\rho(r(t))$  :

$$s_\rho^\infty(r(t)) = \lim_{t \rightarrow \infty} s_\rho(r(t)) = \lim_{t \rightarrow \infty} \frac{\partial r(t)}{\partial \rho} \quad (2.15)$$

Par extension, lorsque la sensibilité asymptotique par rapport à un signal est nulle, on parlera de *robustesse asymptotique*.

#### Méthodes de génération de résidus

D'un point de vue général, un résidu peut être vu comme la sortie d'un système dynamique, appelé *générateur de résidus (GR)*, construit à partir du modèle de bon fonctionnement (fonctions  $f$  et  $h$ ), ayant en entrée les signaux d'excitation connus et les mesures issues des capteurs (figure 2.3). Ce système doit être conçu, ou synthétisé, de manière à ce que le résidu possède des propriétés de robustesse/sensibilité fixées :

- robustesse vis à vis des perturbations et erreurs de modélisation
- robustesse asymptotique par rapport à des conditions initiales inconnues
- sensibilité par rapport à des défaillances

Notons que la robustesse et la sensibilité sont des propriétés associées aux signaux. Un système générateur d'un résidu robuste (resp. sensible) par rapport à un signal  $\rho$  sera dit *découplé* de (resp. *couplé* à) ce signal.

Trois approches principales sont utilisées pour construire un générateur de résidus [Pat94, Ise05] :

## Les approches d'identification

Si la structure du modèle est connue, les approches à base d'estimation de paramètres sont adaptées pour synthétiser le générateur de résidus. Elles consistent à identifier en ligne les divers paramètres du système et à comparer ces estimations aux valeurs nominales des paramètres. L'erreur d'estimation est utilisée comme résidu. Le lecteur intéressé par cette approche pourra consulter par exemple les travaux de Isermann [Ise84, IF91] ou les travaux de Fliess, Sira-Ramirez et Join [FJ03, FJSR04].

## Approches à base de relations de redondance analytique ou espace de parité

Le principe est de *récrire* les équations du modèle de manière à obtenir des relations particulières appelées RRA : Relations de Redondance Analytique. Cette redondance est directe, si l'on utilise des relations *statiques* [DGH79], ou bien *temporelle* [CW84] si les relations d'entrées/sorties sont dynamiques, en faisant intervenir les dérivées des signaux.

Les RRA ont pour propriété de ne lier que des grandeurs connues, disponibles en ligne. Les résidus sont obtenus en substituant dans ces RRA les variables littérales par leurs valeurs réelles, prélevées sur le système en fonctionnement.

L'obtention hors-ligne des RRA est un problème général d'élimination de variables dans un système d'équations algébriques ou différentielles. Lorsque le modèle est linéaire, l'élimination peut se faire par projection dans un sous-espace appelé *espace de parité* [CW84]. Dans le cas non linéaire, des techniques d'élimination formelle peuvent être utilisées.

**Projection** Cette méthode a été proposée dans [CW84]. Les relations de redondance s'expriment dans l'espace non observable du système. Les auteurs ont traité le cas linéaire. La non-linéarité des modèles impose des contraintes supplémentaires, liées notamment à la détermination de l'espace de non-observabilité. Des développements récents, basés notamment sur des techniques de géométrie différentielle [Isi95], ont proposé la génération de relations de redondance pour des classes de systèmes non linéaires à l'instar de [LWC05].

**Élimination** Les relations de redondance sont obtenues par combinaison des relations existantes dans le modèle analytique. La structuration des résidus et leur robustesse sont obtenus en éliminant les variables correspondantes des relations. Plusieurs implémentations logicielles existent, les plus importantes sont basées sur la théorie de l'élimination [Dio91], les bases de Gröbner [CLO91] et les ensembles caractéristiques [Rit50]. Les bases de Gröbner ont été utilisées pour générer des RRA non linéaires dans [MCC<sup>+</sup>97, SCV01].

Il faut noter que les techniques à base de relations de redondance analytique possèdent des similarités avec les approches de détection par observateurs à entrées inconnues (voir paragraphe suivant). Cet aspect a été initialement souligné dans [Mas86] et développé dans [PC91, Chr01, CCJ04].

### Approches à base d'observateurs ou de filtres détecteurs

Ce sont les plus couramment utilisées. Les premiers travaux datent des années soixante-dix [Bea71, Jon73, Cla78]. Les observateurs ou filtres sont des outils bien connus des automaticiens à des fins de commande en boucle fermée. Le principe général est de concevoir un système dynamique permettant de donner une estimation, de certaines variables, ou combinaisons de variables, nécessaires au bouclage. Lorsque le système est dynamique et que certaines variables (conditions initiales) sont inconnues, l'estimation n'est correcte qu'après un certain temps de convergence, fixé par la dynamique de l'observateur. Le principe général consiste à comparer des fonctions de sorties estimées avec les mêmes fonctions des sorties mesurées. L'écart entre ces fonctions est utilisé comme résidu [Fra1, Cla78, Cla89].

Des variantes existent, notamment les approches à base d'observateurs à entrées inconnues, basées sur le principe l'observateur de Luenberger généralisé [Lue66], permettant d'obtenir des résidus robustes [Fra90], ou de filtres détecteurs [Bea71, Jon73] et [Mas86, MVW89] où le problème a été réécrit sous un formalisme géométrique. Depuis le début des années quatre-vingt dix, les approches de détection à base d'observateurs et de filtres ont été étendues aux systèmes non linéaires en utilisant des approches à base de géométrie différentielle :

- pour le cas général [Isi95, BSC04],
- pour les systèmes affines en l'état et bilinéaires [HKE99b, HKE01],
- pour les systèmes affines en la commande [DI00, DI01, Loo01].

#### 2.3.6 Résidus structurés pour la localisation de défauts

**Définition 2.8** La structure d'un résidu  $r_i$  par rapport à un ensemble  $\rho$  de  $\eta_\rho$  défauts est le mot binaire  $S_{r_i}$  composé de  $\eta_\rho$  bits  $(S_{i,j})$  positionnés de la manière suivante :

- $S_{i,j} = 1$  si le résidu  $r_i$  sensible au  $j^{\text{ième}}$  élément de  $\{\rho\}$ .
- $S_{i,j} = 0$  si le résidu  $r_i$  est robuste au  $j^{\text{ième}}$  élément de  $\{\rho\}$ .

Les résidus structurés sont conçus de manière à être sensibles chacun à un sous-ensemble de fautes  $\rho_1$  et robustes par rapport aux fautes restantes  $\rho_2, \rho_3, \dots$  (fig. 2.4). Ainsi, lorsqu'une défaillance apparaît, seul un sous-ensemble de résidus réagit.

Les informations de sensibilité et de robustesse souhaitées pour les résidus sont répertoriées dans une table binaire appelée *table de signatures théoriques*. Les lignes de cette table correspondent à la structure de chaque résidu. Les colonnes correspondent à la signature d'un défaut particulier.

Afin de localiser un défaut sans ambiguïté, les signatures des défauts doivent être différentes. Une distance de Hamming élevée entre les signatures permettra de tolérer d'éventuelles erreurs de décisions.

Un ensemble de résidus possédant des propriétés spécifiques de sensibilité ou de robustesse par rapport aux défauts est donc calculé en ligne. La procédure de décision appliquée à chaque résidu, permet d'obtenir *la signature réelle* des résidus à un instant donné. Si cette signature est nulle, le système est déclaré sain. Dans le cas contraire,

la signature réelle est comparée aux signatures théoriques. Le défaut est déterminé en considérant le plus proche voisin de la signature calculée.

Deux démarches peuvent être adoptées pour générer un vecteur de résidus robustes et structurés. La première démarche décrite par la figure 2.4, est classiquement adoptée dans la littérature. Elle consiste à construire, à partir du modèle du système, un ensemble de générateurs permettant d'obtenir des résidus possédant les propriétés de robustesse/sensibilité souhaitées et garantissant la robustesse (au moins asymptotique) vis à vis des conditions initiales.

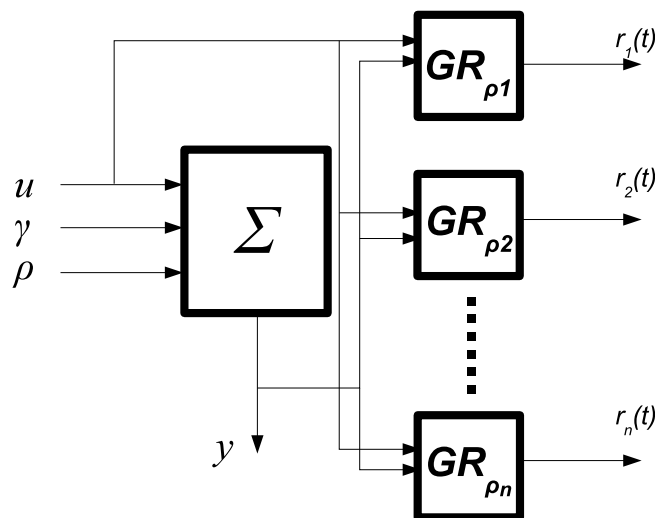


FIG. 2.4 – Banc de générateurs de résidus robustes

La deuxième démarche est décrite par la figure 2.5. Le principe est de décomposer le modèle de fonctionnement défaillant en plusieurs sous-modèles. Chaque sous-modèle est découplé d'un sous-ensemble de défaillances, ce qui implique que ses sorties seront robustes par rapport à celles-ci.

Par ailleurs, ce sous-modèle sera couplé au reste des défaillances, les sorties seront donc potentiellement sensibles à celles-ci. Bien entendu, tous les sous-modèles seront découplés des perturbations.

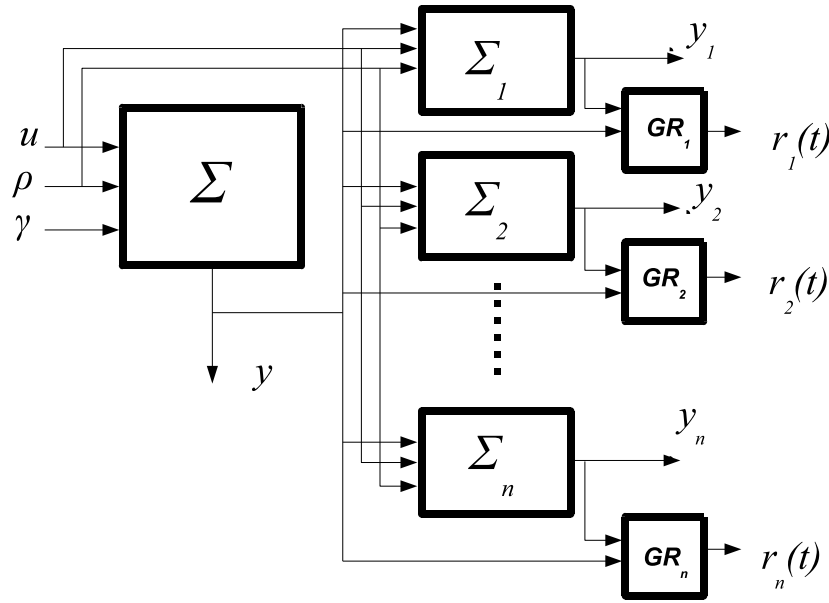


FIG. 2.5 – Utilisation d'un banc de sous-systèmes découplés pour la génération de résidus structurés robustes

La différence principale entre les deux approches réside dans la méthode de synthèse du bloc générateur de résidu.

Considérons le système  $\Sigma$  suivant

$$\Sigma : \begin{cases} \dot{x} = f(x, w) \\ y = h(x, u, \rho) \end{cases}, w = \begin{bmatrix} u \\ \rho \\ \gamma \end{bmatrix} \quad (2.16)$$

où  $x \in X \subseteq \mathbb{R}^n$ ,  $w \in W \subseteq \mathbb{R}^l$  et  $y \in Y \subseteq \mathbb{R}^m$  représentent respectivement les vecteurs d'état, d'excitation et de sortie. Rappelons que vecteur d'excitation est composé de trois vecteurs :  $u \in U \subseteq \mathbb{R}^q$  est le vecteur de commandes,  $\gamma$  est le vecteur de perturbations et  $\rho$  est le vecteur de défaillances.

Les différents générateurs de résidus  $GR_i$  ont la forme suivante [HKE99a, DI00]

$$GR_i : \begin{cases} \dot{z}_i = \Lambda(z_i, u, y) \\ r_i = \Theta(z, y) \end{cases} \quad (2.17)$$

Le résidu  $r_i$  possède trois propriétés essentielles :

- être robuste au vecteur de perturbations  $\gamma_i$
- être sensible aux défaillances regroupées dans  $\rho_i$
- convergence asymptotique en absence de défaillance :

$$\text{si } \rho = 0, \forall \gamma_i : \lim_{t \rightarrow +\infty} r_i(t) = 0$$



Les vecteurs  $\gamma_i$  et  $\rho_i$  sont obtenus par permutation des composantes des vecteurs  $\rho$  et  $\gamma$ .

Suivant la méthode classique, le générateur de résidus doit être construit en respectant simultanément les trois propriétés de robustesse, de sensibilité et de convergence. La synthèse de tels générateurs peut être difficile à mener dans un contexte non linéaire, puisque de nombreuses contraintes doivent être satisfaites en même temps en manipulant le modèle complet du système.

La seconde méthode propose de résoudre le problème de convergence (3<sup>e</sup> propriété du résidu) séparément, en synthétisant le générateur de résidu à partir des sous-modèles  $\Sigma_i$  (2.18)

$$\Sigma(X, W, Y, f, h) \implies \begin{cases} \Sigma_1(X_1, u \cup \rho_1, Y_1, f_1, h_1) \\ \Sigma_2(X_2, u \cup \rho_2, Y_2, f_2, h_2) \\ \vdots \\ \Sigma_k(X_k, u \cup \rho_i, Y_k, f_k, h_k) \end{cases} \quad (2.18)$$

Cette distribution des contraintes permet de simplifier l'opération de synthèse du générateur de résidus.

Chaque sous-modèle  $\Sigma_i$  est équivalent en fonctionnement avec une partie du modèle complet. L'ensemble des  $\Sigma_i$  complémentaires sont équivalents au modèle complet. Par conséquent, pour des conditions initiales identiques, l'évolution de  $\Sigma$  ou des différentes parties de  $\Sigma$  est estimée par les sous-modèles  $\Sigma_i$ . Si les conditions initiales ne sont pas connues, un générateur de résidu  $GR_i$  est synthétisé à partir de chaque sous-modèle  $\Sigma_i$  pour garantir la convergence de l'indicateur de défaillances. La procédure de synthèse est simplifiée puisque chaque sous-modèle  $\Sigma_i$  génère par définition des sorties robustes aux entrées de l'ensemble  $\gamma_i$  et sensibles aux entrées de l'ensemble  $u \cup \rho_i$ .

La procédure de décomposition est détaillée dans le chapitre 4.

## 2.4 Surveillance à base de modèles événementiels

### 2.4.1 Introduction

La surveillance des systèmes à dynamique événementielle poursuit les mêmes objectifs que la surveillance des systèmes à dynamique temporelle. Les outils pour réaliser la surveillance par contre sont différents. Puisque les états que peut prendre le système sont dénombrables, il est possible dans de nombreux cas de réaliser des simulations exhaustives pour reconstituer le fonctionnement du modèle événementiel. Les approches à base de simulation sont plus répandues que dans le cas des systèmes à états continus. Nous allons étendre la méthode proposée dans la section précédente à la surveillance des systèmes décrits par des modèles événementiels. Le principe est identique : synthétiser des sous-modèles qui possèdent des propriétés de couplage ou de découplage par rapport aux sollicitations extérieures. Les sorties des sous-modèles sont ensuite comparées aux sorties mesurées et un indicateur de défaillances est gé-

né. Dans cette section, nous étendons les notions propres à la surveillance à base de modèles temporels au cadre événementiel.

### 2.4.2 Représentation événementielle : principe et rappels

Une dynamique événementielle est souvent représentée par des machines séquentielles (MS), appelées également automate à états finis (AEF). Une machine séquentielle est constituée d'états et de transitions. Le calcul de l'état suivant tient compte de l'état présent de la machine et de l'entrée d'excitation. C'est donc un modèle d'évolution comportemental. Il existe deux classes de machines séquentielles : déterministe et non déterministe. Un automate est déterministe si à partir de n'importe quel état prédécesseur, un seul état successeur existe pour une entrée donnée. Toutes les machines qui ne remplissent pas cette condition sont non déterministes.

La machine séquentielle déterministe a deux avantages principaux :

- la simplicité de modélisation
- une représentation intuitive du processus

et un inconvénient : le nombre d'états augmente considérablement dès que des mécanismes complexes sont modélisés. La synthèse de ce type d'automates s'accompagne souvent d'une étape de réduction du nombre d'états.

La représentation non déterministe permet de réduire la taille de l'automate, et de l'enrichir par des mécanismes de transitions d'état avancés (synchronisation, parallélisme,...). L'inconvénient réside dans le fait que pour la simulation, une étape de détermination, par l'ajout de transitions vides<sup>1</sup>, est nécessaire. Ces opérations sont coûteuses en temps de calcul. Cela implique que leur utilisation en temps réel est délicate. Dans ce mémoire, seules les machines séquentielles déterministes sont considérées.

Il existe plusieurs types d'automates : les machines de Mealy et de Moore, les automates à arbres, les automates accepteurs... Nous utiliserons la modélisation de systèmes événementiels par la machine de Moore définie ci-dessous.

**Définition 2.9** La machine de Moore est un modèle événementiel  $\Sigma(S, I, O, \delta, \lambda)$  où  $S$  est l'ensemble d'état,  $I$  est l'ensemble d'entrée ou alphabet d'entrée, et  $O$  l'ensemble de sortie et les applications  $\delta$  et  $\lambda$  sont définies par

$$\begin{array}{ll} \delta: S \times I \longrightarrow S & \lambda: S \longrightarrow O \\ (s, i) \longrightarrow s^+ = \delta(s, i) & s \longrightarrow o = \lambda(s) \end{array}$$

Si l'application  $\lambda$  est l'identité, la machine de Moore devient une *machine d'état*.

**Définition 2.10** Une machine d'état est représentée par le triplet  $(S, I, \delta)$  où  $S$  est l'ensemble fini d'états,  $I$  est l'ensemble fini d'entrées et  $\delta: S \times I \rightarrow S$  est l'application de transition d'état.

<sup>1</sup>Les transitions vides, ou  $\epsilon$ -transitions, sont des transitions qui ne sont pas liées à une condition ou à un événement, et qui sont franchies automatiquement (voir les références [SSL<sup>+</sup>95, SSL<sup>+</sup>96] pour plus d'information).

Une machine séquentielle est communément représentée par une table de transitions d'état.

	$i_1$	$\dots$	$i_l$	$o$
$s_1$	$\delta(s_1, i_1)$	$\dots$	$\delta(s_1, i_l)$	$\lambda(s_1)$
$\vdots$	$\vdots$		$\vdots$	$\vdots$
$s_n$	$\delta(s_n, i_1)$	$\dots$	$\delta(s_n, i_l)$	$\lambda(s_n)$

TAB. 2.1 – Table de transition d'état

Le fonctionnement d'une machine peut être également décrit par un graphe d'état (fig. 2.6)

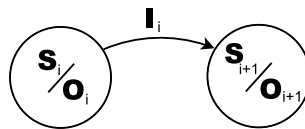


FIG. 2.6 – Graphe d'état d'une machine de Moore

**Illustration** La figure (2.7) montre l'exemple d'une machine séquentielle et de sa table de transitions d'état (2.2)

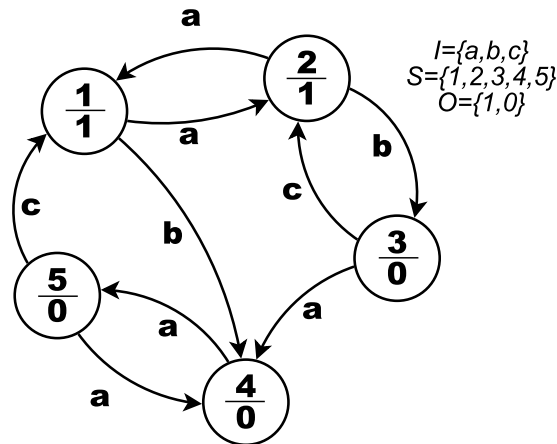


FIG. 2.7 – Graphe d'état de  $\Sigma$

S\I	a	b	c	Y
1	2	4	1	1
2	1	3	2	1
3	4	3	2	0
4	5	4	4	0
5	4	5	1	0

TAB. 2.2 – Table de transition d'état de  $\Sigma$

Le fonctionnement d'une machine séquentielle est décrit par des *chemins* qui expriment la réaction de la machine à une séquence définie sur l'ensemble d'entrées, ou *alphabet*. Dans la représentation temporelle, le chemin correspond à une trajectoire et la séquence à un signal.

**Définition 2.11** Un « chemin » est une suite de transitions d'état consécutives. Il est noté

$$[(s_0, i_1, s_1) \cdots (s_{i-1}, i_i, s_i)]$$

avec  $s_i \in S$ ,  $i_i \in I$  et  $(s_{j-1}, i_j, s_j) \in (S \times I \times S)$ . Une suite d'événements  $i_1 \cdots i_k$  est appelée indifféremment « trace » ou « étiquette » ou « séquence ».

### 2.4.3 Modèle de bon fonctionnement

Le modèle de bon fonctionnement d'un système à dynamique événementielle est représenté par une machine séquentielle  $\Sigma(S, I, O, \lambda, \delta)$  qui décrit son évolution en absence de défauts. Les machines séquentielles sont considérées *complètes*<sup>2</sup>, et *vivantes*, c'est-à-dire qu'elles ne possèdent pas d'états *bloquants*<sup>3</sup>.

L'alphabet d'entrée  $I$  du modèle de bon fonctionnement est composé de

- $I_c$  : ensemble d'événements connus associés aux actions du système de contrôle ou aux transitions d'état.
- $I_\gamma$  : ensemble d'événements inconnus associés aux perturbations externes.

En représentation événementielle, la perturbation se manifeste par un changement imprévu de l'état du système qui n'est pas causé par un défaut.

Nous qualifierons les états et événements de « connus » ou « mesurés », plutôt que d'observables ou observés — utilisés par la communauté des systèmes à dynamique événementielle — pour faire la distinction entre la propriété d'observabilité qui est une propriété structurelle d'un modèle, et l'accessibilité d'un état à partir des mesures.

### 2.4.4 Modèle de fonctionnement défaillant

Le modèle de fonctionnement défaillant est une machine séquentielle construite sur la base du modèle de bon fonctionnement auquel est ajouté l'influence des *défaillances*.

**Définition 2.12** Dans une dynamique événementielle, la *défaillance* est un événement inconnu associé à un défaut physique donné.

L'alphabet d'entrée du modèle de fonctionnement défaillant  $\Sigma(S, I, O, \delta, \lambda)$  inclut un ensemble d'événements non mesurés  $I_\rho$  qui regroupe les défaillances possibles

$$I = I_c \cup I_{nc} = I_c \cup (I_\gamma \cup I_\rho)$$

<sup>2</sup>Toutes les paires  $(s, i) \in S \times I$  ont une seule et unique image *via*  $\delta$

<sup>3</sup>Un état  $s$  est dit bloquant si  $\forall i \in I : \delta(s, i) = s$

où  $I_{nc}$  représente l'ensemble de tous les événements inconnus et  $I_p$  représente l'ensemble des défaillances. La défaillance provoque un changement d'état du système qui passe du fonctionnement normal au fonctionnement défaillant. Dans certaines représentations, les états défaillants du système sont représentés par un ensemble d'états  $S_p$  ajoutés à l'ensemble d'états de *bon fonctionnement*  $S_e$ . L'ensemble d'états est formé alors de deux sous-ensembles :  $S = S_e \cup S_p$ .

Une défaillance à plusieurs degrés de détectabilité : fortement détectable, faiblement détectable ou non détectable.

**Définition 2.13** Une défaillance est dite :

**fortement détectable** si dès que la défaillance se produit, les sorties deviennent anormales,

**faiblement détectable** si l'occurrence de la défaillance entraîne des sorties anormales du système après un nombre fini (non nul) de transitions d'état,

**non détectable** si la défaillance n'entraîne jamais de sorties anormales.

**Illustration** Trois exemples sont donnés pour les trois degrés de détectabilité possibles. Considérons la machine séquentielle  $\Sigma_1$  décrite par le diagramme (2.8).

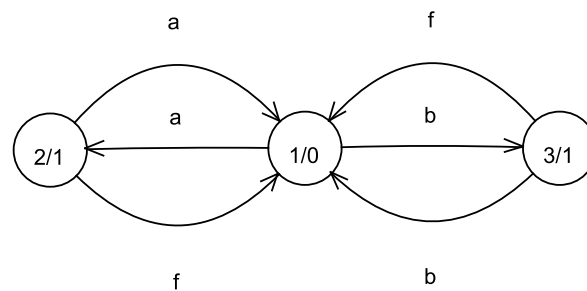


FIG. 2.8 – Représentation de la machine séquentielle  $\Sigma_1$

L'occurrence de l'événement inconnu  $f$  provoque un changement de sortie immédiat, quelle que soit la séquence d'entrée (table 2.3).  $\Sigma_1$  est initialement dans l'état 1 avec une sortie 0.

Entrée	a	a	a	f	a	b	b	f	b	b
État	2	1	2	1	2	2	2	1	3	1
Sortie	1	0	1	0	1	1	1	0	1	0
Est. état	2	1	2	1	3	1	3	1		
Est. sortie	1	0	1	0	1	0	1	0		

TAB. 2.3 – Evolution de la machine  $\Sigma_1$

La table 2.3 décrit les chemins d'évolution de  $\Sigma_1$  (modèle défaillant et modèle normal) en réponse à la séquence d'entrée :

$$seq_e = [a, a, a, f, a, b, b, f, b, b]$$

Le tableau doit être lu comme suit :  $\Sigma_1$  étant à l'état 1 avec une sortie à 0, l'événement *a* (première colonne) la fait passer à l'état 2 (au dessous) avec la sortie 1.

Considérons à présent la machine  $\Sigma_2$  (figure 2.9).

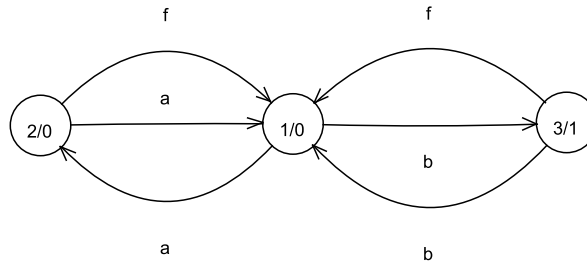


FIG. 2.9 – Représentation de la machine séquentielle  $\Sigma_2$

Si l'événement *f* se produit, un changement de sorties peut être observé seulement si une séquence particulière se produit à l'entrée (table 2.4).

Entrée	a	a	a	f	a	b	b	f	b	b
État	2	1	2	1	2	2	2	1	3	1
Sortie	0	0	0	0	0	0	0	0	1	0
Est. état	2	1	2	1	3	1	3	1		
Est. sortie	0	0	0	0	1	0	1	0		

TAB. 2.4 – Evolution de la machine  $\Sigma_2$

Le dernier cas concerne une défaillance non détectable. La figure (2.10) décrit une machine séquentielle  $\Sigma_3$  sujette à une défaillance *f* impossible à détecter.

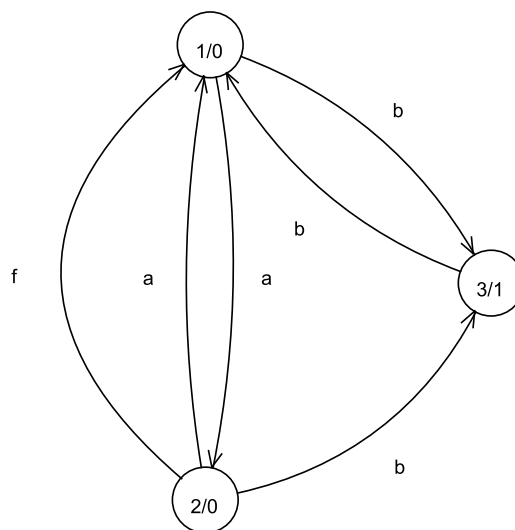


FIG. 2.10 – Représentation de la machine séquentielle  $\Sigma_3$

Quelle que soit la séquence d'entrée, l'occurrence de  $f$  ne peut pas être observée sur les sorties (table 2.5).

Entrée	a	a	a	f	a	b	b	f	b	b
État	2	1	2	1	2	3	1	1	3	1
Sortie	0	0	0	0	0	1	0	0	1	0
Est. état	2	1	2	1	3	1	3	1		
Est. sortie	0	0	0	0	1	0	1	0		

TAB. 2.5 – Evolution de la machine  $\Sigma_3$

La défaillance peut être diagnosticable ou non diagnosticable. Un événement inconnu est diagnosticable si le chemin d'évolution qu'il induit est impossible à reproduire en son absence avec toute autre séquence d'événements [SSL<sup>+</sup>95, SSL<sup>+</sup>96].

**Définition 2.14** Soit un événement  $e \in I$  et deux séquences  $a_0 \in I^*$  et  $a_1 \in I^*$  telles que  $I' = (I - \{e\})$  et  $\Pi_{I'}(a_0) = a_1$ . On dit que l'événement  $e$  est diagnosticable si et seulement si  $\lambda(\delta(s_0, a_0), a_0) = \lambda(\delta(s_0, a_1), a_1)$ .

La notation  $\Pi_{I'}(a_0)$  désigne la projection de  $a_0$  sur l'ensemble  $I'$ , et la notation  $I^*$ , où  $I$  est un ensemble fini d'éléments qui désigne l'ensemble de toutes les séquences possibles de longueur finie formées d'éléments de  $I$ .

La notion de diagnosticabilité est l'équivalent événementiel de la notion de localisabilité pour une défaillance dans le cadre temporel.

L'influence de la défaillance sur l'évolution du modèle est représentée par des séquences de transitions d'états anormales, qui peuvent être reproduites par le modèle de fonctionnement défaillant mais pas par le modèle de bon fonctionnement.

## 2.4.5 Techniques de surveillance à base de modèles événementiels

### Généralités

De manière très générale, la surveillance d'un système représenté par un modèle événementiel s'effectue selon le schéma de la figure (2.11).

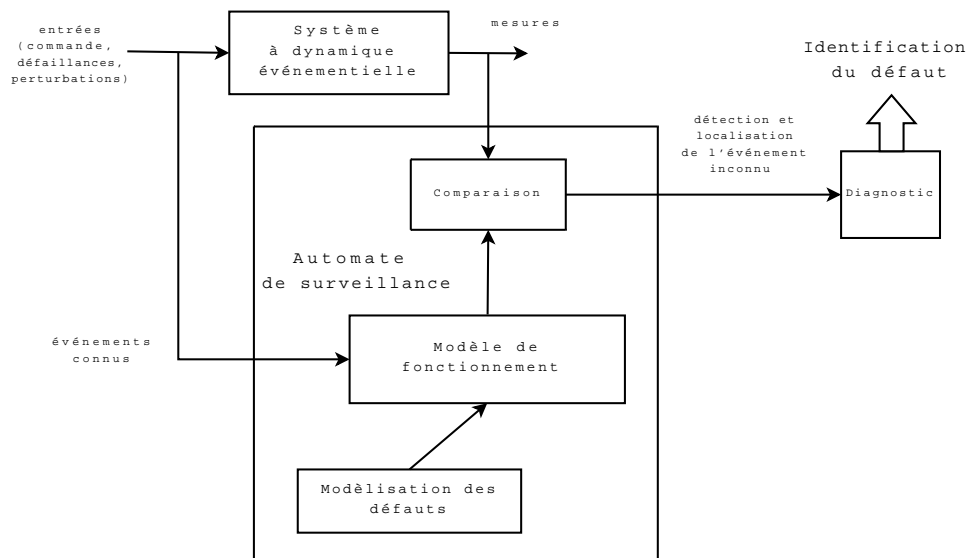


FIG. 2.11 – Méthode de surveillance des systèmes à dynamique événementielle

Les mesures relevées sur le système réel sont confrontées aux estimations fournies par le modèle de fonctionnement. La différence entre les deux séquences est calculée en comparant la sortie mesurée et la sortie estimée par le modèle. Cette différence peut prendre les valeurs suivantes

$$r_k = \begin{cases} 0 & o_k = \hat{o}_k \\ 1 & o_k \neq \hat{o}_k \end{cases}$$

où  $o_k$  représente la mesure à l'itération  $k$ , et  $\hat{o}_k$  représente la mesure estimée à l'instant  $k$ . La séquence<sup>4</sup>  $r$  est analysée : le « 0 » représente un fonctionnement normal, alors que le « 1 » renseigne sur l'occurrence d'un dysfonctionnement. Le module « diagnostic » se charge d'identifier le défaut qui a causé le dysfonctionnement.

La séquence  $r$  sert d'indicateur de défaillance. Il y sera fait référence par séquence indicatrice, ou plus simplement par indicateur de défaillance.

Une forte similitude existe entre la figure (2.11) et la figure (2.3) qui représente le schéma général de surveillance des systèmes représentés par des modèles temporels. La différence principale est que dans le cas événementiel, la comparaison peut se faire soit entre les sorties à un instant donné, soit entre les séquences de sorties sur un intervalle donné. L'analyse des séquences est réalisée en temps réel ou hors ligne selon l'application. Cette analyse permet de reconstituer, à partir de la modélisation des défauts, l'événement qui a causé la déviation du système du chemin prévu et le module de diagnostic permet d'identifier le défaut incriminé.

Notons que le *diagnostiqueur*, défini dans [SSL<sup>+</sup>96, LTS<sup>+</sup>01], s'apparente à l'automate de surveillance de la figure (2.11).

<sup>4</sup>voir définition 2.11



### Approches classiques de surveillance à base de modèle événementiel

Les principales approches de surveillance à base de modèles événementiels, qu'ils soient déterministes ou non déterministes, sont présentées. Pour le cas déterministe, deux méthodes existent, classées par type de représentation du défaut.

La première se base sur une représentation de la condition du système par des états. Il s'agit alors d'estimer dans quel état se trouve le système à un instant donné, connaissant les sorties. Les états du système sont répartis en sous-ensembles, chaque sous-ensemble correspondant à une condition du système : fonctionnement normal ou bien sujet à l'une des défaillances possibles. Dans [LW88], ce problème est étudié dans les deux cas hors-ligne et en ligne. Les travaux de Zad [HZKW03, Zad99] vont également dans ce sens, en utilisant un automate de surveillance pour estimer la condition du système en estimant l'état dans lequel il se trouve.

La seconde approche est une approche basée sur une classification de la condition du système en fonctions des chemins générés [SSL<sup>+</sup>95, SSL<sup>+</sup>96]. Il s'agit de détecter l'occurrence d'une défaillance en comparant les chemins relevés avec des *signatures* de défaillances. Ces signatures sont des chemins obtenus par analyse du modèle de fonctionnement défaillant.

Si la connaissance des chemins est insuffisante pour détecter les défauts, l'utilisation d'automates temporisés peut être considérée. Un automate temporisé est un automate avec des délais d'exécution associés aux transitions ou aux places. Cette information permet d'enrichir l'algorithme de surveillance pour localiser les défaillances qui ne seraient pas diagnosticables autrement (voir [ZKW99, BCD05]).

Le cas de défauts simultanés est traité dans [UUTY03] en utilisant une modélisation à événements discrets concurrents introduite dans [TU03].

La majorité des recherches menées sur le thème de la surveillance des systèmes événementiels non déterministes, consistent en la surveillance de processus représentés par des réseaux de Petri. Dans les travaux [UOO98, Ush02], une extension de l'approche par *diagnostiqueur* [SSL<sup>+</sup>96] a été proposée. L'auteur construit un observateur discret sur les graphes de marquage générés à partir du réseau de Petri. Dans [Gha05] l'auteur utilise une représentation en automates temporisés pour synthétiser un automate de surveillance.

Une approche originale a été présentée par les auteurs de [HV99, Had03] qui proposent des outils algébriques pour détecter et corriger les défaillances. Le principe repose sur la synthèse d'un banc de machines séquentielles redondantes en utilisant des techniques de codage proches des codes de Hamming, pour reconstituer la défaillance et la corriger. La représentation choisie est l'espace d'état discret, ce qui permet aux auteurs de traiter les deux cas : déterministe et non déterministe.

#### 2.4.6 Localisation d'événements inconnus par structuration des indicateurs de défaillances

L'approche proposée est une extension de la méthode présentée dans la section dédiée aux systèmes à dynamique temporelle, et qui consiste à décomposer le modèle du système à surveiller en sous-modèles découplés pour générer des résidus structurés.

Dans le cadre événementiel, la structuration des indicateurs de défauts est obtenue par synthèse de sous-modèles découplés chacun d'un sous-ensemble d'entrées en particulier.

Avant d'aller plus loin, nous allons donner les définitions des notions de couplage et de découplage.

**Définition 2.15** Soit une machine séquentielle décrite par  $\Sigma(S, I, O, \delta, \lambda)$ .  $\Sigma$  est couplée à un événement  $e \in I$  si et seulement si l'occurrence de  $e$  cause un changement d'état de  $\Sigma$

$$\exists s \in S : \delta(s, e) \neq s$$

**Définition 2.16** Soit une machine séquentielle décrite par  $\Sigma(S, I, O, \delta, \lambda)$ .  $\Sigma$  est découplée d'un événement  $e \in I$  si et seulement si l'occurrence de  $e$  ne cause aucun changement d'état de  $\Sigma$

$$\forall s \in S : \delta(s, e) = s$$

La machine séquentielle qui réalise le modèle de fonctionnement défaillant du système complet est décomposée en machines séquentielles partielles  $\Sigma_i$  (figure 2.12).  $i_i$  et  $o_i$  représentent respectivement la séquence d'entrée et la séquence de sortie.

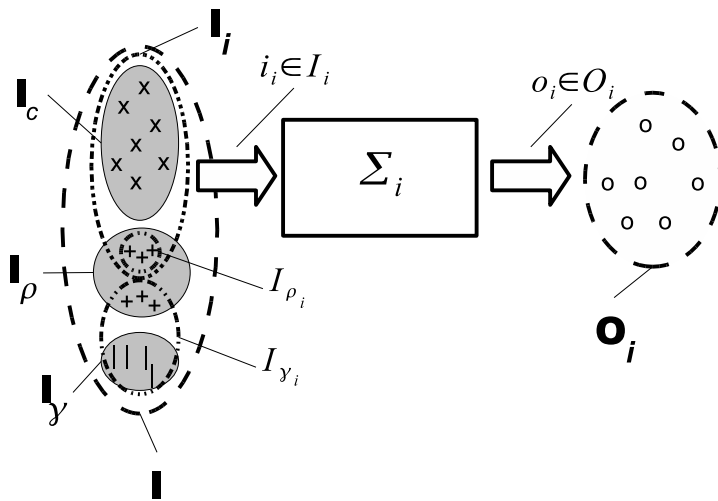


FIG. 2.12 – Diagramme représentant une machine séquentielle partielle

$\Sigma_i$  est décrite par le quintuplet  $\Sigma_i(S_i, I_i, O_i, \delta, \lambda)$  avec

- $I_i$  : l'alphabet d'entrée  $I_i = I_c \cup I_{\rho_i} \subset I$  où  $I_c$  est l'ensemble des événements connus,  $I_{\rho_i}$  l'ensemble des événements inconnus couplés à  $\Sigma_i$ .
- $S_i$  : l'ensemble d'états de  $\Sigma_i$ .  $S_i$  est une *partition* de  $S$ . La notion de partition est abordée au chapitre suivant.
- $O_i$  : l'ensemble de sorties de  $\Sigma_i$ .  $O_i$  est une partition de  $O$ .

Chacune de ces machines est couplée à  $I_c$  et à un sous-ensemble  $I_{\rho_i}$  de l'ensemble de défaillances  $I_{\rho} \cup I_{\gamma}$  et découplée du sous-ensemble  $I_{\gamma_i} \subseteq I_{\rho} \cup I_{\gamma}$ .

Les sous-ensembles  $I_{\rho_i}$  et  $I_{\gamma_i}$  correspondants sont déterminés par une méthode semblable à celle vue dans le cadre temporel.

$$\begin{array}{lll}
\Sigma : & I_\gamma = \{\gamma_1, \gamma_2, \dots, \gamma_q\} & I_\rho = \{\rho_1, \rho_2, \dots, \rho_k\} \\
\Sigma_1 : & I_{\gamma_1} = I_\gamma \cup \{\rho_1\} & I_{\rho_1} = \{\rho_2, \dots, \rho_k\} \\
\Sigma_2 : & I_{\gamma_2} = I_\gamma \cup \{\rho_2\} & I_{\rho_2} = \{\rho_1, \rho_3, \dots, \rho_k\} \\
\vdots & \vdots & \vdots \\
\Sigma_k : & I_{\gamma_k} = I_\gamma \cup \{\rho_k\} & I_{\rho_k} = \{\rho_1, \dots, \rho_{k-1}\}
\end{array} \quad (2.19)$$

Les machines partielles  $\Sigma_i$ ,  $i = 1, \dots, k$  permettent de générer  $k$  séquences indicatrices  $r_i$  obtenues en comparant les mesures  $o$  avec les sorties de  $\Sigma_i$ , notées  $o_i$ . Puisque une machine séquentielle partielle  $\Sigma_i$  est un sous-modèle de  $\Sigma$  alors son évolution est équivalente à une partie de la machine séquentielle complète. En outre, si les états initiaux de  $\Sigma$  et de  $\Sigma_i$  sont identiques alors les sorties de  $\Sigma_i$  permettent d'estimer la partie de  $\Sigma$  décrite par  $(S_i, I_i, O_i)$ , et en fonctionnement normal les sorties mesurées et estimées sont cohérentes.

L'ensemble de la structure de détection et de localisation est résumé dans le schéma (2.13).

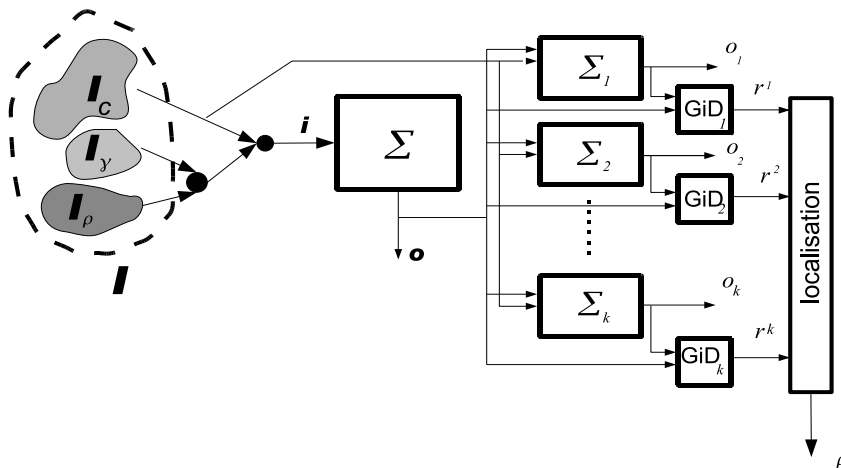


FIG. 2.13 – Banc de machines séquentielles partielles

Les blocs « GiD » représentent les *générateurs d'indicateurs de défaillances* à partir desquels est obtenue la séquence  $r^i$ . L'analyse de l'ensemble des séquences permet de détecter tout événement détectable et de localiser tout événement diagnosticable. Si la décomposition selon les critères (2.19) est possible et que toutes les défaillances sont diagnostiquables, les séquences  $r^i$  sont robustes aux événements de  $I_{\gamma_i}$  et sensibles aux événements de  $I_{\rho_i}$ .

Pour illustrer la méthode de surveillance un exemple illustratif est présenté. La notion de partition est utilisée de manière intuitive. Elle sera définie formellement dans le chapitre suivant à la section 3.2.7.

**Illustration** Considérons un système fictif représenté par la machine séquentielle  $\Sigma$  décrite par la figure 2.14 et la table de transitions d'état (2.6)

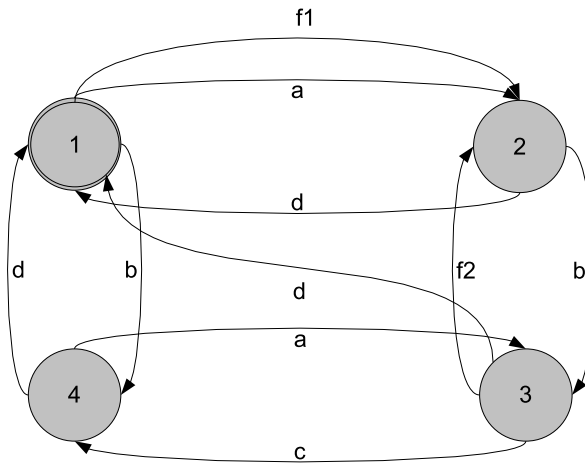


FIG. 2.14 – Graphe d’évolution de la machine séquentielle  $\Sigma$

	a	b	c	d	$f_1$	$f_2$	o
1	2	4	-	-	2	-	1
2	-	3	-	1	-	-	2
3	-	-	4	1	-	2	3
4	3	-	-	1	-	-	4

TAB. 2.6 – Table de transitions d’états de la machine séquentielle  $\Sigma$

Les événements  $a, b, c, d$  sont des événements connus (la commande) et  $f_1$  et  $f_2$  sont des événements inconnus (défaillances ou autres). Pour simplifier, nous considérons que tous les états sont mesurés.

Nous voulons détecter l’apparition d’un événement inconnu  $f_1$  ou  $f_2$  et éventuellement discerner l’événement  $f_1$  de  $f_2$ . Pour la détection de  $f_1$ , nous allons décomposer  $\Sigma$  en deux parties : un sous-modèle découplé de  $f_1$  et un sous-modèle couplé à  $f_1$ . Les sorties du sous-modèle découplé sont robustes par rapport à  $f_1$  alors que celles du sous-modèle couplé y sont sensibles. Les défaillances  $f_1$  et  $f_2$  sont fortement détectables et discernables.

Considérons le cas d’une seule défaillance  $f_1$ , et le problème de la détection de  $f_1$ .

La première étape consiste à déterminer la partie de  $\Sigma$  qui est couplée à  $f_1$ . Il s’agit de trouver la partition de l’ensemble d’état de  $\Sigma$ ,  $S = \{1, 2, 3, 4\}$ , qui ne regroupe pas les états 1 et 2 reliés par  $f_1$  et les ensembles d’entrées et de sorties associés. Les partitions

$$\begin{aligned}
 \pi_1 &= \{\bar{1}, \bar{2}, \bar{3}, \bar{4}\} & \pi_2 &= \{\bar{1}, \bar{2}, \bar{3}, \bar{4}\} \\
 \pi_3 &= \{\bar{1}, \bar{2}, \bar{3}, 4\} & \pi_4 &= \{\bar{1}, \bar{3}, 4, \bar{2}\} \\
 \pi_5 &= \{\bar{1}, \bar{4}, \bar{2}, \bar{3}\} & \pi_6 &= \{\bar{1}, \bar{4}, \bar{2}, \bar{3}\} \\
 \pi_7 &= \{\bar{1}, \bar{2}, \bar{3}, 4\} & \pi_8 &= \{\bar{1}, \bar{3}, \bar{2}, \bar{4}\} \\
 \pi_9 &= \{\bar{1}, \bar{3}, \bar{2}, 4\}
 \end{aligned}$$

remplissent cette condition. Par exemple, la partie décrite par la partition  $\pi_4$ , avec un en-

semble d'états  $S' = \{\overline{1, 3, 4, 2}\}$ , un ensemble d'entrées  $I' = \{a, b, c, d\}$  et un ensemble de sorties  $O' = S'$  est couplée par rapport à  $f_1$ . Le modèle qui correspond à cette partie  $\Sigma'(S', I', O', \delta', \lambda')$  est montré par la figure 2.15. Les relations  $\delta$  et  $\lambda$  sont des restrictions<sup>5</sup> des relations  $\delta$  et  $\lambda$ . (2.1) La description du modèle  $\Sigma'$  montre un non-déterminisme : si  $\Sigma'$  est dans l'état 134 et

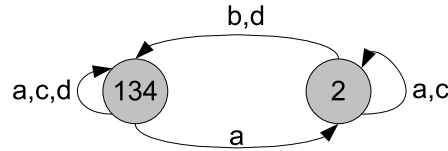


FIG. 2.15 – Représentation de la machine séquentielle partielle  $\Sigma'$

que l'événement  $a$  se produit deux évolutions sont possibles. Cela implique que  $\Sigma'$  n'est pas un sous-modèle de  $\Sigma$  et que la partie  $(S', I', O')$  ne peut pas être réalisée par un sous-modèle de  $\Sigma$ .

Par contre, la partition  $\pi_7 = \{\overline{1, 2, 3, 4}\}$  décrit un modèle  $\Sigma''$  qui est un sous-modèle  $\Sigma$  et dont la synthèse n'entraîne pas de non-déterminisme (figure 2.16).

La détection de la défaillance  $f_1$  s'effectue en comparant les sorties estimées par la machine séquentielle partielle  $\Sigma''$  avec les sorties de  $\Sigma$  (table 2.7).

	a	b	c	a	d	a	$f_1$	b	d	b	d	a	b	$f_2$	d
$\Sigma$	2	3	4	3	1	2	1	1	1	4	1	2	3	2	1
$\Sigma''$	234	234	234	234	1	234	234	1	234	1	234	234	234	1	1

TAB. 2.7 – Évolution parallèle des machines séquentielles  $\Sigma$  et  $\Sigma''$

Une différence de sorties existe si les sorties n'appartiennent pas au même bloc de la partition de l'ensemble de sorties de  $\Sigma$ ; ici 1 et 234 n'appartiennent pas au même bloc. La sortie 234 se lit « la sortie est égale à 2 ou 3 ou 4 ».

Si l'événement  $f_1$  se produit, les sorties de  $\Sigma$  et de  $\Sigma''$  ne sont plus cohérentes et permettent de détecter l'occurrence de  $f_1$ . Remarquons que l'occurrence de  $f_2$  ne produit pas d'écart des sorties, puisque  $\Sigma''$  est découplée de  $f_2$ .

Cet exemple illustre grossièrement le principe de la méthode de surveillance basée en une décomposition du modèle initial avec des critères de couplage et de découplage par rapport aux entrées. La méthodologie de la décomposition est détaillée dans le chapitre 4.

<sup>5</sup>voir chapitre 3 section 3.2.2

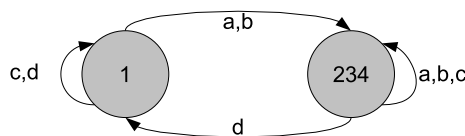


FIG. 2.16 – Schéma représentant la machine séquentielle partielle  $\Sigma''$

## 2.5 Conclusion

### 2.5.1 Synthèse des problèmes à traiter

Dans ce chapitre nous avons effectué un survol des trois aspects qui fixent le cadre de cette thèse : la modélisation des systèmes dynamiques en fonctionnement défaillant, la surveillance des systèmes à base de modèles, et la décomposition de modèles pour la génération d'indicateurs structurés de défaillances . Il existe une multitude de méthodes et d'approches de la surveillance, qui sont choisies en fonction des défauts à détecter, et de la dynamique du processus à surveiller. Si l'objectif est la surveillance de processus complexes et la détection de défauts graves, qui impliquent des variations brusques de paramètres, un choix tout naturel est d'utiliser des techniques de surveillance basées sur un modèle événementiel. Si par contre, on cherche à faire apparaître un défaut de faible amplitude, éventuellement caractérisé par une modification lente des paramètres, une représentation à base de modèles temporels s'impose.

Par ailleurs, en fonction du nombre de capteurs disponibles, le choix se porte sur la redondance analytique (si le nombre de capteurs est insuffisant pour la surveillance), ou sur la redondance matérielle (si le nombre de capteurs permet des mesures redondantes).

Cette diversité de méthodes peut poser certains problèmes de commodité et a pour conséquence le fractionnement de la communauté de surveillance par rapport aux méthodes utilisées.

L'objectif de ce mémoire est de proposer une approche de détection et de localisation de défaillances qui s'abstrait du type de représentation, et de la nature du modèle. Cette approche traite différents systèmes en utilisant des outils similaires, avec un minimum d'adaptation. Pour réaliser cet objectif, deux points seront développés :

- tracer un parallèle et relever les similitudes entre les approches « temporelle » et « événementielle » pour proposer une solution adaptée aux deux problématiques et développer des outils mathématiques adéquats.
- proposer une solution à la structuration des indicateurs de défauts et la réaliser en utilisant les outils mathématiques développés.

Ainsi, deux thèmes vont être traités par ce mémoire.

### 2.5.2 Outils algébriques pour la surveillance à base de modèles temporels et événementiels

Des similarités entre les approches de surveillance des systèmes à dynamique temporelle et des systèmes à dynamique événementielle sont mises en évidence afin de définir une méthodologie de surveillance commune. A cet effet, des outils mathématiques adaptés doivent être choisis. Ces outils, du fait du niveau d'abstraction nécessaire, sont de nature algébrique, et ont été introduits, sous le nom d'algèbre des paires, dans l'ouvrage de Hartmanis et Stearns [HS66] pour la mise sous forme canonique des machines séquentielles. Une extension de ces outils a été présentée par les professeurs A. Shumsky et A.N. Zhirabok dans de nombreux travaux touchant de près ou de loin à la génération de résidus pour les systèmes non linéaires [Shu91, Shu94, Zhi96, Zhi97,

ZP93a, ZS87, ZS93, SZ05]. Ces outils mathématiques ont été introduits sous le nom d'*algèbre des fonctions*. Dans ce mémoire nous nous placerons dans un autre contexte, en proposant une méthode de détection et de localisation de défaillances qui s'étend au cas des systèmes à dynamique événementielle. Cette méthode se base sur l'utilisation de l'algèbre des paires et sur une formulation adéquate de la problématique de la détection et de la localisation de défaillances au cas événementiel.

Un autre aspect est traité, celui de l'implémentation des outils de l'algèbre des fonctions, qui font apparaître des difficultés calculatoires certaines. Des méthodes de calcul sont proposées dans chaque cas : linéaire et non linéaire, continu dans le temps ou discrétisé.

Le dernier aspect traité est celui de l'amélioration des outils existants et des étapes de l'approche proposée dans [Zhi97] :

- des méthodes de calcul sont proposées pour les opérateurs de l'algèbre des fonctions.
- les contraintes d'application de l'approche initiale [Zhi97] sont relaxées en utilisant des techniques d'élimination de variables.

### 2.5.3 La décomposition de modèles pour la surveillance

Une méthode de détection et de localisation de défaillances est proposée, basée sur une décomposition de modèles avec des critères de couplage et de découplage de signaux d'entrée. La particularité de la solution proposée est de solutionner le problème de génération de résidus structurés en deux étapes. En effet, dans la littérature [DI00, HKE99b, JPS03], la synthèse du générateur de résidus repose sur deux contraintes :

- la robustesse aux perturbation et la sensibilité aux défaillances,
- la convergence du résidu.

Ces deux aspects peuvent être traités séparément, en deux étapes distinctes. Ce point est abordé chapitre 5.

# Chapitre 3

## Rappels mathématiques

### 3.1 Introduction

Ce chapitre présente des rappels de notions essentielles de la théorie des ensembles, de l'algèbre des paires et de l'algèbre des fonctions. Ces notions sont utilisées pour aborder la problématique de décomposition de modèles dans le cadre événementiel ainsi que pour présenter l'extension des outils au cas des modèles continus (temporels). Les rappels théoriques sont en grande majorité tirés de [HS66].

L'algèbre des paires a été développée dans une première utilisation pour la conception de circuits électriques et tout particulièrement pour minimiser le coût des circuits intégrés (ou imprimés) et pour la conception de systèmes de faible consommation électrique [FKKS02a, FKKS02b]. Les circuits électroniques sont modélisés sous forme de machines séquentielles et des outils mathématiques permettent de décomposer ses machines en *machines partielles* en suivant certains critères.

Après étude du cadre événementiel, l'extension de l'algèbre des paires au domaine temporel est traitée. Cette extension a été proposée par Shumsky et Zhirabok dans [Shu91, ZS93] sous le nom d'*algèbre des fonctions* (AF). Cet ensemble d'outils permet le partitionnement de l'espace d'état dans lequel est défini un modèle temporel. Ces outils sont présentés et les principales notions de l'algèbre des paires sont étendues au domaine continu.

Dans la dernière partie du chapitre, les outils de l'algèbre des fonctions sont comparés aux outils de la géométrie différentielle<sup>6</sup>.

### 3.2 Terminologie de la théorie des ensembles

#### 3.2.1 Ensembles

**Définition 3.1** *Toute collection d'objets mathématiques est appelée un ensemble.*

Les objets qui appartiennent à un ensemble sont appelés éléments. L'ensemble  $S$  de

---

<sup>6</sup>Les principales notions de la géométrie différentielle sont rappelés dans l'annexe B



tous les éléments possédant la propriété  $\mathcal{P}$  est noté

$$S = \{s \mid s \text{ a la propriété } \mathcal{P}\}$$

**Illustration** Tous les éléments  $k$  modulo  $2\pi$  où  $k \in \mathbb{N}$  forment l'ensemble

$$S = \{i \mid i = 2\pi k, k \in \mathbb{N}\}$$

Si  $s$  est un élément de  $S$ , on note  $s \in S$ .  $T$  est inclus dans  $S$ , ou  $T$  est un sous-ensemble de  $S$ , si et seulement si  $s \in T \Rightarrow s \in S$  et on note  $T \subset S$ .

Différentes définitions associées aux ensembles sont rappelées.

**Définition 3.2** Soient  $S, T, A$  des ensembles quelconques.

- L'ensemble  $S$  est égal à l'ensemble  $T$ , et on note  $S = T$ , si et seulement si  $S \subset T$  et  $T \subset S$ .
- Un ensemble  $S$  est vide, et on note  $S = \emptyset$ , si et seulement s'il ne contient aucun élément.
- L'ensemble des éléments communs aux deux ensembles  $S$  et  $T$ , noté  $S \cap T$ , est défini par

$$S \cap T = \{s \mid s \in S \text{ et } s \in T\}$$

Cette définition peut être étendue à un groupe d'ensembles

$$\bigcap_{\alpha \in A} S_{\alpha} = \{s \mid s \in S_{\alpha}, \alpha \in A\}$$

- L'ensemble des éléments des ensembles  $S$  et  $T$ , noté  $S \cup T$  est défini par

$$S \cup T = \{s \mid s \in S \text{ ou } s \in T\}$$

L'union d'un groupe d'ensembles est définie par

$$\bigcup_{\alpha \in A} S_{\alpha} = \{s \mid s \in S_{\alpha}, \forall \alpha \in A\}$$

- Les ensembles  $S$  et  $T$  sont disjoints, si et seulement si  $S \cap T = \emptyset$ .  
Une famille d'ensembles  $\{S_{\alpha} \mid \alpha \in A\}$  est disjointe, si chaque paire d'ensembles prise séparément est disjointe.

**Définition 3.3** Le produit cartésien  $\times$  d'un groupe d'ensembles  $S_1, S_2, S_3, \dots, S_n$  est l'ensemble de tous les  $n$ -uplets  $(s_1, s_2, \dots, s_n)$ , avec  $s_i \in S_i$ , des éléments des  $n$  ensembles.

$$S_1 \times S_2 \times \dots \times S_n = \times_{1 \leq i \leq n} S_i = \{(s_1, s_2, \dots, s_n) \mid s_i \in S_i\} \quad (3.1)$$

### 3.2.2 Applications

**Définition 3.4** Soit  $S$  et  $T$  deux ensembles non vides. La loi qui associe chaque élément de  $S$  à un élément unique de  $T$  est appelée application de  $S$  dans  $T$

Soit une application  $f$ , telle que

$$\begin{aligned} f : S &\rightarrow T \\ s &\mapsto t = f(s) \end{aligned} \quad (3.2)$$

Pour chaque élément  $s \in S$ , l'élément  $t \in T$  qui lui est associé est appelé l'image de  $s$  par  $f$  et est noté  $f(s)$ .

Les définitions qui vont suivre rappellent les principales propriétés d'une application. Les ensembles  $S$  et  $T$  sont deux ensembles non vides.

**Définition 3.5** Soit  $f$  une application de  $S$  dans  $T$ ,  $f$  est injective si pour tout  $t \in T$  il existe au plus un élément  $s \in S$  tel que  $f(s) = t$ . Tout élément  $t$  de  $T$  admet au plus un antécédent  $s$  par rapport à  $f$ .

$f$  est dite injective si pour tous  $s$  et  $s'$  dans  $S$ ,  $f(s) = f(s')$  implique  $s = s'$ .

**Illustration**  $f : x \mapsto f(x) = x^3$  est injective.  $f : x \mapsto f(x) = \sin(x)$  n'est pas injective.

**Définition 3.6** Une application  $f$  de  $S$  dans  $T$  est surjective si pour tout  $t \in T$  il y a au moins un élément  $s \in S$  tel que  $f(s) = t$ . Tout élément  $t$  de  $T$  admet au moins un antécédent  $s$  par l'application  $f$ .

$f$  est surjective si l'image directe  $f(S)$  est égale à l'ensemble d'arrivée  $T$ .

**Illustration**  $f : x \mapsto f(x) = x^3$  est surjective.  $f : x \mapsto f(x) = x^2$  n'est pas surjective si  $T = \mathbb{R}$ .

**Définition 3.7** Une application  $f$  de  $S$  dans  $T$  est bijective si pour tout  $t \in T$  il existe un et un seul  $s \in S$  tel que  $f(s) = t$ . En outre, tout élément  $t$  de  $T$  admet un unique antécédent  $s$  (par  $f$ ). Une bijection est une application qui est à la fois injective et surjective.

**Illustration**  $f : x \mapsto f(x) = x^3$  est bijective.  $f : x \mapsto f(x) = \cos(x)$  n'est pas bijective.

**Définition 3.8** Soit  $U$  un sous-ensemble de  $S$ . L'application  $g : U \rightarrow T$  est une restriction de  $f : S \rightarrow T$  à  $U$  si  $\forall s \in U : f(s) = g(s)$

**Définition 3.9** Soit  $f : S \rightarrow T$  avec  $S \subseteq U$  et  $T \subseteq V$ . L'application  $g : U \rightarrow V$  est une extension de  $f$  si  $\forall s \in S : f(s) = g(s)$ .

### 3.2.3 Relations

**Définition 3.10** Une relation est un lien qui existe entre les éléments de deux ensembles  $S$  et  $T$ . La relation  $\diamond$  est définie comme le sous-ensemble  $\mathcal{R} \subseteq S \times T$  et notée  $s \diamond t$ .

$$\mathcal{R} = \{(s, t) | s \diamond t\} \quad (3.3)$$

**Illustration** La relation "est le diviseur de" est une relation qui associe les éléments de l'ensemble  $\mathbb{N}$ . Elle définit des sous-ensembles de nombres qui possèdent le même diviseur.

Soit  $S$  un ensemble non vide et  $\diamond$  une relation sur  $S \times S$ . Les principales propriétés d'une relation sont données ci-dessous

**Réflexivité** La relation  $\diamond$  est réflexive si et seulement si tout élément est en relation avec lui-même :

$$\forall s \in S, (s \diamond s)$$

*Illustration* La relation inférieur ou égal «  $\leq$  » sur  $\mathbb{R}$  est réflexive. La relation inférieur «  $<$  » n'est pas réflexive.

**Symétrie** La relation  $\diamond$  est symétrique si et seulement si toutes les relations entre les éléments de  $S$  sont réciproques.

$$\forall (s, t) \in S^2, (s \diamond t) \Leftrightarrow (t \diamond s)$$

*Illustration* La relation d'égalité «  $=$  » sur  $\mathbb{R}$  est symétrique, puisque  $a = b \Leftrightarrow b = a$ . La relation inférieur «  $<$  » n'est pas symétrique :  $a < b$  n'implique pas  $b < a$ .

**Antisymétrie** La relation  $\diamond$  est antisymétrique s'il n'existe pas deux éléments distincts de  $S$  qui sont mis en relation mutuelle par  $\diamond$  :

$$\forall (s, t) \in S^2, [(s \diamond t) \text{ et } (t \diamond s)] \Rightarrow (s = t)$$

*Illustration* La relation « inférieur ou égal »  $\leq$  sur  $\mathbb{N}$  est antisymétrique. La relation "ont un diviseur commun" sur  $\mathbb{N}$  n'est pas antisymétrique.

**Transitivité**  $\diamond$  est transitive si et seulement si deux éléments sont mis en relation par le biais d'un troisième élément :

$$\forall (s, t, u) \in S^3, [(s \diamond t) \text{ et } (t \diamond u)] \Rightarrow (s \diamond u)$$

*Illustration* La relation « inférieur »  $<$  sur  $\mathbb{N}$  est transitive. La relation ont un diviseur commun notée «  $|$  » sur  $\mathbb{N}$  n'est pas transitive :  $(6|3)$  et  $(2|6)$  n'implique pas  $(2|3)$ .

**Définition 3.11** La relation  $\diamond$  est une relation d'équivalence si elle est réflexive, symétrique et transitive.

Chaque relation d'équivalence  $\diamond$  sur  $S$  définit une classe d'équivalence, notée  $B_\diamond$ , pour chaque  $s \in S$

$$B_\diamond(s) = \{t | s \diamond t\} \tag{3.4}$$

*Illustration* La relation "ont le même signe" sur  $\mathbb{R}$  est une relation d'équivalence.

### 3.2.4 Relations d'ordre

Une relation  $\diamond$  dans un ensemble  $S$  est une relation d'ordre si elle permet une comparaison des éléments de  $S$ .

**Définition 3.12** Une relation d'ordre est une relation binaire réflexive, transitive et antisymétrique.

Un ensemble muni d'une relation d'ordre est dit ensemble ordonné ou ordre.

Certaines relations ne remplissent pas toutes ces conditions. Elles permettent néanmoins d'ordonner leur ensemble de définition, de la même manière qu'une relation d'ordre, sauf qu'aucun élément n'est mis en relation avec lui-même.

Ces relations sont appelées relations d'*ordre strict*. L'ordre strict est transitif, antisymétrique et non réflexif. Une relation d'ordre strict est une relation qui est :

- Transitive.
- Fortement antisymétrique : c'est-à-dire que deux éléments, distincts ou non, ne peuvent être en relation mutuelle,

$$\forall (s, t) \in S^2, \quad (s \diamond t) \Rightarrow (t \not\Diamond s)$$

Par analogie, toute relation binaire antisymétrique et transitive est

- un ordre strict : si la relation est non réflexive.
- un ordre large : si la relation est réflexive.

Cet ordre peut être total ou partiel.

**Définition 3.13** Une relation d'ordre  $\diamond$  est totale si et seulement si elle ordonne toutes les paires d'éléments possibles

$$\forall (s, t) \in S^2, (s \diamond t) \text{ ou } (t \diamond s)$$

L'ensemble S est alors dit totalement ordonné. La relation d'ordre  $\diamond$  est un ordre total.

Une relation d'ordre  $\diamond$  est dite partielle si elle n'est pas totale

$$\exists (s, t) \in S^2, (s \not\Diamond t) \text{ et } (t \not\Diamond s)$$

L'ensemble S est alors dit partiellement ordonné.

### 3.2.5 Structures algébriques

#### Structures communes

Une structure algébrique est formée d'un ensemble combiné à une ou plusieurs opérations (appelées également lois de composition), éventuellement complétées par un ordre ou une topologie, le tout satisfaisant un certain nombre d'axiomes. Les structures algébriques de base ne comportent que des opérations internes. Les plus importantes sont les structures de groupe, d'anneau et de corps.

**Le groupe** est un monoïde (c'est-à-dire un ensemble muni d'une opération interne associative et d'un élément neutre) dont tous les éléments sont inversibles. En d'autres termes, un groupe noté  $(\mathcal{G}, \star, e)$  est un ensemble  $\mathcal{G}$  muni d'un élément neutre nécessairement unique  $e$ , et d'une opération interne  $\star$  qui satisfait les axiomes suivants :

- **Identité** :  $\forall x \in \mathcal{G}, x \star e = e \star x = x$ ;
- **Inverse** :  $\forall x \in \mathcal{G}, \exists y \in \mathcal{G}, x \star y = y \star x = e$ ,  $y$  est dit inverse de  $x$  et noté  $x^{-1}$  ;
- **Associativité** :  $\forall x, y, z \in \mathcal{G}, x \star (y \star z) = (x \star y) \star z$

Lorsque  $\mathcal{G}$  est un ensemble fini,  $(\mathcal{G}, \star, e)$  est un groupe fini ; sinon  $(\mathcal{G}, \star, e)$  est un groupe infini. Pour un groupe fini, l'ordre de ce groupe est le nombre de ses éléments.

**L'anneau** est un ensemble muni d'une structure de groupe (l'opération est l'addition) et d'une opération de multiplication, la multiplication étant distributive sur l'addition. Un anneau est unitaire (resp. intègre<sup>7</sup>, commutatif) si la multiplication est unifière (resp. intègre, commutative) et l'ensemble possède un élément neutre pour la multiplication (dans le cas contraire, il s'agit d'un pseudo-anneau).

**Le corps** est un anneau où l'élément neutre de l'addition n'est pas celui de la multiplication et où tout élément non nul a un inverse multiplicatif. Le corps est :

- un *corps commutatif* pour un corps effectivement commutatif,
- un *corps quelconque* pour un corps non nécessairement commutatif.

Une autre structure algébrique commune importante est l'**espace vectoriel**. Soit  $\mathbb{K}$  un corps. On appelle  $\mathbb{K}$ -*espace vectoriel* tout triplet  $(E, +, \cdot)$  où  $E$  est un ensemble, « + » une opération interne sur  $E$  et «  $\cdot$  » une opération externe sur  $E$  à scalaire dans  $\mathbb{K}$  telle que

- L'élément unité « 1 » du corps  $\mathbb{K}$  est neutre à gauche pour la loi «  $\cdot$  » :

$$\forall u \in E, \quad 1 \cdot u = u$$

- L'opération «  $\cdot$  » est distributive à gauche par rapport à l'addition « + » dans  $E$  :

$$\forall \lambda \in \mathbb{K}, \forall u \in E, \forall v \in E, \quad \lambda \cdot (u + v) = (\lambda \cdot u) + (\lambda \cdot v)$$

- L'opération «  $\cdot$  » est distributive à droite par rapport à l'addition du corps  $\mathbb{K}$  :

$$\forall \lambda \in \mathbb{K}, \forall \mu \in \mathbb{K}, \forall u \in E, \quad (\lambda + \mu) \cdot u = (\lambda \cdot u) + (\mu \cdot u)$$

- L'opération «  $\cdot$  » est associative par rapport à la multiplication du corps  $\mathbb{K}$  (elle *l'importe* dans l'espace vectoriel) :

$$\forall \lambda \in \mathbb{K}, \forall \mu \in \mathbb{K}, \forall u \in E, \quad (\lambda \cdot \mu) \cdot u = \lambda \cdot (\mu \cdot u)$$

Le *module* sur un anneau unitaire est une généralisation de l'espace vectoriel. Dans un espace vectoriel l'ensemble des scalaires forment un corps tandis que dans un module, ils sont munis d'une structure d'anneau (non nécessairement commutatif).

## Algèbre

La structure algébrique qui associe le corps et l'espace vectoriel est nommée *algèbre*.

Soient  $\mathbb{K}$  un corps et  $(E, +, \cdot)$  un espace vectoriel sur  $\mathbb{K}$  associé à l'opération binaire<sup>8</sup> «  $\times$  ». Si «  $\times$  » est *bilinéaire*<sup>9</sup> alors  $(E, \mathbb{K}, +, \cdot, \times)$  est une algèbre sur  $\mathbb{K}$  et on dit que  $E$  est

<sup>7</sup>Un anneau est intègre lorsque aucun des éléments n'est diviseur de l'élément nul

<sup>8</sup>c'est-à-dire  $\forall x, y \in E$ , le résultat  $x \times y \in E$  est le « produit » de  $x$  et  $y$

<sup>9</sup>Soit  $\forall x, y, z \in E$  et  $\forall a, b \in \mathbb{K}$ , l'opération «  $\times$  » est bilinéaire si les propriétés suivantes sont vérifiées :

- distributivité et associativité par rapport à « + » :  $(x + y) \times z = x \times z + y \times z$  et  $x \times (y + z) = x \times y + x \times z$
- «  $\times$  » possède un élément neutre dans  $E$
- $(ax) \times (by) = (ab)(x \times y)$

une  $\mathbb{K}$ -algèbre et  $\mathbb{K}$  est la base de  $E$ . L'opérateur binaire  $\times$  est souvent la multiplication dans  $E$ .

**Remarque 3.2.1** *Si  $\mathbb{K}$  n'est pas un corps mais un anneau commutatif,  $E$  et  $\mathbb{K}$  forment un module. Dans ce cas aussi,  $E$  est une  $\mathbb{K}$ -algèbre et  $\mathbb{K}$  est l'anneau de base de  $E$ . Cette structure s'appelle algèbre sur un anneau et combine les structures de module et d'anneau commutatif.*

### 3.2.6 Treillis et ensembles ordonnés

Soit  $S$  un ensemble ordonné par la relation d'ordre «  $\leq$  ». La paire  $(S, \leq)$  définit un treillis, noté  $L$ .

$$L = (S, \leq)$$

Un treillis est un ordre total ou partiel, qui permet d'ordonner chaque élément de  $S$  par rapport à n'importe quel autre élément de  $S$ . Un treillis peut être défini de manière similaire par :

**THÉORÈME 3.1** *Un ensemble  $S$  muni de deux lois internes notées  $\vee$  et  $\wedge$  qui vérifient les propriétés suivantes*

**Commutativité**  $\forall (s, t) \in S^2, (s \vee t) = (t \vee s)$  et  $(s \wedge t) = (t \wedge s)$

**Associativité**  $\forall (s, t, u) \in S^3, s \vee (t \vee u) = (s \vee t) \vee u$  et  $s \wedge (t \wedge u) = (s \wedge t) \wedge u$

**Idempotence**  $\forall s \in S, (s \vee s) = s$  et  $(s \wedge s) = s$

**Absorbance**  $\forall (s, t) \in S^2, s \wedge (s \vee t) = s$  et  $s \vee (s \wedge t) = s$

forme un treillis  $L = (S, \vee, \wedge)$ .

**Preuve** *Soit la relation  $\leq$  telle que*

$$(s \leq t) \Leftrightarrow (s \vee t) = t$$

*Cette relation est une relation d'ordre (total ou partiel) puisque*

- *La propriété d'associativité assure la transitivité.*
- *La propriété d'idempotence assure la réflexivité.*
- *L'antisymétrie est vérifiée par la définition même.*

*Par ailleurs, en utilisant la propriété d'idempotence, il est montré que*

$$(s \leq t) \Leftrightarrow (s \wedge t) = s$$

*On vérifie que  $(s \vee t)$  donne pour résultat l'élément supérieur parmi  $s$  et  $t$  et que  $(s \wedge t)$  donne pour résultat l'élément inférieur, ce qui assure que  $(S, \leq)$  est bien un treillis.*

Par conséquent, pour n'importe quel paire d'éléments du treillis, il existe un élément maximal et un élément minimal. Ce résultat est formalisé dans le théorème ci-dessous

**THÉORÈME 3.2** Un ensemble  $S$  muni d'une relation d'ordre  $\leq$  qui vérifie que pour tout élément  $s$  et  $t$  de  $S$ , il existe une borne supérieure et une borne inférieure à la paire  $(s, t)$ , forme un treillis  $L = (S, \leq)$ .

**Preuve** Soient les deux opérations internes à  $S$  suivantes

- $s \vee t = \max(s, t)$
- $s \wedge t = \min(s, t)$

Les deux lois sont commutatives. Elles sont également associatives. L'idempotence et l'absorbance peuvent être démontrées également.  $(S, \leq)$  définit bien un treillis

Donc, à partir d'un treillis muni d'une relation d'ordre, il est possible de construire deux lois qui remplissent les conditions du théorème 3.1.

Le treillis est représenté habituellement par un diagramme de Hasse. Le haut du diagramme est occupé par le plus grand élément noté  $\mathbb{I}$  et le bas par le plus petit élément par rapport à la relation d'ordre associée au treillis, que l'on note  $\mathbb{O}$ . Tous les éléments sont représentés par des points, et disposés selon leur ordre de bas en haut, en commençant par  $\mathbb{O}$ . Tous les éléments de même ordre occupent une même ligne. Chaque relation entre deux éléments adjacents est représentée par une droite.

**Illustration** Les sous-ensembles de l'ensemble  $S = \{1, 2, 3\}$

$$S^* = \{\emptyset, \{1\}, \{2\}, \{3\}, \{2, 3\}, \{1, 3\}, \{1, 2\}, \{1, 2, 3\}\}$$

munis de la relation «  $\subseteq$  » forment un treillis  $(S^*, \subseteq)$  (figure 3.1)

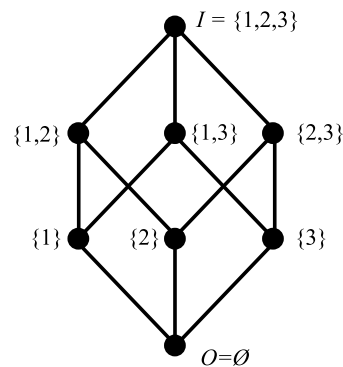


FIG. 3.1 – Treillis formé par les sous-ensembles de  $\{1, 2, 3\}$  associés à «  $\subseteq$  »

### 3.2.7 Partitions

Une partition d'un ensemble non vide  $S$  est un ensemble  $\pi$  de sous-ensembles non vides de  $S$  deux à deux disjoints et qui forment un recouvrement de  $S$ . Les sous-ensembles de  $\pi$  sont désignés comme *blocs* de  $\pi$  et notés  $B_\pi(s)$  pour un élément  $s \in S$  spécifique. L'ensemble de tous les blocs  $B_\pi$  décrit le partitionnement de  $S$  par  $\pi$  (3.5).

$$\pi = \{B_{\alpha_i} \mid B_{\alpha_i} \cap B_{\alpha_j} = \emptyset \text{ si } i \neq j \text{ et } \bigcup_{i=1}^n B_{\alpha_i} = S\} \quad (3.5)$$

avec  $i, j = 1, 2, \dots, n$ .

La notation  $s \equiv t(\pi)$  signifie que les éléments  $s$  et  $t$  appartiennent au même bloc de  $\pi$ . Par ailleurs, nous représentons le bloc d'une partition par l'ensemble surligné des éléments contenus dans ce bloc.

**Illustration** Soit  $S = \{1, 2, 3, 4\}$  un ensemble de 4 éléments. Les partitions  $\pi_1$  et  $\pi_2$  de trois blocs telles que

$$\begin{array}{l} \pi_1 : \quad B_{\pi_1}(1) = 1, 2 \quad B_{\pi_1}(2) = 1, 2 \quad B_{\pi_1}(3) = 3 \quad B_{\pi_1}(4) = 4 \\ \pi_2 : \quad B_{\pi_2}(1) = 1 \quad B_{\pi_2}(2) = 2 \quad B_{\pi_2}(3) = 3, 4 \quad B_{\pi_2}(4) = 3, 4 \end{array}$$

sont notées  $\pi_1 = \{\overline{1, 2}, \overline{3}, \overline{4}\}$  et  $\pi_2 = \{\overline{1}, \overline{2}, \overline{3, 4}\}$

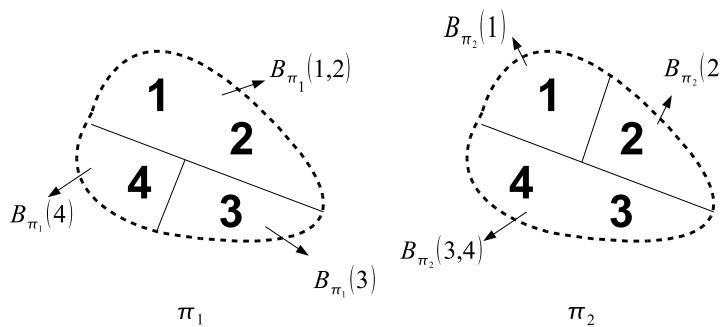


FIG. 3.2 – Graphe représentant les partitions  $\pi_1$  (gauche) et  $\pi_2$  (droite)

### Relations et opérations sur les partitions

Soient  $\pi_1$  et  $\pi_2$  deux partitions de l'ensemble non vide  $S$ . Les opérations suivantes, définies sur l'ensemble de toutes les partitions de  $S$  permettent de manipuler deux partitions quelconques.

**Multiplication de partitions** La multiplication des deux partitions  $\pi_1$  et  $\pi_2$  d'un ensemble  $S$ , notée «  $\pi_1 \cdot \pi_2$  », est la partition de l'ensemble  $S$  qui satisfait

$$s \equiv t(\pi_1 \cdot \pi_2) \Leftrightarrow s \equiv t(\pi_1) \text{ et } s \equiv t(\pi_2)$$

Les blocs de la partition  $\pi_1 \cdot \pi_2$  sont déterminés de la manière suivante

$$B_{\pi_1 \cdot \pi_2}(s) = B_{\pi_1}(s) \cap B_{\pi_2}(s) \tag{3.6}$$

La multiplication de plusieurs partitions est notée

$$\pi_1 \cdot \pi_2 \cdots \pi_n = \prod_{i=1}^n \pi_i$$



**Illustration** Soit les deux partitions sur l'ensemble  $S = \{0, 1, 2, 3, 4, 5, 6\}$  telles que

$$\begin{aligned}\pi_1 &= \{\overline{0, 1, 2}, \overline{3, 4, 5}, \overline{6}\} \\ \pi_2 &= \{\overline{0}, \overline{1, 2, 3}, \overline{4, 5}, \overline{6}\}\end{aligned}$$

La multiplication de  $\pi_1$  et  $\pi_2$  donne

$$\pi_1 \cdot \pi_2 = \{\overline{0}, \overline{1, 2}, \overline{3}, \overline{4}, \overline{5}, \overline{6}\}$$

Ce résultat est obtenu comme suit

$$\begin{aligned}B_{\pi_1 \pi_2}(0) &= B_{\pi_1}(0) \cap B_{\pi_2}(0) = \{0, 1, 2\} \cap \{0\} = \{0\} \\ B_{\pi_1 \pi_2}(1) &= B_{\pi_1}(1) \cap B_{\pi_2}(1) = \{0, 1, 2\} \cap \{1, 2, 3\} = \{1, 2\} \\ B_{\pi_1 \pi_2}(2) &= B_{\pi_1}(2) \cap B_{\pi_2}(2) = \{0, 1, 2\} \cap \{1, 2, 3\} = \{1, 2\} \\ B_{\pi_1 \pi_2}(3) &= B_{\pi_1}(3) \cap B_{\pi_2}(3) = \{3\} \cap \{3\} = \{3\} \\ B_{\pi_1 \pi_2}(4) &= B_{\pi_1}(4) \cap B_{\pi_2}(4) = \{4, 5\} \cap \{4\} = \{4\} \\ B_{\pi_1 \pi_2}(5) &= B_{\pi_1}(5) \cap B_{\pi_2}(5) = \{4, 5\} \cap \{5\} = \{5\} \\ B_{\pi_1 \pi_2}(6) &= B_{\pi_1}(6) \cap B_{\pi_2}(6) = \{6\} \cap \{6\} = \{6\}\end{aligned}$$

**Addition de partitions** L'addition de deux partitions  $\pi_1$  et  $\pi_2$  d'un ensemble  $S$ , notée «  $\pi_1 + \pi_2$  », est la partition de l'ensemble  $S$  qui satisfait :

$$s \equiv t(\pi_1 + \pi_2) \Leftrightarrow \exists s_0, s_1, \dots, s_n : s_i \equiv s_{i+1}(\pi_1) \text{ ou } s_i \equiv s_{i+1}(\pi_2)$$

avec  $0 \leq i \leq n-1$  et  $s_0 = s, s_n = t$ .

L'addition de plusieurs partitions est notée

$$\pi_1 + \pi_2 + \dots + \pi_n = \sum_{i=1}^n \pi_i$$

Les blocs de la partition  $\pi_1 + \pi_2$  sont déterminés par l'itération suivante : pour un élément  $s$ , on pose

$$B_1(s) = B_{\pi_1}(s) \cup B_{\pi_2}(s)$$

Pour chaque itération  $i > 1$  la relation suivante est évaluée

$$B_{i+1}(s) = B_i(s) \cup \{B \mid (B \in \pi_1 \text{ ou } B \in \pi_2) \text{ et } B \cap B_i(s) \neq \emptyset\} \quad (3.7)$$

Tous les blocs de la partition  $\pi_1 + \pi_2$  remplissent la condition

$$B_{\pi_1 + \pi_2}(s) = \{B_i(s) \mid \forall i : B_{i+1}(s) = B_i(s)\} \quad (3.8)$$

Les relations (3.7) et (3.8) sont employés de la manière suivante : Il s'agit de grouper tous les blocs des deux partitions qui présentent un élément commun.

**Illustration** Prenons les mêmes partitions  $\pi_1$  et  $\pi_2$  de l'exemple précédent . L'addition des deux partitions donne

$$\pi_1 + \pi_2 = \overline{\{0, 1, 2, 3, 4, 5, 6\}}$$

Ce résultat est obtenu comme suit

$$\begin{aligned} B_{\pi_1+\pi_2}(0) &= B_{\pi_1}(0, 1, 2) \cup B_{\pi_2}(0) \cup B_{\pi_2}(1, 2, 3) \cup B_{\pi_1}(3) \\ &= \{0, 1, 2\} \cup \{1, 2, 3\} \cup \{3\} \cup \{0\} = \{0, 1, 2, 3\} = B_{\pi_1+\pi_2}(0, 1, 2, 3) \\ B_{\pi_1+\pi_2}(1) &= B_{\pi_1+\pi_2}(0, 1, 2, 3) \\ B_{\pi_1+\pi_2}(2) &= B_{\pi_1+\pi_2}(0, 1, 2, 3) \\ B_{\pi_1+\pi_2}(3) &= B_{\pi_1+\pi_2}(0, 1, 2, 3) \\ B_{\pi_1+\pi_2}(4) &= B_{\pi_1}(4, 5) \cup B_{\pi_2}(4) \cup B_{\pi_2}(5) = \{4, 5\} \cup \{4\} \cup \{5\} = \{4, 5\} = B_{\pi_1+\pi_2}(4, 5) \\ B_{\pi_1+\pi_2}(5) &= B_{\pi_1+\pi_2}(4, 5) \\ B_{\pi_1+\pi_2}(6) &= B_{\pi_1}(6) \cup B_{\pi_2}(6) = \{6\} \cup \{6\} = \{6\} \end{aligned}$$

**Relation d'ordre partiel** La partition  $\pi_1$  est dite inférieure ou égale à  $\pi_2$  et est notée

$$\pi_1 \leq \pi_2 \tag{3.9}$$

si et seulement si chaque bloc de  $\pi_1$  est contenu dans un bloc de  $\pi_2$ .  
 $\pi_1$  est une partition plus *fine* que  $\pi_2$ .

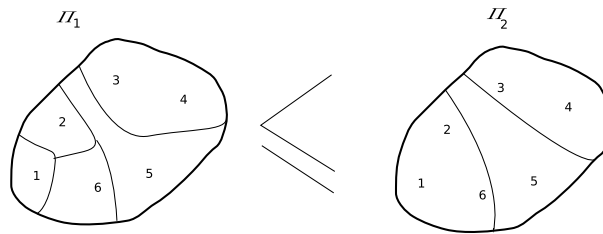


FIG. 3.3 – Diagramme représentant la relation d'ordre  $\pi_1 \leq \pi_2$

### Treillis de partitions

Considérons les deux partitions  $\pi_1$  et  $\pi_2$ . Si la relation  $\pi_1 \leq \pi_2$  est vraie alors

$$\pi_1 \leq \pi_2 \Leftrightarrow \begin{cases} \pi_1 \cdot \pi_2 = \pi_1 \\ \pi_1 + \pi_2 = \pi_2 \end{cases}$$

D'après la définition du treillis vue précédemment, les partitions d'un ensemble quelconque  $S$  forment un treillis muni de l'ordre partiel  $\leq$  et des opérations « + » et « . ». La partition minimale  $\mathbb{O}$  est la partition dont le nombre de blocs sera égal au nombre d'éléments de  $S$  et chaque bloc contient un élément unique. La partition maximale  $\mathbb{I}$  est formée d'un seul bloc contenant tous les éléments de  $S$ .

### 3.3 L'algèbre des paires

#### 3.3.1 Introduction

Cette section est dédiée à la description de l'algèbre des paires, introduite par [HS66]. L'algèbre des paires est une structure algébrique définie par rapport à l'ensemble des partitions de deux ensembles  $S$  et  $I$  et d'une application  $\delta$  définie par  $\delta : S \times I \rightarrow S$ . Une algèbre est définie par un ensemble muni d'opérations. Les éléments caractéristiques de l'algèbre des paires sont présentés dans ce qui suit avec le rappel d'une propriété importante, qui est la propriété de substitution de partitions.

#### 3.3.2 Propriété de substitution

La propriété de substitution est une notion essentielle pour définir l'algèbre des paires. Elle décrit la possibilité de remplacer une partition d'un ensemble par une partition qui lui est équivalente. Cette équivalence est définie par rapport à une application donnée. La restriction de cette application sur la nouvelle partition est équivalente à l'application initiale, si la partition remplaçante possède la propriété de substitution.

**Définition 3.14** La partition  $\pi$  d'un ensemble  $S$  possède la propriété de substitution par rapport à une application  $\delta : S \times I \rightarrow S$ , si et seulement si l'image de chaque bloc de  $\pi$  par  $\delta$  est contenue dans un autre bloc de  $\pi$ .

$$s \equiv t(\pi) \Rightarrow \forall a \in I, \delta(s, a) \equiv \delta(t, a)(\pi) \tag{3.10}$$

Les blocs  $B_\pi$  forment les éléments d'un nouvel ensemble  $\pi$  qui se substitue à  $S$  et pour lequel il existe une restriction de  $\delta$ .

**Définition 3.15** Soit  $\pi$  une partition des éléments de  $S$  qui possède la propriété de substitution. L'application  $\delta_\pi : \pi \times I \rightarrow \pi$  telle que

$$\delta_\pi(B_\pi, i) = B'_\pi \Leftrightarrow \delta_\pi(B_\pi, i) \subseteq B'_\pi$$

où  $i \in I$ , est appelée l'image selon  $\pi$  de  $\delta$ .

L'application  $\delta_\pi$  est une restriction de  $\delta$ .

**Illustration** Considérons l'application  $\delta$  décrite par la table (3.11)

$\delta$				
	$a$	$b$		
1	5	3		
2	1	5		
3	4	6		
4	6	2		
5	3	4		
6	2	1		

$\delta_\pi$		
$1' = 1, 3, 6$	$a$	$b$
$2' = 2, 4, 5$	$2'$	$1'$
	$1'$	$2'$

(3.11)

avec  $S = \{1, 2, 3, 4, 5, 6\}$  et  $I = \{a, b\}$ . La partition  $\pi = \{\overline{1, 3, 6}, \overline{2, 4, 5}\}$  de  $S$  est une partition avec propriété de substitution. L'application image  $\delta_\pi$  est décrite par la table (3.11).

D'autres partitions avec propriété de substitution existent sur  $S$ , par exemple

$$\pi' = \{\overline{1, 4}, \overline{2, 3}, \overline{5, 6}\}$$

Pour générer de nouvelles partitions possédant la propriété de substitution à partir d'une partition avec propriété de substitution connue, il est nécessaire d'utiliser le théorème [HS66] suivant

**THÉORÈME 3.3** *Si  $\pi_1, \pi_2$ , deux partitions de  $S$ , possèdent la propriété de substitution, alors les partitions  $\pi_1.\pi_2$  et  $\pi_1 + \pi_2$  possèdent également la propriété de substitution.*

Ce qui veut dire qu'en effectuant toutes les additions et multiplications possibles de deux partitions possédant la propriété de substitution, de nouvelles partitions avec la propriété de substitution sont obtenues.

Par ailleurs, l'ensemble des partitions avec propriété de substitution est un ensemble ordonné. Le théorème qui suit permet de structurer les partitions en treillis [HS66].

**THÉORÈME 3.4** *L'ensemble de toutes les partitions de  $S$  avec propriété de substitution par rapport à une application  $\delta$ , forme un treillis  $L$  défini par*

- $L : (S, \leq)$ , où  $\leq$  est l'ordre partiel des partitions,
- $L' : (S, +, .)$ , où « + » et « . » sont les opérations d'addition et de multiplication définies sur l'ensemble des partitions.

$L$  inclut les partitions triviales  $\mathbb{I}$  et  $\mathbb{O}$

### 3.3.3 Paires algébriques et opérateurs de l'algèbre des paires

Une paire algébrique, ou paire de partitions, est formée par deux partitions de  $S$  qui possèdent la propriété suivante : les blocs de la première partition sont les antécédents, par rapport à une application  $\delta$  donnée, des blocs de la seconde partition.

**Définition 3.16** *Une paire algébrique par rapport à une application  $\delta$ , notée  $(\pi, \pi')$ , est un binôme ordonné de partitions sur  $S$  tel que*

$$\forall i \in I : s \equiv t(\pi) \Rightarrow \delta(s, i) \equiv \delta(t, i)(\pi')$$

Si  $\pi$  et  $\pi'$  forment une paire algébrique, alors tous les blocs de  $\pi$  seront projetés vers des blocs de  $\pi'$  par  $\delta$ .

$$\forall x \in I, \forall B_\pi \in \pi, \exists B_{\pi'} : \delta(B_\pi, x) \subseteq B_{\pi'}$$

La paire algébrique est anti-symétrique. Si  $(\pi, \pi')$  et  $(\pi', \pi)$  forment des paires alors forcément  $\pi \equiv \pi'$ . Nous définissons alors un membre de gauche et un membre de droite dans  $(\pi, \pi')$  :  $\pi'$  forme une paire à gauche avec  $\pi$  et  $\pi$  forme une paire à droite avec  $\pi'$ .

**Remarque 3.3.1** La propriété de substitution est liée à la notion de « paire algébrique ». Si  $\pi$  est une partition avec propriété de substitution par rapport à  $\delta$ , alors  $(\pi, \pi)$  forment une paire algébrique.

**Illustration** Considérons l'exemple précédent et l'application  $\delta$  décrite par la table (3.11). La partition

$$\pi_1 = \{\overline{1, 2}, \overline{3, 4}, \overline{5, 6}\}$$

forme une paire algébrique avec la partition  $\{\overline{1, 3, 5}, \overline{2, 4}, \overline{6}\}$  puisque l'image des états 1 et 2 est le bloc  $\{1, 3, 5\}$ , mais l'inverse n'est pas vrai, puisque l'image du bloc  $\{1, 3, 5\}$  est le bloc  $\{2, 3, 4, 5, 6\}$  qui n'appartient pas à  $\pi_1$ .

Par contre, la partition

$$\pi_3 = \{\overline{1, 4}, \overline{2, 3}, \overline{5, 6}\}$$

possède la propriété de substitution. Par conséquent, elle forme une paire algébrique avec elle-même :  $(\pi_3, \pi_3)$  puisque chacun des blocs de  $\pi_3$  est projeté par  $\delta$  vers un autre bloc de  $\pi_3$ .

Considérons maintenant l'ensemble de toutes les partitions  $\pi'$  qui forment une paire à droite avec une partition  $\pi$  donnée, c'est-à-dire  $(\pi, \pi')$  forme une paire algébrique. Cet ensemble est borné par la plus petite partition possible des  $\pi'$  (à droite). Par analogie, l'ensemble des partitions  $\pi$  qui forment une paire algébrique (à gauche) avec  $\pi'$  est borné par la plus grande partition de l'ensemble des  $\pi$ . Ces deux extremums sont déterminées par les relations

$$\pi' = \prod \{\pi_i | (\pi, \pi_i) \text{ est une paire algébrique sur } \Sigma\} \quad (3.12)$$

$$\pi = \sum \{\pi_i | (\pi_i, \pi') \text{ est une paire algébrique sur } \Sigma\} \quad (3.13)$$

Les notions du *plus grand* et du *plus petit* correspondent respectivement à la partition la moins fine et à la partition la plus fine. La partition la plus petite est notée  $m$  (pour *minimale*) et la partition la plus grande est notée  $M$  (pour *maximale*). Les opérateurs  $m$  et  $M$  sont définis ci-dessous.

**Définition 3.17** Soit  $\pi$  une partition de  $S$ .  $m(\pi)$  est la partition la plus fine (minimale) qui forme une paire à droite avec  $\pi$

$$m(\pi) = \prod \{\pi_i | (\pi, \pi_i) \text{ forme une paire algébrique}\}$$

**Définition 3.18** Soit  $\pi$  une partition de  $S$ .  $M(\pi)$  est la partition la moins fine (maximale) qui forme une paire à gauche avec  $\pi$

$$M(\pi) = \sum \{\pi_i | (\pi_i, \pi) \text{ forme une paire algébrique}\}$$

**Remarque 3.3.2** Une partition plus fine permet d'avoir le plus de distinction entre les éléments d'un ensemble. Par contre, une partition grossière amène des blocs qui regroupent plusieurs éléments. Dans la partition  $\{\overline{1, 2}, \overline{3}\}$  par exemple, il n'est pas possible de distinguer

les éléments 1 et 2. Par contre, la partition  $\{\bar{1}, \bar{2}, \bar{3}\}$  permet de distinguer chaque élément. L'exemple des deux cas extrêmes est encore plus parlant : La partition maximale  $\mathbb{I}$  et la partition minimale  $\mathbb{O}$  sur un ensemble quelconque  $S = \{s_1, s_2, \dots, s_n\}$ .

$$\mathbb{I} = \{\overline{s_1, s_2, \dots, s_n}\} \quad \mathbb{O} = \{\overline{s_1}, \overline{s_2}, \dots, \overline{s_n}\}$$

La partition  $\mathbb{I}$  est composée d'un seul bloc de  $n$  éléments. C'est la partition maximale sur  $S$ ; il n'est possible de distinguer aucun élément. A l'opposé,  $\mathbb{O}$ , qui est composée de  $n$  blocs unitaires, est la plus petite partition possible : la connaissance d'un bloc implique directement la connaissance d'un élément.

Soit  $\pi$  une partition quelconque de  $S$ . Les relations suivantes sont toujours vérifiées

$$\begin{aligned} \forall \pi : (\pi, \mathbb{I}) \text{ est une paire algébrique} \\ \forall \pi : (\mathbb{O}, \pi) \text{ est une paire algébrique} \end{aligned} \quad (3.14)$$

Ces deux paires algébriques s'interprètent comme suit :

- N'importe quel bloc de n'importe quelle partition possède forcément une image via l'application  $\delta$ .
- N'importe quel bloc de n'importe quelle partition possède forcément un antécédent via  $\delta$

### 3.3.4 Structure algébrique et définitions de l'algèbre des paires

L'ensemble des paires algébriques, muni des opérations d'addition et de multiplication forme une algèbre constituée de paires algébriques. C'est l'algèbre des paires.

**Définition 3.19** Soit  $L_1$  et  $L_2$  deux treillis. Un sous-ensemble de  $L_1 \times L_2$  muni des opérations « + » et « . », est une algèbre des paires sur  $L_1 \times L_2$ , notée  $\Delta$ , si les relations suivantes sont vérifiées :

1.  $\forall x_1, x_2 \in L_1 \quad \forall y_1, y_2 \in L_2 : (x_1, y_1) \in \Delta \text{ et } (x_2, y_2) \in \Delta \Rightarrow (x_1.x_2, y_1.y_2) \in \Delta \text{ et } (x_1 + x_2, y_1 + y_2) \in \Delta$
2.  $\forall x \in L_1, \forall y \in L_2 : (x, \mathbb{I}) \in \Delta \text{ et } (\mathbb{O}, y) \in \Delta$

Pour un ensemble  $S$  donné, les propriétés (3.14) permettent de déduire que toutes les paires algébriques sur  $S$  forment une algèbre des paires sur  $L \times L$ , où  $L$  est le treillis des partitions de  $S$ .

**Remarque 3.3.3** Pour une algèbre des paires donnée  $\Delta$ , la notation  $(\pi_1, \pi_2) \in \Delta$  veut dire que  $\pi_1$  et  $\pi_2$  forment une paire algébrique .

**Lemme 3.4.1** Soit  $\Delta$  une algèbre des paires sur  $L_1 \times L_2$ . Si  $(x, y) \in \Delta$  alors

$$x' \leq x \text{ et } y \leq y' \Rightarrow (x', y) \in \Delta \text{ et } (x, y') \in \Delta \text{ et } (x', y') \in \Delta$$

Cela veut dire qu'à partir d'une paire algébrique donnée il est possible d'obtenir d'autres paires algébriques en utilisant une partition plus petite (plus fine) à gauche et une partition plus grande (moins fine) à droite. Ce lemme permet de reformuler les définitions 3.17 et 3.18.

**Définition 3.20** Soit  $\Delta$  une algèbre des paires sur  $L_1 \times L_2$ . Soit  $x \in L_1$  et  $y \in L_2$  deux partitions. Les opérateurs  $m$  et  $M$  sont déterminés par

$$m(x) = \prod \{y_i | (x, y_i) \in \Delta\} \quad (3.15)$$

$$M(y) = \sum \{x_i | (x_i, y) \in \Delta\} \quad (3.16)$$

Ces opérateurs sont minimaux (ou maximaux) par rapport à la relation d'ordre des partitions (3.9), puisque la multiplication répétée des partitions tend à affiner le résultat donné par  $m$  et l'addition tend à faire l'inverse pour le résultat donné par  $M$ . Une hiérarchie des paires existe, et une relation d'ordre sur l'ensemble des paires algébriques peut être définie.

**Définition 3.21** Soit  $(x, y)$  et  $(x', y') \in L_1 \times L_2$ .  $(x, y) \leq (x', y')$  si et seulement si  $x \leq x'$  dans  $L_1$  et  $y \leq y'$  dans  $L_2$

Une algèbre  $\Delta$  est donc elle-même un treillis sur l'ensemble des paires algébriques munie de la relation d'ordre partiel  $\leq$ .

Pour clore cette section, quelques propriétés essentielles des opérateurs  $m$  et  $M$  sont données. Ces propriétés se déduisent directement des définitions, lemmes et théorèmes précédemment énoncés.

- $(M(y), y)$  et  $(x, m(x))$  forment des paires algébriques,
- si  $y \geq m(x)$  ou  $x \leq M(y)$  alors  $(x, y)$  forment une paire algébrique,

où  $x \in L_1$  et  $y \in L_2$ .

Les opérateurs  $m$  et  $M$  sont distributifs par rapport aux opérations « + » et « . » :

- $m(x_1 + x_2) = m(x_1) + m(x_2)$
- $m(x_1 \cdot x_2) \leq m(x_1) \cdot m(x_2)$
- $M(y_1 + y_2) \geq M(y_1) + M(y_2)$
- $M(y_1 \cdot y_2) \geq M(y_1) \cdot M(y_2)$

avec  $x_i \in L_1$  et  $y_i \in L_2$ .

Les opérateurs  $m$  et  $M$  sont également monotones

- si  $x_1 \leq x_2$  alors  $m(x_1) \leq m(x_2)$
- si  $y_1 \geq y_2$  alors  $M(y_1) \geq M(y_2)$

et non reversifs

- $M(m(x)) \geq x$
- $m(M(y)) \leq y$

### 3.3.5 Calcul des opérateurs $m$ et $M$

#### a. Détermination de l'opérateur $m$

Soit une application  $\delta$  définie par

$$\begin{aligned}\delta : S &\rightarrow S \\ s &\mapsto s^+ = \delta(s)\end{aligned}$$

et  $\pi$  une partition de  $S$ . Le calcul de  $m(\pi)$  s'effectue selon les étapes suivantes :

1. La partition  $\pi$  est décomposée en partitions  $\pi_{i,j}$  où  $i$  et  $j$  sont des éléments de  $S$ . La partition  $\pi_{i,j}$  est une partition qui regroupe dans un seul bloc les éléments  $i$  et  $j$  et dont le reste des blocs contient un seul élément. L'addition de l'ensemble des  $\pi_{i,j}$  a pour résultat la partition  $\pi$  :

$$\sum \pi_{i,j} = \pi$$

**Illustration** Soit un ensemble  $S$  tel que  $S = \{1, 2, 3, 4\}$  et une partition  $\pi$  sur  $S$  telle que  $\pi = \{\overline{1, 2, 3, 4}\}$ . La partition  $\pi$  est décomposée en

– une partition  $\pi_{1,2} = \{\overline{1, 2, 3, 4}\}$

– une partition  $\pi_{1,3} = \{\overline{1, 3, 2, 4}\}$

puisque

$$\{\overline{1, 2, 3, 4}\} + \{\overline{1, 3, 2, 4}\} = \{\overline{1, 2, 3, 4}\} = \pi$$

Les décompositions suivantes :

–  $\pi_{1,2}$  et  $\pi_{2,3}$

–  $\pi_{1,3}$  et  $\pi_{2,3}$

sont également possibles.

2. La partition  $m(\pi_{i,j})$  est obtenue en regroupant dans un seul bloc les éléments de  $\delta(i)$  et  $\delta(j)$ .

**Illustration** Soit la fonction  $\delta$  définie sur l'ensemble  $S$  de l'illustration précédente telle que :

$$\begin{aligned}\delta(1) &= 2 & \delta(2) &= 3 \\ \delta(3) &= 4 & \delta(4) &= 1\end{aligned}$$

La partition  $m(\pi_{1,2})$  est calculée en regroupant les éléments  $\delta(1) = 2$  et  $\delta(2) = 3$  au sein d'un même bloc :

$$m(\pi_{1,2}) = \{\overline{2, 3, 1, 4}\}$$

3. L'ensemble des partitions  $m(\pi_{i,j})$  est additionné pour calculer  $m(\pi)$ .

$$m(\pi) = \sum m(\pi_{i,j})$$



**Illustration** La partition  $m(\pi)$  telle que  $\pi = \{\overline{1, 2, 3, 4}\}$  est donnée par

$$\begin{aligned} m(\pi) &= m(\pi_{1,2}) + m(\pi_{1,3}) \\ &= \{\overline{2, 3, 1, 4}\} + \{\overline{2, 4, 1, 3}\} \\ &= \{\overline{1, 2, 3, 4}\} \end{aligned}$$

Si  $\delta$  est une application définie sur un produit cartésien d'ensembles

$$\begin{aligned} \delta : S \times I &\rightarrow S \\ (s, \alpha) &\mapsto s^+ = \delta(s, \alpha) \end{aligned}$$

où  $I$  est un ensemble quelconque, alors la partition  $m(\pi_{i,j})$  où  $i, j$  sont des éléments de  $S$  est calculée comme suit :

Définissons une partition  $\pi_{i,j}^\alpha$  telle que  $\pi_{i,j}^\alpha$  est le résultat du calcul de l'opérateur  $m(\pi_{i,j})$  pour un élément  $\alpha \in I$  donné. La partition  $m(\pi_{i,j})$  est donnée par la relation

$$m(\pi_{i,j}) = \sum \pi_{i,j}^\alpha, \quad \forall \alpha \in I$$

Le reste de la procédure de calcul de  $m$  demeure identique.

**Illustration** Considérons les ensembles  $S = \{1, 2, 3\}$  et  $I = \{a, b\}$  et une application  $\delta : S \times I \rightarrow S$  définie par

$$\begin{array}{ll} \delta(1, a) = 2 & \delta(1, b) = 3 \\ \delta(2, a) = 3 & \delta(2, b) = 1 \\ \delta(3, a) = 1 & \delta(3, b) = 2 \end{array}$$

Soit une partition  $\pi_{1,2} = \{\overline{1, 2, 3}\}$ .  $m(\pi_{1,2})$  est calculé comme suit : Les partitions  $\pi_{1,2}^\alpha$  pour tous les  $\alpha \in I$  sont données par :

$$\begin{aligned} \pi_{1,2}^a &= \{\overline{2, 3, 1}\} \\ \pi_{1,2}^b &= \{\overline{3, 1, 2}\} \end{aligned}$$

La partition  $m(\pi_{1,2})$  est donnée par

$$m(\pi_{1,2}) = \pi_{1,2}^a + \pi_{1,2}^b = \{\overline{1, 2, 3}\}$$

## b. Détermination de l'opérateur M

Soit un ensemble  $S$ , une partition quelconque  $\pi$  de  $S$  et une application  $\delta$  définie par  $\delta : S \rightarrow S$  ou  $\delta : S \times I \rightarrow S$ . L'opérateur  $M$  est calculé à l'aide de la relation suivante

$$M(\pi) = \sum \{\pi_{i,j} \mid m(\pi_{i,j}) \leq \pi\} \forall i, j \in I \quad (3.17)$$

Remarquons que cette relation nécessite le calcul préalable des différents  $m(\pi_{i,j})$  pour toutes les paires  $i, j$  possibles.

**Remarque 3.3.4** La relation (3.17) est justifiée par le raisonnement suivant : Considérons toutes les partitions  $\pi_{i,j}$  tels que  $m(\pi_{i,j}) \leq \pi$ . Cela veut dire également que  $(\pi_{i,j}, \pi)$  forment une paire algébrique. Les partitions  $\pi_{i,j}$  forment une couverture complète de S, et les paires algébriques  $(\pi_{i,j}, m(\pi_{i,j}))$  constituent une base toutes les paires algébriques possibles sur S, et par conséquent, à partir de la définition 3.20, l'opérateur M est donné par la relation (3.17).

**Illustration** Soit un ensemble S tel que  $S = \{1, 2, 3\}$  et une partition  $\pi$  sur S telle que  $\pi = \{\overline{1}, \overline{2}, \overline{3}\}$ . Soit une application  $\delta$  sur S telle que

$$\delta(1) = 2 \quad \delta(2) = 3 \quad \delta(3) = 3$$

Pour calculer la partition  $M(\pi)$  il est nécessaire de calculer les différents  $m(\pi_{i,j})$  pour toutes les paires  $i, j$  de S

$$m(\pi_{1,2}) = \{\overline{1}, \overline{2}, \overline{3}\} \not\leq \pi$$

$$m(\pi_{1,3}) = \{\overline{1}, \overline{2}, \overline{3}\} \not\leq \pi$$

$$m(\pi_{2,3}) = \{\overline{1}, \overline{2}, \overline{3}\} \leq \pi$$

La partition  $M(\pi)$  est donnée alors par

$$M(\pi) = \pi_{2,3} = \{\overline{1}, \overline{2}, \overline{3}\}$$

### 3.3.6 Application aux machines séquentielles

Considérons une machine séquentielle  $\Sigma$  définie par le quintuplet  $(S, I, O, \delta, \lambda)$ . Il est possible de définir une algèbre des paires  $\Delta$  par rapport aux ensembles S, I, O et aux relations  $\delta, \lambda$ . Quatre types de paires algébriques peuvent être formés : les paires S – S, S – O, I – S et I – O

**Définition 3.22** Les définitions suivantes sont valables sur une machine séquentielle  $\Sigma$

– Soit  $\pi, \tau$  deux partitions sur S.  $(\pi, \tau)$  est une paire S-S si et seulement si pour tout  $x \in I$

$$s \equiv t(\pi) \Rightarrow \delta(s, x) \equiv \delta(t, x)(\tau)$$

– Soit  $\xi$  une partition sur I et  $\tau$  une partition sur S.  $(\xi, \tau)$  est une paire I-S si et seulement si pour tout  $s \in S$

$$a \equiv b(\xi) \Rightarrow \delta(s, a) \equiv \delta(s, b)(\tau)$$

– Soit  $\pi$  une partition sur S et  $\omega$  une partition sur O.  $(\pi, \omega)$  est une paire S-O si et seulement si pour tout  $x \in I$

$$s \equiv t(\pi) \Rightarrow \lambda(s, x) \equiv \lambda(t, x)(\omega)$$

– Soit  $\xi$  une partition sur I et  $\omega$  une partition sur O.  $(\xi, \omega)$  est une paire I-O si et seulement si pour tout  $s \in S$

$$b \equiv a(\xi) \Rightarrow \lambda(s, a) \equiv \lambda(s, b)(\omega)$$

Pour les machines traitées dans ce mémoire, c'est-à-dire les machines de Moore, il n'existe aucune paire algébrique  $\{S-O\}$  non triviale<sup>10</sup>.

Le théorème suivant [HS66] valide la possibilité de former quatre algèbres des paires pour une machine séquentielle  $\Sigma$ .

**THÉORÈME 3.5** *Soit une machine séquentielle  $\Sigma$ . Les ensembles  $\Omega_{S-S}$ ,  $\Omega_{S-O}$ ,  $\Omega_{I-S}$ ,  $\Omega_{I-O}$ , qui regroupent respectivement toutes les paires  $S-S$ ,  $S-O$ ,  $I-S$ ,  $I-O$  pour  $\Sigma$ , sont des algèbres des paires.*

Les opérateurs  $M$  et  $m$  relatifs à une paire  $S-S$  (resp.  $S-O, I-S, I-O$ ), sont notés  $M_{S-S}$  et  $m_{S-S}$  (resp.  $M_{S-O}$  et  $m_{S-O}, \dots$ ).

### La propriété de substitution pour les machines séquentielles

Posons  $L = L_1 = L_2$ . Soit  $\Delta_\delta$  une algèbre des paires où  $\delta$  est l'application de transition d'état de la machine d'état  $\Sigma(S, I, \delta)$ . Les partitions de l'ensemble d'état  $S$  qui possèdent la propriété de substitution sont définies par :

**Définition 3.23** *Soit  $\pi \in L$  une partition sur  $S$ .  $\pi$  possède la propriété de substitution par rapport à  $\Delta$  si et seulement si  $(\pi, \pi)$  est une paire algébrique.*

L'ensemble des partitions avec la propriété de substitution pour une machine séquentielle, et la relation d'ordre des partitions  $\leq$ , forment un treillis  $L_\Delta : (\Delta, \leq)$

$$L_\Delta = \{\pi \in L \mid (\pi, \pi) \in \Delta\}$$

**Corollaire 3.5.1** *Soit  $\Delta$  une algèbre des paires sur  $L \times L$ . Le treillis  $L_\Delta$  est un sous-treillis de  $L$  avec les propriétés suivantes*

1. Le plus grand élément est  $\mathbb{I} : (\mathbb{I}, \mathbb{I}) \in \Delta$
2. Le plus petit élément est  $\mathbb{O} : (\mathbb{O}, \mathbb{O}) \in \Delta$
3.  $L_\Delta = \{x \in L \mid (x \geq m(x))\} = \{x \in L \mid (x \leq M(x))\}$

### 3.3.7 Conclusion

Dans cette section, les définitions de l'algèbre des paires ont été données et ses propriétés et opérations ont été présentées dans le cadre général d'une application entre deux ensembles et étendues au cas d'une machine séquentielle. Les outils que nous fournit l'algèbre permettent de décrire de manière particulière une application entre deux ensembles. Par extension, elle permet également de décrire l'évolution d'une machine séquentielle à l'aide d'un treillis. Ces résultats seront exploités dans le chapitre suivant pour traiter la décomposition des modèles événementiels, car la représentation en treillis permet de proposer des méthodes algorithmiques efficaces. Dans la suite de ce chapitre, l'extension de l'algèbre des paires au domaine des ensembles infinis est présentée.

<sup>10</sup>Les partitions triviales sont les partitions  $\mathbb{I}$  et  $\mathbb{O}$

## 3.4 L'algèbre des fonctions

### 3.4.1 Introduction

Au début des années 1990, les professeurs Zhirabok et Shumsky [ZS87, Shu91, ZS93] ont proposé un ensemble d'outils mathématiques pour la manipulation de modèles comportementaux analytiques. Ces outils sont directement inspirés de l'algèbre des paires. Ils ont reçu le nom d'algèbre des fonctions, puisqu'il s'agit de relations entre ensembles infinis. L'objectif fixé est de proposer une alternative aux outils de géométrie différentielle à base d'algèbre de Lie [Isi95]. Le choix fait par les auteurs était de situer la méthode par rapport aux méthodes existantes, c'est-à-dire par rapport à la géométrie différentielle [Shu94, Zhi97, SZ05], l'espace de parité [Shu99, Shu97, Shu00], les méthodes à base d'observateurs [Shu97] et d'appliquer l'algèbre des fonctions dans différents domaines : robotique [FVZ03, FVZ99] et détection de pannes de capteurs [ZP93b, ZP93a].

### 3.4.2 Principe

L'algèbre des fonctions repose fondamentalement sur les notions de l'algèbre des paires. L'algèbre des paires permet de manipuler le partitionnement d'ensemble  $S$  par rapport à une application  $\delta$ , pour faire ressortir par exemple des propriétés de l'application.

L'algèbre des fonctions reprend cette idée de partitionnement pour des ensembles infinis  $X$ , par rapport à une fonction  $f$ . Ici, le terme *fonction* décrit la relation entre deux ensembles infinis qui associe à chaque élément de l'ensemble de départ un seul élément de l'ensemble d'arrivée.

L'intérêt de l'algèbre des fonctions est d'éviter les calculs compliqués qu'impliquerait la manipulation des partitions d'un ensemble infini. Cela est possible grâce à la dualité entre les partitions et les fonctions. Il suffit de définir un ensemble de fonctions vectorielles différentiables sur  $X$  pour définir une partition. Considérons la fonction suivante

$$\alpha : X \longrightarrow S$$

La fonction  $\alpha(x)$  partitionne son ensemble de définition  $X$ . La partition obtenue est notée  $\omega$  :

**Définition 3.24** Soit  $(x, x') \in X \times X$ .  $x$  et  $x'$  appartiennent à la même partition  $\omega$  si et seulement s'ils possèdent la même image  $\alpha(x) = \alpha(x')$ .

$$(x, x') \in X \times X : x \equiv x'(\omega) \Leftrightarrow \alpha(x) = \alpha(x')$$

En utilisant cette définition, il devient possible de manipuler les partitions en manipulant les fonctions qui les décrivent, ce qui est plus commode car la manipulation de partitions devient compliquée si elles comportent un nombre important d'éléments.

Définissons l'ensemble de toutes les fonctions vectorielles possibles sur un ensemble  $X \subseteq \mathbb{R}^n$  noté  $D_X$ . L'algèbre des fonctions se base sur le treillis de toutes les

fonctions  $\alpha_i \in D_X$ ,  $\forall i$ . Ces fonctions serviront à décrire le fonctionnement du modèle continu  $\Sigma(X, Y, W, f, h)$  en utilisant des outils similaires à ceux décrits dans la section précédente.

### 3.4.3 Définitions

Pour former un treillis, une relation d'ordre doit être associée à l'ensemble  $D_X$ . Par ailleurs, pour former une algèbre, deux opérations internes doivent être associées à  $D_X$ . Les définitions formelles des trois éléments sont données.

#### Relation d'ordre sur $D_X$

Soient les deux fonctions vectorielles  $\alpha$  et  $\beta$  définies sur  $X$

$$\begin{aligned}\alpha &: X \longrightarrow S \\ \beta &: X \longrightarrow T\end{aligned}$$

avec  $S, T \subseteq X$ . La hiérarchie entre les deux fonctions  $\alpha$  et  $\beta$  est obtenue à l'aide des définitions suivantes

**Définition 3.25** Soit la relation  $\leq$  définie sur  $D_X$ .  $\alpha$  est inférieure ou égale à  $\beta$  ( $\alpha \leq \beta$ ) si et seulement si il existe une fonction  $v : S \longrightarrow T$  telle que  $v(\alpha) = v \circ \alpha = \beta$

$$\alpha, \beta \in D_X \exists v \in D_X : v \circ \alpha = \beta$$

**Définition 3.26** Soit la relation  $\geq$  définie sur  $D_X$ .  $\alpha$  est supérieure ou égale à  $\beta$  si et seulement si  $\beta$  est inférieure ou égale à  $\alpha$

$$\alpha \geq \beta \Leftrightarrow \beta \leq \alpha$$

La relation d'équivalence entre  $\alpha$  et  $\beta$  est définie par

**Définition 3.27** Soit la relation  $\equiv$  définie sur  $D_X$ .  $\alpha$  est équivalente à  $\beta$  et on note  $\alpha \equiv \beta$  si et seulement si  $\alpha$  est supérieure à  $\beta$  et  $\beta$  est supérieure à  $\alpha$  simultanément

$$\alpha \equiv \beta \Leftrightarrow \alpha \leq \beta \text{ et } \beta \leq \alpha$$

**Illustration** Considérons les fonctions suivantes

$$\begin{aligned}\alpha_1(x) &= \begin{bmatrix} x_1 \\ x_1 + x_2 \end{bmatrix} & \alpha_2(x) &= \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} & \alpha_3(x) &= \begin{bmatrix} x_1 \\ x_2 x_1 \end{bmatrix} \\ \alpha_4(x) &= \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} & \beta(x) &= \begin{bmatrix} x_1 \\ x_2 + x_1 \\ x_2 - x_3 \end{bmatrix} & \text{avec } x &= \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\end{aligned}$$

Les relations suivantes sont valides

$$\begin{aligned}\beta &\leq \alpha_1 & \beta &\leq \alpha_2 & \beta &\leq \alpha_3 \\ \beta &\geq \alpha_4 & \beta &\leq \alpha_4 & \beta &\equiv \alpha_4\end{aligned}$$

Remarquons que  $\alpha_2 \not\leq \alpha_3$  et que  $\alpha_3 \not\leq \alpha_2$ . Il n'est pas possible de définir une hiérarchie entre ces deux fonctions.

Le treillis  $L_X = (D_X, \leq)$  est constitué de toutes les fonctions contenues dans l'ensemble  $D_X$ . Chaque groupe d'éléments de  $L_X$  possède un minima et un maxima au sens défini par la relation d'ordre  $\leq$ .

**Définition 3.28** L'opération max qui donne le plus grand élément d'un ensemble de fonctions donné est définie par

$$\begin{aligned} \max : L_X \times \cdots \times L_X &\longrightarrow L_X \\ (\alpha_1, \dots, \alpha_k) &\longrightarrow \{\alpha_i | \forall j \leq k : \alpha_j \leq \alpha_i\} \end{aligned}$$

**Définition 3.29** L'opération min qui donne le plus petit élément d'un ensemble de fonctions donné est définie par

$$\begin{aligned} \min : L_X \times \cdots \times L_X &\longrightarrow L_X \\ (\alpha_1, \dots, \alpha_k) &\longrightarrow \{\alpha_i | \forall j \leq k : \alpha_i \leq \alpha_j\} \end{aligned}$$

**Illustration** Considérons les fonctions définies dans l'exemple précédent

$$\begin{aligned} \min(\alpha_1, \alpha_3, \alpha_4) &= \alpha_4 \\ \max(\alpha_1, \alpha_3, \alpha_4) &= \alpha_3 \end{aligned}$$

### Opérations internes sur $D_X$

Le treillis  $L_X$ , est muni de deux opérations internes qui sont définies selon le théorème 3.2 :

**Définition 3.30** On définit l'opération «  $\square$  » de  $L_X \times L_X$  vers  $S \cap T$  par

$$\alpha \square \beta = \max\{g | \alpha \leq g \text{ et } \beta \leq g\}$$

où  $g : X \rightarrow S \cap T$

**Définition 3.31** On définit l'opération «  $\otimes$  » de  $X \times X$  vers  $S \cup T$  par

$$\alpha \otimes \beta = \min\{w | w \leq \alpha \text{ et } w \leq \beta\}$$

où  $w : X \rightarrow S \cup T$

**Illustration** Considérons les fonctions du premier exemple

$$\begin{array}{lll} \alpha_1 \square \beta = \alpha_1 & \alpha_1 \otimes \beta = \beta & \alpha_1 \square \alpha_2 = \begin{bmatrix} x_1 \end{bmatrix} \\ \alpha_1 \otimes \alpha_2 = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} & \alpha_3 \otimes \alpha_2 = \begin{bmatrix} x_1 \\ x_1 x_2 \\ x_3 \end{bmatrix} & \alpha_3 \square \alpha_2 = \begin{bmatrix} x_1 \end{bmatrix} \end{array}$$

Les opérations  $\square$  et  $\otimes$ , définies par rapport au treillis  $L_X$  possèdent les propriétés de commutativité, d'associativité, d'idempotence et d'absorbance (théorème 3.1). Par ailleurs, l'ensemble  $D_X$ , qui est l'espace vectoriel défini par  $f$ , muni des opérations  $\square$  et  $\otimes$  forme une algèbre par rapport à  $f$ .

### 3.4.4 Paire algébrique en continu

L'algèbre des fonctions est une extension de l'algèbre des paires aux ensembles infinis. Une paire algébrique décrit le flux d'information véhiculé par une fonction  $f$  et n'est valide que par rapport à cette fonction. Dans ce qui suit, nous définissons la paire état-état ( $S - S$ ). Il est aisé d'étendre cette définition au reste des cas, par analogie avec 3.22.

**Définition 3.32** Les fonctions  $\alpha$  et  $\beta$  forment une paire algébrique, et on note  $(\alpha, \beta) \in \Delta_f$  si et seulement si

$$\alpha(x) \leq \beta(f(x))$$

Selon que les fonctions  $\alpha$  et  $\beta$  soient supérieures, équivalentes ou inférieures, il sera question de surplus, d'égalité ou d'insuffisance de connaissance d'antécédents pour déterminer l'image, ce qui reprend, au formalisme près, la description d'une algèbre des paires.

#### Opérateurs $m$ et $M$

La définition 3.32 est utilisée pour définir les opérateurs *minimum*  $m$  et *maximum*  $M$ .

**Définition 3.33**  $m(\alpha)$  est la plus petite fonction possible dans le sens de la relation  $\leq$  qui forme une paire à gauche avec la fonction  $\alpha$

$$(\alpha, m(\alpha)) \in \Delta_f \Leftrightarrow \forall (\alpha, \beta) \in \Delta_f : m(\alpha) \leq \beta$$

**Définition 3.34**  $M(\beta)$  est la plus grande fonction possible, dans le sens de la relation  $\leq$ , qui forme une paire à droite avec la fonction  $\beta$

$$(M(\beta), \beta) \in \Delta_f \Leftrightarrow \forall (\alpha, \beta) \in \Delta_f : \alpha \leq M(\beta)$$

$m$  décrit le maximum d'informations pouvant être déterminées connaissant l'information  $\alpha$  sur l'état présent.  $M$  décrit le minimum d'informations nécessaires pour obtenir l'information  $\beta$  sur l'état suivant.

**Illustration** Considérons la fonction suivante

$$f(x) = \begin{bmatrix} x_2 x_3 \\ x_2 + x_2 x_3 \\ x_1 \end{bmatrix} \quad \text{avec } x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (3.18)$$

De nombreuses fonctions forment des paires algébriques par rapport à  $f$ , par exemple

$$\left( \begin{bmatrix} x_2 \\ x_2 x_3 \end{bmatrix}, \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right) \in \Delta_f \quad \left( [x_1], [x_3] \right) \in \Delta_f \quad \left( [x_2], [x_1 - x_2] \right) \in \Delta_f$$

Détaillons la première relation :

$$\alpha(x) = \begin{bmatrix} x_2 \\ x_2x_3 \end{bmatrix} \text{ et } \beta(x) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\beta(f(x)) = \begin{bmatrix} x_2x_3 \\ x_2 + x_2x_3 \end{bmatrix}$$

Les fonctions  $\alpha$  et  $\beta$  forment une paire algébrique puisque la relation de la définition 3.32 est satisfaite  $\alpha \leq \beta(f(x))$

Définissons les fonctions  $\alpha_i$

$$\alpha_1(x) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \alpha_2(x) = \begin{bmatrix} x_2 \end{bmatrix} \qquad \alpha_3(x) = \begin{bmatrix} x_3 \end{bmatrix}$$

Les opérateurs  $m$  et  $M$  appliqués aux différentes fonctions  $\alpha$  donnent les résultats suivants

$$M(\alpha_1)(x) = \begin{bmatrix} x_1 \\ x_2 \\ x_2x_3 \end{bmatrix} \qquad M(\alpha_2)(x) = \begin{bmatrix} x_2 \\ x_2x_3 \end{bmatrix} \qquad M(\alpha_3)(x) = \begin{bmatrix} x_1 \end{bmatrix}$$

$$m(\alpha_1)(x) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad m(\alpha_2)(x) = \begin{bmatrix} x_1 - x_2 \end{bmatrix} \qquad m(\alpha_3)(x) = \begin{bmatrix} \emptyset \end{bmatrix}$$

### Propriétés

Les propriétés principales des opérateurs de l'algèbre des fonctions [ZS93] sont données

- $\alpha \leq \beta \Leftrightarrow \alpha \otimes \beta \equiv \alpha \Leftrightarrow \alpha \square \beta \equiv \beta$
- $\alpha \leq \delta$  et  $\beta \leq \delta \Leftrightarrow \alpha \square \beta \leq \delta$
- $\alpha \leq \beta \Leftrightarrow m(\alpha) \leq m(\beta)$
- $M(\alpha \otimes \beta) \equiv M(\alpha) \otimes M(\beta)$
- $M(m(\alpha)) \geq \alpha$
- $m(M(\beta)) \leq \beta$



### 3.4.5 Parallèle entre les méthodes géométriques et les méthodes algébriques

Pour illustrer les opérateurs de l'algèbre des fonctions, un parallèle est fait avec la géométrie différentielle<sup>11</sup> pour le cas général non linéaire. Le parallèle avec le cas linéaire peut être déduit.

Considérons les deux fonctions vectorielles de classe  $C^\infty$  suivantes

$$\begin{aligned}\alpha &: X \rightarrow S \\ \beta &: X \rightarrow T\end{aligned}$$

La dimension de  $\alpha$  est  $l_1$  et celle de  $\beta$  est  $l_2$ . Les composantes de  $\alpha$  et  $\beta$  sont notées respectivement  $\alpha_i$  et  $\beta_i$  où le  $i$  représente le numéro de la composante.  $\alpha_i$  (ou  $\beta_i$ ) peut être exprimée avec une 1-forme différentielle dont l'expression en coordonnées locales est donnée par

$$d\alpha_i = \sum_{j=1}^n \frac{\partial \alpha_i}{\partial x_j} dx_j$$

Aux fonctions  $\alpha$  et  $\beta$  correspondent deux co-distributions  $\mathbb{A}$  et  $\mathbb{B}$  décrites par les deux espaces vectoriels  $\{d\alpha_1(x), \dots, d\alpha_{l_1}(x)\}$  et  $\{d\beta_1(x), \dots, d\beta_{l_2}(x)\}$ . La fonction  $\alpha$  est complètement définie dans l'espace vectoriel  $\mathbb{A}$ , et il est possible de représenter  $\alpha(x)$  pour tout  $x \in X$  avec des combinaisons linéaires des  $d\alpha_i(x)$ . Il en va de même pour  $\beta$ .

Par conséquent,  $\mathbb{A}$  et  $\mathbb{B}$  sont les représentations géométriques de  $\alpha$  et de  $\beta$ .

#### Relation d'ordre

La relation d'ordre  $\leq$  associée à  $L_X$  peut être interprétée comme suit : soit une fonction  $\beta$  telle que  $\beta = v \circ \alpha$ . Cela équivaut à l'écriture  $\beta_i = v_i \alpha$  et implique que

$$d\beta_i = \sum_{j=1}^n \frac{\partial v_i}{\partial \alpha_j} d\alpha_j$$

Cette équation est interprétée géométriquement par  $\mathbb{B} \subseteq \mathbb{A}$ . La réciproque est également vraie : soit une co-distribution  $\mathbb{B} \subseteq \mathbb{A}$ . Il est évident que  $\mathbb{B}$  peut être exprimée dans l'espace vectoriel décrit par  $\{d\alpha_1, \dots, d\alpha_{l_1}\}$  ce qui implique que  $d\beta_i = \eta_i d\alpha$ . Par conséquent  $\beta \geq \alpha$ .

L'équivalence  $\alpha \equiv \beta$  devient par conséquent  $\mathbb{A} = \mathbb{B}$ .

#### Opérations internes $\square$ et $\otimes$

L'opération  $\alpha \otimes \beta$  détermine la fonction vectorielle  $\mu$  (associée à la co-distribution  $\mathbb{M}$ ) de dimension minimale ( $l_1 + l_2$  si  $\alpha$  et  $\beta$  sont indépendantes) dont les composantes  $\mu_i$  peuvent exprimer les composantes de  $\alpha$  et de  $\beta$ . Géométriquement cela correspond à

<sup>11</sup>Les notions de la géométrie différentielle sont introduites en annexe B

$$\mu \leq \alpha \text{ et } \mu \leq \beta \Leftrightarrow \mu = \alpha \otimes \beta \quad (3.19)$$

$$\mathbb{A} \subseteq \mathbb{M} \text{ et } \mathbb{B} \subseteq \mathbb{M} \Leftrightarrow \mathbb{M} = \mathbb{A} \cup \mathbb{B} \quad (3.20)$$

En suivant le même type de raisonnement, nous montrons que l'opération  $\alpha \square \beta$  est interprétée géométriquement par  $\mathbb{A} \cap \mathbb{B}$

### Algèbre de Lie et algèbre des fonctions

Nous allons interpréter géométriquement la notion de paire à travers l'analyse de l'équation différentielle affine suivante

$$\dot{x} = \sum_i g_i(x)$$

où les différents  $g_i$  sont de classe  $C^1$ . Soit une fonction  $\alpha(x)$ . L'opérateur  $M$  est calculé par l'expression suivante

$$M(\alpha) = \otimes_i \left( \frac{d\alpha}{dx} g_i \right) \quad (3.21)$$

Rappelons que la dérivée de Lie (voir annexe) s'écrit  $L_{g_i}(\alpha) = \sum_j \frac{\partial \alpha_j}{\partial x_j} g_i = \frac{d\alpha}{dx} g_i$ . Les deux équations sont identiques ce qui permet de dire que l'opérateur  $M$  est similaire à l'opérateur de la dérivée de Lie  $L_g$ .

Posons une co-distribution  $\mathbb{A}$ , où  $\mathbb{A}$  est involutive par rapport au champ de vecteurs  $\{g_1, \dots, g_k\}$ . Cela implique que

$$[\mathbb{A}, \mathbb{A}] \subset \mathbb{A}$$

et par conséquent

$$\forall g_i : L_{g_i}(\mathbb{A}) \subseteq \mathbb{A}$$

et

$$L_{g_i}(d\alpha_i) = \sum_j \eta_{i,j}(x) d\alpha_j$$

Puisque  $\mathbb{A}$  est involutive, chaque différentielle  $d\alpha_i$  peut être représentée dans les espaces vectoriels associés à  $\mathbb{A}$ . Par ailleurs, nous savons que

$$L_f(d\alpha_i) = dL_f(\alpha_i) = d\left(\frac{d\alpha}{dx} g_i\right)$$

(annexe B ou [Isi95]). Cette relation implique que  $\frac{d\alpha}{dx} g_i \geq \alpha$  et par analogie  $M(\alpha_i) \geq \alpha$ . Puisque  $M(\alpha) = \otimes_i M(\alpha_i)$  alors  $M(\alpha) \geq \alpha$  ce qui satisfait la condition  $(\alpha, \alpha) \in \Delta_f$  par rapport à l'équation différentielle (3.21).

Il existe donc des similitudes entre les outils et concepts algébriques et leurs équivalents géométriques notamment entre la dérivée de Lie et l'opérateur  $M$  et le crochet de Lie et la notion de paire algébrique.

### 3.5 Conclusion

Ce chapitre a permis de détailler les outils mathématiques utilisés dans ce mémoire : l'algèbre des paires et l'algèbre des fonctions. La relation entre les structures de treillis et d'algèbre est montrée et utilisée pour définir l'algèbre des paires à partir d'un ensemble de partitions d'ensembles finis et d'un ensemble d'opérations particulières. Ces opérations sont utilisées pour décrire une notion d'équivalence de partitions, la « propriété de substitution », qui servira par la suite pour formuler un critère d'équivalence de modèles événementiels au chapitre suivant.

L'algèbre des paires est ensuite étendue aux ensembles infinis en utilisant des fonctions continues pour décrire les partitions. L'ensemble des opérateurs de l'algèbre des paires est ensuite transposé dans ce cadre, ainsi que la notion d'équivalence entre partitions qui est étendue et formulée en utilisant des fonctions continues. De la même manière, cette notion d'équivalence est utilisée pour formuler un critère d'équivalence de modèles temporels dans la suite de ce mémoire.

Une analogie avec les outils de la géométrie différentielle est montrée dans la dernière partie de ce chapitre pour justifier l'utilisation des outils de l'algèbre des fonctions .

Pour de plus amples informations le lecteur peut consulter l'ouvrage [HS66] pour l'algèbre des paires et les travaux de Shumsky et de Zhirabok [Shu91, ZS93]. Pour les notions d'algèbre de Lie et de géométrie différentielle, un bref résumé est présenté en annexe B. Pour une information plus complète, l'ouvrage du professeur Isidori [Isi95] est recommandé.

# Chapitre 4

## Décomposition de modèles comportementaux pour la surveillance

### 4.1 Introduction

Ce chapitre est consacré à la description de la méthodologie de décomposition. La problématique de la décomposition est formulée de manière générale et une solution en plusieurs étapes est proposée. Ces étapes sont construites avec un niveau d'abstraction suffisant pour ne pas tenir compte de la forme des équations qui constituent le modèle à décomposer. Pour atteindre ce niveau d'abstraction, les étapes de la décomposition sont décrites avec des moyens algébriques présentés dans le chapitre 3.

L'objectif de la décomposition est de déterminer la partie du modèle complet qui possède des caractéristiques spécifiques. C'est une décomposition avec critères. Les critères utilisés sont le couplage à un sous-ensemble d'entrées ou le découplage de ce sous-ensemble. La décomposition permet d'obtenir des sous-modèles qui sont couplés à (ou découplés de) ces entrées, et qui sont *cohérents* en sortie avec le modèle initial. La notion de cohérence ici indique que pour des entrées identiques, il existe une relation unique et constante qui lie les sorties du sous-modèle aux sorties du modèle complet. Cette relation permet la comparaison des sorties entre les deux modèles. La différence entre l'évolution des sous-modèles couplés et découplés apparaît si l'excitation est différente entre le modèle complet et les sous-modèles :

- Les sorties de sous-modèle couplé ne sont plus cohérentes si une entrée *couplée* excite uniquement le modèle complet et pas le sous-modèle .
- Les sorties du sous-modèle découplé restent cohérentes pour n'importe quelle entrée *découplée* qui excite uniquement le modèle complet.

Ces deux types de sous-modèle sont utilisés pour synthétiser les indicateurs de défaillances structurés décrits dans le chapitre 2.

Le présent chapitre est organisé comme suit :

- La première partie est dédiée à la formulation du problème de la décomposition dans un cadre général. Les contraintes et les critères de la décomposition sont détaillés et les étapes de l'algorithme de décomposition sont énumérées.
- La deuxième partie concerne la décomposition de modèles comportementaux

temporels. L'algorithme de décomposition est implémenté en utilisant les opérateurs de l'algèbre des fonctions, et des illustrations de décompositions de modèles linéaires et non linéaires sont fournies. Les aspects calculatoires de la méthode sont traités également, et des solutions pour calculer les opérateurs de l'algèbre des fonctions sont proposés, ainsi qu'une extension de l'étape d'initialisation de l'algorithme en utilisant des techniques d'élimination de variables.

- La troisième partie concerne la décomposition de modèles comportementaux événementiels (machines séquentielles). L'algorithme de décomposition est écrit en utilisant les opérateurs de l'algèbre des paires, et des illustrations de décompositions de machines séquentielles sont données.

## 4.2 Principe général de la décomposition

### 4.2.1 Formulation du problème de décomposition

Considérons le modèle comportemental  $\Sigma$  décrit par le quintuplet  $(\mathcal{X}, \mathcal{W}, \mathcal{Y}, \mathcal{F}, \mathcal{H})$  (définition 2.1, chapitre 2) avec

$$\mathcal{F} : \mathcal{X} \times \mathcal{W} \rightarrow \mathcal{X} \qquad \mathcal{H} : \mathcal{X} \rightarrow \mathcal{Y}$$

Le cas où  $\mathcal{H} : \mathcal{X} \times \mathcal{W} \rightarrow \mathcal{Y}$  n'est pas considéré ici, puisque la démarche de décomposition n'est pas modifiée.

Soit  $\mathcal{W}' \subseteq \mathcal{W}$  un sous-ensemble d'entrées non vide. Il existe une partie de  $\Sigma$  qui est régie par  $\mathcal{W}'$  et uniquement par elle. Cette partie de  $\Sigma$  est un *sous-modèle*, noté  $\Sigma'$ , qui est défini par le quintuplet  $(\mathcal{X}', \mathcal{W}', \mathcal{Y}', \mathcal{F}', \mathcal{H}')$ , où  $\mathcal{X}' \subseteq \mathcal{X}$  est déterminé en considérant  $\mathcal{W}'$  et  $\mathcal{Y}' = \mathcal{H}(\mathcal{X}')$ .  $\mathcal{F}'$  et  $\mathcal{H}'$  sont des restrictions de  $\mathcal{F}$  et  $\mathcal{H}$  aux ensembles  $\mathcal{W}', \mathcal{X}'$  et  $\mathcal{Y}'$ .

$$\mathcal{F}' : \mathcal{X}' \times \mathcal{W}' \rightarrow \mathcal{X}' \qquad \mathcal{H}' : \mathcal{X}' \rightarrow \mathcal{Y}'$$

Selon cette formulation, pour un sous-ensemble d'entrées donné  $\mathcal{W}'$ , seul le sous-ensemble  $\mathcal{X}'$  est inconnu. La relation qui permet d'obtenir  $\Sigma'$  à partir de  $\Sigma$  est notée  $\varphi$

$$\begin{aligned} \varphi : \Psi &\rightarrow \Psi \\ \Sigma &\mapsto \Sigma' \\ (\mathcal{X}, \mathcal{W}, \mathcal{Y}, \mathcal{F}, \mathcal{H}) &\mapsto (\mathcal{X}', \mathcal{W}', \mathcal{Y}', \mathcal{F}', \mathcal{H}') \end{aligned} \tag{4.1}$$

où  $\Psi$  est l'ensemble de tous les quintuplets qu'il est possible d'obtenir à partir de  $(\mathcal{X}, \mathcal{W}, \mathcal{Y}, \mathcal{F}, \mathcal{H})$ . La relation  $\varphi$  permet de déterminer les éléments du quintuplet qui décrit  $\Sigma'$ .  $\varphi$  existe si le sous-modèle est réalisable.

Il a été mentionné précédemment que la connaissance de  $\mathcal{W}'$  et de  $\mathcal{X}'$  permet de définir complètement le sous-modèle  $\Sigma'$ . Ici,  $\mathcal{W}'$  est utilisé pour formuler le critère de décomposition et  $\mathcal{X}'$  est le résultat de la décomposition.

Le problème de décomposition se réduit alors à une forme plus simple : définir, pour un sous-ensemble d'entrées  $\mathcal{W}' \subseteq \mathcal{W}$  donné, la fonction  $\psi$  telle que

$$\begin{aligned} \psi : \mathcal{X} &\rightarrow \mathcal{X}' \subseteq \Psi_{\mathcal{X}} \\ x &\mapsto x' = \psi(x) \end{aligned} \quad (4.2)$$

où  $\Psi_{\mathcal{X}}$  est l'ensemble de toutes les combinaisons possibles d'éléments de  $\mathcal{X}$ .  $\mathcal{X}'$  est l'image de  $\mathcal{X}$  par  $\psi$ .

La partie restante  $\Sigma_*$  du modèle  $\Sigma$  est la partie qui n'est pas influencée par les entrées  $\mathcal{W}'$ . Le sous-modèle  $\Sigma_*$  est le complément du sous-modèle  $\Sigma'$  et est décrit par le quintuplet

$$(\mathcal{X}_*, \mathcal{W}_*, \mathcal{Y}_*, \mathcal{F}_*, \mathcal{H}_*)$$

où  $\mathcal{W}_* \cup \mathcal{W}' = \mathcal{W}$  et  $\mathcal{X}_* \subseteq \mathcal{X}$  est à déterminer et  $\mathcal{Y}_* = \mathcal{H}_*(\mathcal{X}_*)$ .  $\mathcal{F}_*$  et  $\mathcal{H}_*$  sont des restrictions de  $\mathcal{F}$  et  $\mathcal{H}$  aux ensembles  $\mathcal{W}_*, \mathcal{X}_*$  et  $\mathcal{Y}_*$ .

Par analogie avec la relation (4.2), la détermination de  $\Sigma_*$  passe par la définition d'un sous-ensemble d'entrées  $\mathcal{W}_*$ , et la recherche de la fonction  $\phi$  telle que

$$\begin{aligned} \phi : \mathcal{X} &\rightarrow \mathcal{X}_* \subseteq \Psi_{\mathcal{X}} \\ x &\mapsto x_* = \phi(x) \end{aligned} \quad (4.3)$$

$\mathcal{X}_*$  est l'image de  $\mathcal{X}$  par  $\phi$ .

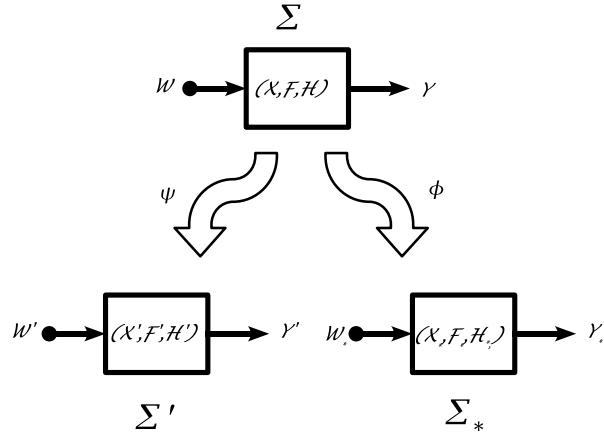


FIG. 4.1 – Diagramme représentant la décomposition d'un modèle  $\Sigma$

En conclusion, la décomposition d'un modèle comportemental a pour objectif de déterminer deux sous-modèles complémentaires  $\Sigma'$  et  $\Sigma_*$  (figure 4.1). Ces deux sous-modèles possèdent des propriétés de couplage et de découplage par rapport à un sous-ensemble d'entrées particulier.  $\Sigma'$  est le sous-modèle *couplé* à  $\mathcal{W}'$  et  $\Sigma_*$  est le sous-modèle qui est *découplé* de  $\mathcal{W}'$ .

Cette décomposition est sujette à des critères et à des contraintes :

- Le critère de la décomposition consiste à choisir un sous-ensemble d'entrées  $\mathcal{W}'$  de couplage ou de découplage. Ce sous-ensemble est constitué d'entrées qui partagent une propriété spécifique.
- La contrainte de la décomposition consiste à vérifier la possibilité de construire les sous-modèles  $\Sigma'$  et  $\Sigma_*$ , et de les écrire sous la même forme que le modèle complet  $\Sigma$  (voir *réalisabilité*). En outre, les sous-modèles obtenus doivent être de la plus grande dimension possible et avoir des sorties qu'il est possible de comparer au modèle complet.

Le problème est donc formulé comme un problème d'optimisation :

- Déterminer la plus grande partie de  $\Sigma$  qui satisfait le critère de décomposition. Cette partie est dite *ensemble de solutions candidates*.
- Déterminer à partir de l'ensemble de solutions candidates celles qui permettent de décrire un sous-modèle couplé à (ou découplé de)  $\mathcal{W}'$  *réalisable*. Le sous-modèle obtenu doit être de dimension maximale.

Un algorithme de décomposition itératif est écrit en fonction de cette formulation. A chaque itération, l'ensemble de solutions candidates est réduit jusqu'à ce que les contraintes de la décomposition soient satisfaites. L'algorithme 1 décrit la méthode de détermination de  $\Sigma'$ . L'algorithme de détermination de  $\Sigma_*$  est identique.

**Entrées :** Initialisation : choix du sous-ensemble  $\mathcal{W}'$  en fonction du critère de décomposition

**Données :**  $\Sigma$ ,  $i$  (compteur)

**Sorties :**  $\Sigma', \psi$

**début**

Initialisation : Déterminer  $\mathcal{X}^0$  à partir de  $\mathcal{W}'$  ;

$i=0$  ;

**tant que** *les contraintes de décomposition ne sont pas remplies* **faire**

    incrémenter  $i$  ;

    Chercher  $\mathcal{X}^i$  à partir de  $\mathcal{X}^{i-1}$  en respectant le critère de décomposition ;

$\psi = \psi^i$  ;

    Déterminer  $\mathcal{Y}', \mathcal{H}'$  ;

**fin**

**retourner**  $\Sigma'$

**Algorithme 1 :** Structure de l'algorithme de décomposition

Cet algorithme a besoin d'être détaillé : les critères et les contraintes de la décomposition sont amenés à être définis en fonction de l'application considérée. Pour répondre à la problématique définie dans le chapitre 2, les paramètres de la décomposition proposée sont détaillés dans les sections suivantes.

#### 4.2.2 Critère de couplage et de découplage pour la surveillance

Le choix du critère de décomposition repose en premier lieu sur le type d'entrées sélectionnées pour le couplage ou le découplage. La décomposition peut avoir pour critère le couplage par rapport à un sous-ensemble d'entrées, pour obtenir un système

dont les sorties sont *sensibles* à ces entrées. Dans ce cas,  $\mathcal{W}'$  est égal au sous-ensemble de couplage, et la décomposition est obtenue par la fonction  $\psi$ . Inversement, si la décomposition a pour objectif le découplage d'un sous-ensemble d'entrées  $\mathcal{W}'$ , alors le sous-modèle  $\Sigma_*$  est défini en utilisant la fonction  $\phi$ .

Dans les deux cas, le critère de décomposition associe à l'ensemble  $\mathcal{W}'$ , un sous-ensemble  $\mathcal{X}' \subseteq \mathcal{X}$ . Le sous-ensemble  $\mathcal{X}'$  est le plus grand sous-ensemble de  $\mathcal{X}$  qui satisfait le critère de décomposition, c'est-à-dire qui est couplé à (ou découplé de)  $\mathcal{W}'$ . Toutes les solutions possibles de la décomposition sont des sous-ensembles ou des ensembles de combinaisons d'éléments de  $\mathcal{X}'$ , choisis tels que la solution finale remplisse la condition d'existence qui est définie ci-dessous.

**Remarque 4.2.1** *Dans le chapitre 2, une méthode de synthèse d'indicateurs de défaillances structurés à base de sous-modèles découplés a été proposée. Afin d'obtenir un modèle découplé d'un sous-ensemble d'entrées inconnues, les entrées à découpler sont regroupées dans l'ensemble  $\mathcal{W}'$ . En général, l'ensemble d'entrées s'écrit sous la forme*

$$\mathcal{W} = \{\mathcal{U}, \gamma, \rho\}$$

où  $\gamma$  est l'ensemble des perturbations,  $\rho$  est l'ensemble des défaillances et  $\mathcal{U}$  l'ensemble des entrées connues. L'ensemble  $\mathcal{W}'$  est composé de tous les éléments de  $\gamma$  et d'éléments choisis de  $\rho$ . Le sous-modèle recherché est découplé de  $\mathcal{W}'$ .

### 4.2.3 Condition d'existence d'un sous-modèle découplé

Le sous-ensemble  $\mathcal{X}'$  est une solution possible de la décomposition, mais il n'existe pas de certitude que le quintuplet  $(\mathcal{X}^0, \mathcal{W}', \mathcal{Y}^0, \mathcal{F}^0, \mathcal{H}^0)$  décrive un sous-modèle. Ici, le sous-ensemble  $\mathcal{Y}^0$  et les relations  $\mathcal{F}^0, \mathcal{H}^0$  sont définies selon les équations vues précédemment. Deux conditions d'existence du sous-modèle  $\Sigma'$  sont définies :

- La fonction de décomposition  $\psi$  ne doit pas altérer la relation de transition d'état  $\mathcal{F}$ .  $\psi$  est alors dite invariante, ou involutive, par rapport à  $\mathcal{F}$ . L'invariance a été définie dans la littérature en utilisant les éléments de la géométrie différentielle (voir la référence [Isi95] ou l'annexe B). Nous proposons ci-dessous, une formulation adaptée à notre problème en utilisant des notions de la théorie des ensembles.
- Il s'agit de démontrer l'existence d'une relation qui lie la sortie de  $\Sigma'$  à la sortie du modèle complet  $\Sigma$ , pour pouvoir les comparer. Nous appellerons cette contrainte la contrainte de *mesurabilité*. La mesurabilité est détaillée dans la deuxième partie de cette section.

Pour résumer, la première condition garantit que le sous-modèle  $\Sigma$  a une évolution équivalente à la partie de  $\Sigma$  qui remplit le critère de décomposition et l'ensemble des deux conditions garantissent que  $\Sigma'$  est équivalent à la partie de  $\Sigma$  qui remplit le critère de décomposition.

#### Contrainte d'invariance

Considérons le sous-modèle  $\Sigma'$ . La relation  $\mathcal{F}'$  est définie comme la restriction de la fonction d'état du modèle complet  $\mathcal{F}$  sur le sous-ensemble d'états  $\mathcal{X}'$ .



$$\mathcal{F}' : \mathcal{X}' \times \mathcal{W} \rightarrow \mathcal{X}'$$

Si  $\Sigma'$  décrit une partie de  $\Sigma$ , alors l'évolution de  $\Sigma'$  en réponse aux entrées de  $\mathcal{W}$  doit être la même que celle du modèle complet  $\Sigma$  pour la même excitation.

$$\forall w \in \mathcal{W}, \forall x \in \mathcal{X} : x' = \psi(x) \Leftrightarrow \mathcal{F}'(x', w) = \psi(\mathcal{F}(x, w)) \quad (4.4)$$

Ce qui implique

$$\mathcal{F}'(\psi(x), w) = \psi(\mathcal{F}(x, w)) \quad (4.5)$$

$$(\Psi \circ \mathcal{F})(\psi(x), w) = \psi(\mathcal{F}(x, w)) \quad (4.6)$$

Cette relation décrit la contrainte d'invariance, c'est-à-dire que l'application  $\psi$  est invariante par rapport à l'application  $\mathcal{F}$ . Puisque  $(\Psi \circ \mathcal{F})$  est la restriction de  $\mathcal{F}$  sur  $\mathcal{X}' = \psi(\mathcal{X})$ , la condition d'invariance (4.4) est remplie si et seulement si la restriction de  $\mathcal{F}$  sur l'ensemble  $\psi(\mathcal{X})$  est une application. Dans ce cas,  $\Sigma$  est décomposable en utilisant l'application  $\psi$  et  $\Sigma'$  est réalisable.

**Illustration** Soient les ensembles  $\mathcal{X}$ ,  $\mathcal{W}$  et la fonction  $\mathcal{F}$  suivants :

–  $\mathcal{W} = \{a, b, c\}$  où  $a, b, c$  sont indifféremment des séquences ou des signaux.

–  $\mathcal{X} = \{x_1, x_2, x_3\}$ .

$$- \mathcal{F}(\mathcal{X}) = \begin{cases} x_1^+ = f_1(x_1, x_2, x_3, a, b) \\ x_2^+ = f_2(x_2, x_3, b, c) \\ x_3^+ = f_3(x_2, x_3, b, c) \end{cases}$$

où l'exposant « + » désigne l'état futur.

Considérons les fonctions  $\psi_1$  et  $\psi_2$  tels que

$$\begin{array}{ll} \psi_1 : \mathcal{X} \rightarrow \mathcal{X}' & \psi_2 : \mathcal{X} \rightarrow \mathcal{X}'' \\ x_1 \mapsto \emptyset & x_1 \mapsto x_1'' \\ x_2 \mapsto x_1' & x_2 \mapsto \emptyset \\ x_3 \mapsto x_2' & x_3 \mapsto x_2'' \end{array} \quad (4.7)$$

Une analyse rapide permet de déterminer que  $\psi_1$  est invariante par rapport à  $\mathcal{F}$  et que  $\psi_2$  ne l'est pas, puisque :

$$\mathcal{F}'_1(\mathcal{X}') = \begin{cases} x_1'^+ = f_2(x_1', x_2', b, c) \\ x_2'^+ = f_3(x_1', x_2', b, c) \end{cases}$$

est une application sur  $\mathcal{X}'$  et

$$\mathcal{F}'_2(\mathcal{X}'') = \begin{cases} x_1''^+ = f_1(x_1'', x_2, x_2'', a, b) \\ x_2''^+ = f_3(x_2, x_2'', b, c) \end{cases}$$

n'est pas une application sur  $\mathcal{X}''$  puisque  $x_2 \notin \mathcal{X}''$ . En effet,  $\mathcal{F}'_2$  est définie par

$$\mathcal{F}'_2 : (\mathcal{X}'' \cup \{x_2\}) \times \mathcal{W} \rightarrow \mathcal{X}''$$

et  $x_2$  est un élément externe à  $\mathcal{X}''$

### Mesurabilité

La contrainte de mesurabilité est remplie s'il existe une relation entre les sorties du sous-modèle  $\Sigma'$  issu de la décomposition, et les sorties du modèle complet  $\Sigma$ . Seul un modèle qui satisfait la contrainte d'invariance est susceptible de satisfaire le critère de mesurabilité. Il est possible alors de comparer les sorties du modèle complet avec les sorties du sous-modèle obtenu par décomposition et ainsi appliquer une méthode de détection de défaillances.

Considérons le modèle  $\Sigma(\mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{F}, \mathcal{H})$  et un sous-modèle  $\Sigma'(\mathcal{W}', \mathcal{X}', \mathcal{Y}', \mathcal{F}', \mathcal{H}')$  obtenu par la décomposition  $\psi$ . Les modèles  $\Sigma$  et  $\Sigma'$  sont équivalents si pour toutes les séquences d'entrées  $\mathcal{W}^{12}$ , les sorties de  $\Sigma'$  permettent d'exprimer les sorties de  $\Sigma$  et inversement, c'est-à-dire :

$$\forall w \in \mathcal{W}, x \in \mathcal{X}, x' \in \mathcal{X}' \exists \xi, \zeta : x' = \psi(x) \text{ et } \Leftrightarrow \xi(y') = \zeta(y) \quad (4.8)$$

avec  $y' = \mathcal{H}'(\mathcal{F}'(x', w))$  et  $y = \mathcal{H}(\mathcal{F}(x, w))$  et où  $\xi$  et  $\zeta$  sont des applications. Si  $\mathcal{F}'$  est une restriction de  $\mathcal{F}$  sur  $\mathcal{X}' \times \mathcal{W}$  et si  $\mathcal{H}'$  est une restriction de  $\mathcal{H}$  sur  $\mathcal{X}'$  et  $\mathcal{Y}'$ , la relation (4.8) est satisfaite, et l'existence d'une restriction sur  $\mathcal{X}'$  et  $\mathcal{Y}'$  garantit la mesurabilité. Il reste cependant difficile de démontrer l'existence d'une restriction dans le cas général.

Puisque  $\Sigma'$  satisfait la contrainte d'existence (4.4), alors

$$\mathcal{F}'(\psi(x), w) = \psi(\mathcal{F}(x, w))$$

et la condition (4.8) devient

$$\forall x \in \mathcal{X}, \exists \zeta, \xi : \xi(\mathcal{H}'(\psi(x))) = \zeta(\mathcal{H}(x)) \quad (4.9)$$

En récrivant la relation (4.8) en terme d'ensembles la contrainte devient

$$\exists \zeta, \xi : \xi(\mathcal{H}'(\mathcal{X}')) = \zeta(\mathcal{H}(\mathcal{X})) \quad (4.10)$$

Si la condition (4.10) est satisfaite alors les sorties du sous-modèle  $\Sigma'$  permettent d'exprimer toutes les sorties du modèle  $\Sigma$ , et toutes les sorties  $y \in \mathcal{Y}$  et  $y' \in \mathcal{Y}'$  sont cohérentes<sup>13</sup>. Cependant pour comparer l'évolution de deux modèle  $\Sigma$  et  $\Sigma'$ , il suffit d'avoir une seule sortie cohérente.

La condition de mesurabilité s'écrit alors sous la forme

$$\exists \zeta, \xi : \xi(\mathcal{H}'(\mathcal{X}')) \cap \zeta(\mathcal{H}(\mathcal{X})) \neq \emptyset \quad (4.11)$$

Si la relation (4.11) est satisfaite, alors  $\Sigma'$  satisfait le critère de mesurabilité.

<sup>12</sup>Nous considérons que  $\Sigma$  et  $\Sigma'$  partagent les mêmes entrées, à la différence que  $\Sigma'$  est découplé de certaines entrées, c'est-à-dire que  $\mathcal{W}' \subseteq \mathcal{W}$

<sup>13</sup>Deux grandeurs sont *cohérentes* si elles appartiennent au même ensemble ou espace. Ici  $y$  et  $y'$  sont rendues cohérentes à l'aide des applications  $\xi$  et  $\zeta$

**Illustration** Considérons la relation  $\mathcal{H}$  suivante

$$\begin{aligned}\mathcal{H} : \mathcal{X} &\rightarrow \mathcal{Y} \\ x_1 &\mapsto y_1 \\ x_2 &\mapsto y_2 \\ x_1 + x_2 &\mapsto y_3\end{aligned}\tag{4.12}$$

où  $\mathcal{X} = \{x_1, x_2, x_3, x_4\}$  et  $\mathcal{Y} = \{y_1, y_2, y_3, y_4\}$ . Soit les relations  $\mathcal{H}'$  et  $\mathcal{H}''$  telles que

$$\begin{aligned}\mathcal{H}' : \mathcal{X} &\rightarrow \mathcal{Y}' \\ x_1 &\mapsto y'_1 \\ x_4 &\mapsto y'_2\end{aligned}\tag{4.13}$$

avec  $\mathcal{Y}' = \{y'_1, y'_2\}$ , et

$$\begin{aligned}\mathcal{H}'' : \mathcal{X} &\rightarrow \mathcal{Y}'' \\ x_1 + x_3 &\mapsto y''_1 \\ x_4 &\mapsto y''_2\end{aligned}\tag{4.14}$$

avec  $\mathcal{Y}'' = \{y''_1, y''_2\}$ .

La première relation  $\mathcal{H}'$  est mesurable puisqu'il existe une relation qui lie  $y'_1$  à  $y_1$ . La seconde relation n'est pas mesurable, puisqu'il n'existe pas de relation qui lie les éléments de  $\mathcal{Y}$  avec les éléments de  $\mathcal{Y}''$ . En effet, il n'est pas possible de relier  $y_1$  avec  $y''_1$  si  $\mathcal{H}''(x_3)$  est inconnu.

#### 4.2.4 Algorithme de décomposition avec critère de découplage

L'algorithme de décomposition général (1) est récrit en détaillant les critères et les contraintes de la décomposition. A partir d'ici, l'objectif visé par la décomposition est l'obtention de sous-modèles découplés de sous-ensembles d'entrées inconnues. Les sous-modèles résultants sont utilisés pour synthétiser les structures de génération d'indicateurs de défaillances structurés.

##### Principe de l'algorithme de décomposition

Comme il a été mentionné précédemment, l'algorithme de décomposition présenté est un algorithme *itératif*. Un ensemble  $\mathcal{X}^0 \subseteq \mathcal{X}$  de solutions candidates à la décomposition est déterminé dans l'étape d'initialisation à partir du critère de décomposition choisi, qui est ici un ensemble d'entrées inconnues à découpler. Une boucle itérative permet de construire le sous-ensemble d'états découplés

$$\mathcal{X}_* = \mathcal{X}^1 \cup \mathcal{X}^2 \cup \dots$$

où les  $\mathcal{X}^j \subseteq \mathcal{X}^0$ . La condition de sortie de l'algorithme qui est la réalisation des deux contraintes de décomposition est double :

- Invariance de  $\phi^i$  par rapport à  $\mathcal{F}$ , où le compteur  $i$  représente la  $i^e$  itération.
- Existence d'une fonction  $\mathcal{H}_*$  qui satisfait le critère de mesurabilité par rapport à la fonction de sortie du modèle complet  $\mathcal{H}$ .

Durant la première itération, le sous-ensemble  $\mathcal{X}^1 \subseteq \mathcal{X}^0$  est choisi de manière à respecter la propriété de mesurabilité.  $\mathcal{X}^1$  est un élément choisi dans l'intersection de l'ensemble des états structurellement découplés des entrées  $\mathcal{W}'$  et de l'ensemble des états mesurés. Puisque la cardinalité de l'ensemble  $\mathcal{X}_*$  ne diminue pas au fil des itérations, le sous-modèle obtenu satisfera cette contrainte.

Le reste des itérations vient augmenter l'ensemble  $\mathcal{X}_*$  en ajoutant les éléments de  $\mathcal{X}^0$  pour répondre à la propriété d'invariance.

Si les conditions de sortie de la boucle sont satisfaites, le reste des éléments du sous-modèle découplé  $\Sigma_*$  est calculé. Si la condition de sortie n'est pas satisfaite et il ne reste plus de solutions candidates, l'algorithme annonce l'échec de la décomposition avec le critère choisi.

### Structure de l'algorithme

Le nouvel algorithme est donné par le pseudo-code (2).

**Entrées :**  $\mathcal{W}' = \gamma \cup \{\dots, \rho_j, \dots\}$  avec  $\rho_j \in \rho$  (défaillances à découpler)

**Données :**  $\Sigma, i$  (compteur)

**Sorties :**  $\Sigma_*, \phi$  (fonction de décomposition)

**début**

Initialisation : Déterminer  $\mathcal{X}^0$  et  $\phi^0$  à partir de  $\mathcal{W}'$  en respectant le critère de découplage;

Choisir  $\mathcal{X}^1 \subseteq \mathcal{X}^0$  et  $\phi^1$  tels que la contrainte de mesurabilité soit respectée ;

$i=2$  ;

**tant que**  $\phi$  n'est pas invariant par rapport à  $\mathcal{F}$  et que  $\mathcal{H}_*$  n'est pas mesurable

**faire**

Incrémenter  $i$  ;

Choisir un sous-ensemble  $\mathcal{X}^i \subseteq \mathcal{X}^0$  tel  $\mathcal{X}_* = \mathcal{X}^i \cup \mathcal{X}^{i-1}$  en respectant les critères de décomposition ;

Déterminer la fonction de découplage  $\phi^i$  correspondante ;

Déterminer  $\mathcal{Y}_*, \mathcal{H}_*$  ;

Déterminer le reste des caractéristiques de  $\Sigma_*$  ;

**fin**

**retourner**  $\Sigma_*$

**Algorithme 2 :** Structure de l'algorithme de décomposition avec critère de découplage

### Discussion de l'algorithme

Deux étapes fondamentales caractérisent l'algorithme 2 :

- la recherche de l'ensemble des solutions candidates. Ces solutions sont déterminées à partir du critère de décomposition.
- la sélection des solutions qui satisfont les contraintes de décomposition.

Les deux étapes sont d'égale importance. La première étape construit une *base* à partir de laquelle le système découplé est synthétisé. La seconde étape sélectionne dans cette base les solutions qui contribuent à l'invariance et à la mesurabilité du sous-modèle.

Cet algorithme est susceptible d'être amélioré : Une optimisation de la première étape, appelée à partir de maintenant étape d'*initialisation*, permet d'améliorer la probabilité de réussite de la décomposition. De la même manière, un relâchement des contraintes de décomposition permet de sélectionner une quantité supplémentaire de solutions candidates qui, rappelons le, sont choisies par rapport au critère de décomposition.

Des solutions existent en relâchant les contraintes de décomposition :

- La contrainte de mesurabilité peut être étendue en augmentant le nombre d'états visibles en sortie (mesurés). Cette augmentation est réalisée par exemple en augmentant le nombre de capteurs sur un système réel. Mais il n'existe pas à notre connaissance de moyens analytiques pour résoudre ce problème.
- Le relâchement de la contrainte d'invariance est un problème connu ; citons par exemple l'ouvrage [Isi95] qui traite entre autres de l'injection de sorties pour le découplage de systèmes non linéaires en utilisant des outils de géométrie différentielle.

Dans la section suivante, une contrainte étendue d'invariance est définie. Ce point permet de proposer une extension de l'algorithme (2).

#### 4.2.5 Extension de l'algorithme de décomposition par injection de sorties

Les sorties de  $\Sigma$  sont utilisées pour compenser la perte d'informations qui résulte de la réduction de l'ensemble d'états imposée par la décomposition.

**Illustration** Pour illustrer ce point, considérons la fonction d'état  $f$  vectorielle suivante :

$$\begin{cases} x_1^+ = f_1(x_1, x_2, x_3) \\ x_2^+ = f_2(x_2) \\ x_3^+ = f_3(x_3) \\ x_4^+ = f_4(x_4) \end{cases} \quad \text{avec } x = [x_1 \ x_2 \ x_3 \ x_4]^T \quad (4.15)$$

et supposons qu'une mesure de l'état  $y = h(x) = x_3$  est disponible. La fonction

$$\phi(x) = [1 \ 1 \ 0 \ 0]x$$

n'est pas invariante par rapport à  $f$ , puisqu'elle définit un sous-modèle avec  $\{x_1, x_2\}$  pour ensemble d'états, ce qui correspond à  $\mathbb{R}^2$ , et que  $x_3$  reste inconnu. Par contre,  $\phi$  est invariante par rapport à  $(f, h)$  puisque  $x_3$  est connu à travers  $y$ . En utilisant une injection de sortie  $y$  dans la première équation, la décomposition par  $\phi$  devient possible et donne

$$\begin{cases} x_1'^+ = f_1(x_1', x_2', y) \\ x_2'^+ = f_2(x_2') \end{cases} \quad \text{avec } x' = [x_1' \ x_2']^T \quad (4.16)$$

La décomposition étendue par l'utilisation de l'injection de sorties produit un sous-modèle  $\Sigma_*$  défini par le quintuplet  $(\mathcal{X}_*, \mathcal{W}_*^e, \mathcal{Y}_*, \mathcal{F}_*, \mathcal{H}_*)$ .  $\mathcal{W}_*^e$  est donnée par

$$\mathcal{W}_*^e = (\mathcal{W} - \mathcal{W}') \cup \mathcal{Y}$$

où  $\mathcal{W}'$  regroupe les entrées à découpler. Tous les états ou les combinaisons d'état qui n'appartiennent pas à  $\mathcal{X}_*$  sont remplacées par des mesures qui permettent de les reconstituer. L'utilisation de l'injection de sorties est donc limitée au cas où les mesures permettent de reconstituer les informations inconnues.

En conclusion, l'injection de sorties permet d'obtenir une contrainte d'invariance moins stricte sous certaines conditions. La section suivante est consacrée à la formalisation de cet aspect.

### Extension de la contrainte d'invariance par injection de sorties

La contrainte d'invariance par rapport à  $\mathcal{F}$  est étendue à l'invariance par rapport à  $(\mathcal{F}, \mathcal{H})$ . La relation (4.4) devient

$$\forall w \in \mathcal{W}, \forall x \in \mathcal{X}, \exists y \in \mathcal{Y} : x' = \psi(x) \Leftrightarrow \mathcal{F}'(x', y, w) = \psi(\mathcal{F}(x, w)) \quad (4.17)$$

en remplaçant  $\psi$  par la fonction de découplage  $\phi$ , cette relation s'écrit

$$\forall w \in \mathcal{W}, \forall x \in \mathcal{X}, \exists y \in \mathcal{Y} : x_* = \phi(x) \Leftrightarrow \mathcal{F}_*(x', y, w) = \phi(\mathcal{F}(x, w)) \quad (4.18)$$

### Structure de l'algorithme étendu

L'utilisation du critère étendu permet de formuler l'algorithme de décomposition avec critère de découplage sous la forme suivante (algorithme 3)

La modification principale apparaît au niveau de la condition de sortie de la boucle, ainsi que dans la structure du sous-modèle découplé obtenu. Le sous-modèle possède un ensemble d'entrées augmenté des mesures d'états relevées sur le modèle complet.

## 4.3 Décomposition des modèles comportementaux temporels

### 4.3.1 Introduction

Cette section est consacrée à la description de la méthodologie de la décomposition de systèmes représentés par des modèles d'évolution temporels. L'algorithme de décomposition (3) est réécrit en utilisant les outils de l'algèbre des fonctions. Pour éviter la répétition, nous nous limiterons ici à l'étude du critère de découplage et à la

**Entrées :**  $\mathcal{W}' = \gamma \cup \{\dots, \rho_j, \dots\}$  avec  $\rho_j \in \rho$  (défaillances à découpler)

**Données :**  $\Sigma, i$  (compteur)

**Sorties :**  $\Sigma_*, \phi$  (fonction de décomposition)

**début**

Initialisation : Déterminer  $\mathcal{X}^0$  et  $\phi^0$  à partir de  $\mathcal{W}'$  en respectant le critère de découplage;

Choisir  $\mathcal{X}^1 \subseteq \mathcal{X}^0$  et  $\phi^1$  tels que la contrainte de mesurabilité est respectée ;

$i=1$  ;

**tant que**  $\phi$  n'est pas invariant par rapport à  $(\mathcal{F}, \mathcal{H})$  et que  $\mathcal{H}_*$  n'est pas mesurable

**faire**

    Incrémenter  $i$  ;

    Choisir un sous-ensemble  $\mathcal{X}^i \subseteq \mathcal{X}^0$  tel que  $\mathcal{X}_* = \mathcal{X}^i \cup \mathcal{X}^{i-1}$  en respectant les critères de décomposition ;

    Déterminer la fonction de découplage  $\phi^i$  correspondante ;

    Déterminer  $\mathcal{Y}_*, \mathcal{H}_*$  ;

    Déterminer le reste des caractéristiques de  $\Sigma_*$  ;

**fin**

**retourner**  $\Sigma_*$

**Algorithme 3 :** Structure de l'algorithme étendu de décomposition avec critère de découplage

recherche du sous-modèle découplé  $\Sigma_*$ . La méthodologie de détermination du sous-modèle couplé  $\Sigma'$  est en tout point semblable.

La section est organisée en deux parties principales :

- une première partie où l'algorithme est retranscrit en utilisant l'algèbre des fonctions,
- une seconde partie où des améliorations sont proposées : les aspects calculatoires de l'algèbre des fonctions sont détaillés et l'étape d'initialisation de l'algorithme est étendue en utilisant des techniques d'élimination de variables.

Des illustrations linéaires [BCC05] viennent agrémenter chaque partie pour aider à la compréhension.

### 4.3.2 Objectif de la décomposition

L'objectif de la décomposition est d'obtenir des sous-modèles découplés de sous-ensembles d'entrées inconnues. Les sous-ensembles de découplage sont déterminés suivant le principe décrit dans le chapitre 2. Ces sous-modèles découplés serviront ensuite à la synthèse de générateurs de résidus structurés.

Pour réaliser cet objectif, les sous-modèles découplés doivent en outre vérifier des critères de couplage par rapport aux défaillances à détecter.

Considérons le système représenté par le modèle analytique suivant :

$$\Sigma : \begin{cases} \dot{x} = f(x, w) \\ y = h(x) \end{cases}, w = \begin{bmatrix} u \\ \rho \\ \gamma \end{bmatrix} \quad (4.19)$$

où  $x \in X \subseteq \mathbb{R}^n$ ,  $w \in W \subseteq \mathbb{R}^l$  et  $y \in Y \subseteq \mathbb{R}^m$  représentent respectivement les vecteurs d'état, d'excitation et de sortie. Le vecteur d'excitation est composé de trois vecteurs :  $u \in U \subseteq \mathbb{R}^l$  le vecteur de commandes,  $\gamma$  le vecteur de perturbations et  $\rho$  le vecteur de défaillances.

Le modèle  $\Sigma$  est décomposable en deux parties :

- une partie découplée d'un vecteur  $\gamma$  et couplée au vecteur  $\rho_i$ ,
- une seconde partie qui décrit le reste du modèle  $\Sigma$  et qui est couplée à  $\gamma$ .

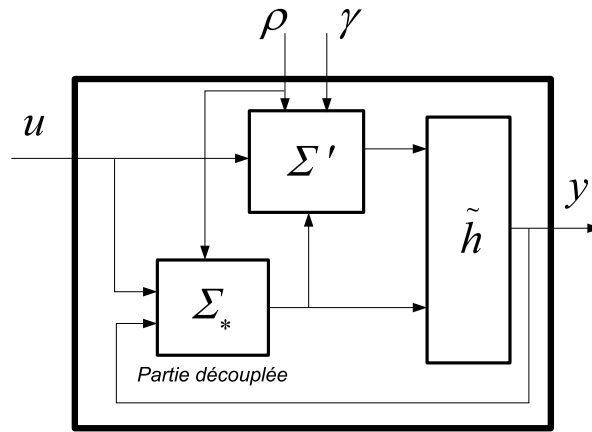


FIG. 4.2 – Décomposition parallèle du modèle  $\Sigma$

Le sous-modèle  $\Sigma_*$  représente la dynamique découplée de l'influence du vecteur de perturbations  $\gamma$

$$\Sigma_* : \begin{cases} \dot{x}_* = f_*(x_*, y, u, \rho_i) \\ y_* = h_*(x_*) \end{cases} \quad (4.20)$$

Cette décomposition est réalisée selon les techniques expliquées dans la section précédente. En vertu de la contrainte de mesurabilité, les sorties du sous-modèle découplé  $y_*$  peuvent s'exprimer en fonction des sorties du système complet

$$y_* = h_*(x_*) = c(y) = c(h(x)) \quad (4.21)$$

où  $c$  est une fonction quelconque.

La fonction de découplage  $\phi$  est une fonction différentiable qui s'écrit

$$\begin{aligned} \phi &: X \rightarrow X_* \\ &: x \mapsto x_* = \phi(x) \end{aligned} \quad (4.22)$$

où  $X$  et  $X_*$  sont les espaces d'état respectifs de  $\Sigma$  et  $\Sigma_*$ .



**Remarque 4.3.1** Le sous-modèle couplé  $\Sigma'$  est complémentaire au sous-modèle  $\Sigma_*$ . Pour déterminer le sous-modèle découplé, une fonction de décomposition  $\psi$  est déterminée, par analogie avec la section précédente.

### 4.3.3 Méthodologie de décomposition d'un modèle temporel

Les critères et les contraintes de décomposition sont décrits en utilisant les opérateurs de l'algèbre des fonctions ainsi que l'algorithme de décomposition étendu (3). Ces critères sont appliqués aux modèles non linéaires. Des exemples de décomposition de modèles linéaires sont donnés pour illustrer chaque notion.

#### Critères de décomposition

La décomposition s'effectue suivant deux critères : un critère de découplage par rapport à  $\gamma$  et un critère de couplage au vecteur  $\rho$ .

#### Le critère de découplage

Le sous-ensemble d'états découplé de  $\gamma$  est calculé en utilisant l'opérateur  $m$  de l'algèbre des fonctions :

$$m_{W-X}(\gamma) = X^0 \quad (4.23)$$

où  $X^0$  est le plus grand sous-ensemble d'états qui satisfait le critère de découplage par rapport à  $\gamma$ . La notation  $W - X$  indique une paire « Entrée-État ». La fonction vectorielle  $\phi^0 \in D_X$ <sup>14</sup> est la fonction par laquelle  $X^0$  est l'image de  $X$ , c'est-à-dire

$$X^0 = \phi^0(X)$$

La fonction  $\phi^0$  décrit le plus grand sous-ensemble de  $X$  qui est découplé de  $\gamma$ . Puisque  $x^0 = \phi^0(x)$  et que

$$\dot{x}^0 = \frac{d\phi^0(x)}{dt} = \frac{d\phi^0}{dx} \frac{dx}{dt} = \frac{d\phi^0}{dx} f(x, w)$$

alors la relation  $\frac{\partial}{\partial \gamma} f^0(x^0, y, u, \rho) = 0$  implique que

$$\frac{\partial}{\partial \gamma} \frac{d\phi^0}{dx} f(x, w) = 0 \quad (4.24)$$

Toutes les fonctions  $\phi_* \geq \phi^0$  donneront des images de  $X$  qui sont contenues dans  $X^0$ .  $\phi^0$  est la plus *petite* fonction de l'ensemble de fonctions qui satisfait le critère de découplage.

$$\phi^0 \leq \phi_* \Leftrightarrow \phi_* \in D_X \text{ et } \phi_*(X) \subseteq X^0 \Rightarrow \frac{d\phi_*}{dx} \frac{\partial f}{\partial \gamma} = 0 \quad (4.25)$$

<sup>14</sup> $D_X$  est l'ensemble de toutes les fonctions possible sur  $X$

où  $\leq$  et  $\geq$  sont les opérateurs d'ordre de l'algèbre des fonctions. Le critère de découplage s'écrit donc

$$\phi^0 \leq \phi \quad (4.26)$$

où  $\phi$  est la fonction de découplage et  $\phi^0$  la fonction vectorielle dont les composantes scalaires forment l'ensemble de fonctions candidates au découplage.

### Le critère de couplage

A l'issu de la décomposition, il est nécessaire de vérifier si le sous-modèle  $\Sigma_*$  reste couplé au sous-ensemble d'entrées  $\rho$ .

$$m_{X-W}(\rho) = X_\rho \quad (4.27)$$

où  $X_\rho$  est le plus grand sous-ensemble d'états qui satisfait le critère de couplage par rapport à  $\rho$ . La manière la plus directe de vérifier le couplage est d'utiliser la relation

$$X_* \cap X_\rho \neq \emptyset$$

où  $X_*$  est l'ensemble d'états de  $\Sigma_*$ . Afin de ne pas alourdir l'algorithme de décomposition, une méthode indirecte est choisie :

Soit le sous-ensemble d'entrées découplé de l'ensemble d'entrées  $\rho$  et qui est donné par le vecteur

$$\mathcal{W}_{\bar{\rho}} = \begin{bmatrix} u \\ \gamma \end{bmatrix}$$

Le sous-ensemble d'états associé est calculé par la relation

$$m_{X-W}(\mathcal{W}_{\bar{\rho}}) = X_{\bar{\rho}} \quad (4.28)$$

Le couplage est vérifié si la relation suivante est satisfaite

$$X_* \not\subseteq X_{\bar{\rho}}$$

Ce critère est traduit maintenant en termes d'algèbre des fonctions :

Soit  $\phi_\rho$  la fonction qui décrit le plus grand sous-ensemble d'états  $X_{\bar{\rho}}$  découplé de  $\rho$ .

$$\forall \beta \in D_X : \phi_\rho \leq \beta \Leftrightarrow \frac{d\beta}{dx} \frac{\partial f}{\partial \rho} = 0 \quad (4.29)$$

Puisque  $\frac{\partial}{\partial \rho} f_*(x_*, y, u, \rho) \neq 0$  alors

$$\frac{\partial}{\partial \rho} \frac{d\phi}{dx} f(x, u, \gamma, \rho) \neq 0 \quad (4.30)$$

Par conséquent

$$\phi_\rho \not\leq \phi \Leftrightarrow \frac{\partial}{\partial \rho} \frac{d\phi}{dx} f(x, u, \gamma, \rho) \neq 0 \quad (4.31)$$

Le critère de couplage par rapport au sous-ensemble d'entrées  $\rho$  s'écrit alors

$$\phi_\rho \preceq \phi \quad (4.32)$$

où  $\phi$  est la fonction de découplage par rapport à  $\gamma$  et  $\phi_\rho$  est la plus grande fonction qui satisfait

$$\frac{\partial}{\partial \rho} \frac{d\phi_\rho}{dx} f(x, u, \gamma, \rho) = 0$$

**Illustration** Si le modèle linéaire  $\Sigma$  exprimé par

$$\Sigma : \begin{cases} \dot{x} &= Ax + Bu + E\gamma + K\rho \\ y &= Cx \end{cases} \quad (4.33)$$

alors nous cherchons à déterminer le sous-modèle découplé

$$\Sigma : \begin{cases} \dot{x}_* &= A_*x_* + B_*u + Jy \\ y_* &= C_*x_* \end{cases} \quad (4.34)$$

où les matrices  $A_*$ ,  $B_*$ ,  $J_*$  et  $C_*$  sont à déterminer. La fonction de découplage est notée  $\Phi$  et permet de passer du modèle  $\Sigma$  au sous-modèle  $\Sigma_*$ .

Soit  $\Phi_\rho$  la matrice de dimension maximale qui satisfait la relation

$$\frac{d(\Phi_\rho x)}{dx} \frac{\partial}{\partial \rho} (Ax + Bu + E\gamma + K\rho) = \Phi_\rho K = 0 \quad (4.35)$$

où  $K$  est la matrice des entrées dont on désire garder le couplage. Le critère de couplage  $\Phi_\rho \preceq \Phi$  s'écrit  $\Phi K \neq 0$  puisque

$$\Phi_\rho \preceq \Phi \Leftrightarrow \Phi_\rho K \preceq \Phi K \Leftrightarrow \Phi K \succeq 0 \Leftrightarrow \Phi K \neq 0 \quad (4.36)$$

De la même manière, soit  $\Phi^0$  la matrice de dimension maximale qui satisfait la relation

$$\frac{d(\Phi^0 x)}{dx} \frac{\partial}{\partial \gamma} (Ax + Bu + E\gamma + K\rho) = \Phi^0 E = 0 \quad (4.37)$$

où  $E$  est la matrice des entrées à découpler. Le critère de découplage  $\Phi^0 \leq \Phi$  s'écrit  $\Phi E = 0$  puisque

$$\Phi^0 \leq \Phi \Leftrightarrow \Phi^0 E \leq \Phi E \Leftrightarrow \Phi E \geq 0 \Leftrightarrow \Phi E = 0 \quad (4.38)$$

## Les contraintes de décomposition

### Contrainte d'existence : la propriété d'invariance

La contrainte d'invariance stipule que la restriction de la fonction d'état  $\mathcal{F}$  sur le sous-ensemble  $\mathcal{X}_* = \phi(\mathcal{X})$  (l'image de  $\mathcal{X}$  par la fonction de découplage  $\phi$ ) est une application. La fonction  $\phi$  est alors dite invariante par rapport à  $\mathcal{F}$ . Cela signifie que

la transformation  $\phi$  n'altère pas la fonction  $\mathcal{F}$ . C'est-à-dire que si l'état d'un modèle appartient à l'ensemble  $\mathcal{X}_*$  (ou est défini dans le repère  $\mathcal{X}_*$ ), l'état suivant du modèle appartient également à  $\mathcal{X}_*$ . La *trajectoire* du modèle  $\Sigma$  évolue toujours dans  $\mathcal{X}_*$ , si l'état initial du système appartient à l'espace décrit par  $\mathcal{X}_*$ .

La formulation algébrique et la formulation géométrique permettent toutes les deux de décrire cette notion d'invariance. La formulation géométrique met en évidence les singularités des équations alors que la formulation algébrique apporte un degré de abstraction plus élevé dans un souci de généralité. Dans cette partie, en règle générale, la formulation algébrique est utilisée, avec quelques illustrations géométriques pour clarifier des notions plus délicates. Les deux formulations restent cependant équivalentes, dans le cas où l'approche géométrique est applicable.

L'opérateur  $m$  permet de calculer l'ensemble qui contient l'état futur du système décrit par la fonction d'état  $f$ , si l'état présent du système est connu. L'opérateur  $M$  permet de déterminer l'inverse : en ayant pour argument l'ensemble qui contient l'état présent,  $M$  calcule l'ensemble qui contient l'état précédent du modèle  $\Sigma$ .

Nous allons récrire la contrainte d'invariance en utilisant l'algèbre des fonctions. Soit  $\phi$  une fonction invariante par rapport à la fonction d'état  $f$  d'un modèle  $\Sigma$ . La fonction  $\phi$  permet de déterminer le sous-ensemble  $X_* = \phi(X) \subseteq X$ . La relation  $f_* = (\phi \circ f)$  définie par

$$\begin{aligned} f_* : \phi(X) \times W &\rightarrow \phi(X) \\ (\phi(x), w) &\mapsto \phi(f(x, w)) \end{aligned} \quad (4.39)$$

est une fonction. La définition (4.39) peut s'interpréter également comme

$$(X_* \cup W) \subseteq (\phi \circ f)(X, W)$$

Puisque  $X_* = \phi(X)$  alors

$$(\phi(X) \cup W) \subseteq (\phi \circ f)(X, W) \quad (4.40)$$

En utilisant la dualité entre les fonctions et les ensembles (chapitre 3), la relation (4.40) s'écrit

$$\Pi_{X \times W}(\phi(x)) \leq \phi(f(x, w)) \quad (4.41)$$

où le symbole  $\Pi$  représente la projection. La relation (4.41) correspond à la définition de la paire algébrique par rapport à  $f$  (3.32). La contrainte d'invariance s'écrit alors

$$(\phi, \phi) \in \Delta_f \quad (4.42)$$

Ce qui peut s'interpréter : La partition de l'ensemble  $X$  définie par la fonction  $\phi$  possède la propriété de substitution.

En suivant le même raisonnement, la contrainte d'invariance étendue s'écrit

$$(\phi \otimes h, \phi) \text{ est une paire algébrique} \quad (4.43)$$

Toute fonction  $\phi$  qui satisfait cette contrainte permet d'obtenir un sous-modèle découplé réalisable  $\Sigma_*$ . Par définition le binôme  $(M(\phi), \phi)$  forme une paire algébrique et  $M(\phi)$  est la plus grande fonction possible qui satisfait cette condition. Par superposition avec (4.43), si  $\phi$  est invariante par rapport à  $(f, h)$  alors

$$M(\phi) \geq \phi \otimes h \quad (4.44)$$

Par analogie, si  $(\phi \otimes h, \phi) \in \Delta_f$  alors par définition

$$\phi \geq m(\phi \otimes h) \quad (4.45)$$

Par conséquent, si les relations (4.45) et (4.44) sont satisfaites alors  $(\phi \otimes h, \phi)$  forme une paire algébrique et la contrainte d'existence du sous-modèle est satisfaite.

**Illustration** Soit le modèle linéaire décrit par (4.33). La contrainte d'invariance s'écrit

$$\Phi \otimes C \leq M(\Phi) \quad (4.46)$$

où  $\Phi(x)$  est la fonction de découplage et  $C$  la fonction de sortie. Puisque  $\Phi$  et  $C$  sont des matrices et si  $\Phi$  et  $C$  sont linéairement indépendants, alors

$$\Phi \otimes C = \Phi \otimes C \begin{bmatrix} \Phi \\ C \end{bmatrix}$$

puisque  $[1 \mid 0][\Phi \mid C]^T = \Phi$  et  $[0 \mid 1][\Phi \mid C]^T = C$ . Si  $\Phi$  et  $C$  ne sont pas linéairement indépendants alors  $\Phi \otimes C$  est une matrice composée des lignes linéairement indépendantes de  $\Phi$  et de  $C$  [BCC05].

Par ailleurs,  $M(\Phi)$  devient  $M(\Phi) = \Phi A$ , puisque par définition

$$(\exists N, Z : NM(\Phi)X + ZU \leq \frac{d\Phi X}{dX}(AX + BU)$$

En négligeant les arguments qui dépendent de  $U$ , la relation précédente devient  $M(\Phi)X = \Phi AX$  et par conséquent  $M(\Phi) = \Phi A$ .

La contrainte (4.46) s'écrit alors

$$N' \begin{bmatrix} \Phi \\ C \end{bmatrix} = \Phi F \quad (4.47)$$

où  $N' \in D_X$  est une fonction dont il faut démontrer l'existence. Cette relation peut être comparée avec la condition d'invariance classique pour un modèle linéaire [Pat94] et qui s'écrit :

$$\Phi A = A_* \Phi + J C \quad (4.48)$$

avec  $A_*$  et  $J$  définis dans l'équation (4.34). En posant

$$N = [A_* J]$$

les équations (4.47) et (4.48) deviennent identiques, ce qui confirme le résultat.

**Influence du type de modèle sur le calcul du critère d'invariance**

Si le modèle  $\Sigma$  s'écrit sous la forme

$$x(t+1) = f(x(t), w(t)) \quad (4.49)$$

alors le critère d'invariance étendu (4.43) s'interprète par la relation

$$\forall x \in X, \forall w \in W, \forall y \in Y, \exists v \in D_X : v(\phi(x), y, u) = m(\phi)f(x, w) \quad (4.50)$$

par contre, si  $\Sigma$  s'écrit

$$\dot{x}(t) = f(x(t), w(t)) \quad (4.51)$$

l'équation (4.43) s'écrit alors

$$\forall x \in X, \forall w \in W, \forall y \in Y, \exists v \in D_X : v(\phi(x), y, u) = \frac{dm(\phi)}{dx} f(x, w) \quad (4.52)$$

**Contrainte de mesurabilité**

La contrainte de mesurabilité vérifie que le sous-modèle découplé  $\Sigma_*$  génère des sorties qu'il est possible d'exprimer en fonction des sorties du modèle complet. Nous partons du principe que tous les états du sous-modèle découplé sont accessibles par mesures, c'est-à-dire que la fonction de sorties  $h_*$  est choisie égale à la matrice identité.

La relation entre les états du sous-modèle découplé et les mesures s'écrit

$$\exists \xi, \exists \zeta : \xi(h_*(x_*)) = \zeta(h(x)) \quad (4.53)$$

ce qui s'écrit en termes d'algèbre des fonctions

$$\exists \xi, \exists \zeta : \xi \circ h_* \cong \zeta \circ h \quad (4.54)$$

Puisque tout l'ensemble d'états de  $\Sigma_*$  est mesuré, alors  $h_*$  est une fonction identité. La relation (4.54) devient

$$\exists \xi, \exists \zeta : \xi \circ \psi \cong \zeta \circ h \quad (4.55)$$

Si la relation (4.55) est satisfaite, alors toutes les sorties de  $\Sigma$  et de  $\Sigma_*$  sont cohérentes.

Pour que le critère de mesurabilité soit satisfait, il suffit de démontrer qu'il existe au moins une sortie de  $\Sigma_*$  qui puisse être exprimée avec les sorties de  $\Sigma$ . Le critère de mesurabilité s'écrit dans ce cas :

$$\phi \square h \neq \text{cste} \quad (4.56)$$

où  $\square$  est l'opération définie dans (3.30) et  $\text{cste}$  est une constante qui représente l'élément nul dans  $D_X$ . La relation (4.56) exprime *la contrainte de mesurabilité* en termes d'algèbre des fonctions.

Le résultat de  $\phi \square h$  est donné par

$$\phi \square h = \mu \circ \phi = \nu \circ h \neq \text{cste} \quad (4.57)$$

où les fonctions  $\mu$  et  $\nu$  sont à déterminer. Ce résultat permet de déterminer la fonction  $\nu$  en choisissant la fonction de sortie du modèle découplé  $\mu = h_*$ .

**Illustration** Soit le modèle linéaire décrit par l'équation (4.34). L'équation (4.57) devient

$$\Phi \square H = H_* \Phi = NH \quad (4.58)$$

où  $\Phi$ ,  $H$ ,  $H_*$  et  $N$  sont des matrices linéaires. L'existence de  $H_*$  et de  $N$  reste à déterminer. Si  $N$  existe alors la contrainte de mesurabilité est satisfaite.

### Implémentation de l'algorithme de décomposition généralisé en utilisant les opérateurs de l'algèbre des fonctions

Avant de passer à la synthèse de l'algorithme de décomposition, il est nécessaire de s'attarder sur la condition d'invariance étendue (4.43). Dans l'état, cette contrainte permet de vérifier l'invariance de la fonction de découplage  $\phi$ . Elle ne permet pas cependant de déterminer cette fonction. La fonction de découplage  $\phi$  doit satisfaire deux conditions principales, le découplage et l'invariance (4.59), pour déterminer un sous-modèle découplé réalisable. Ces conditions s'expriment par :

$$\begin{cases} \phi^0 \leq \phi \\ (\phi \otimes h, \phi) \in \Delta_f \end{cases} \quad (4.59)$$

où  $\phi^0$  représente l'ensemble de solutions candidates. Ensuite, le sous-modèle découplé obtenu vérifiera (ou ne vérifiera pas) les contraintes restantes de *couplage* et de *mesurabilité*. Pour pouvoir vérifier la contrainte d'invariance, les relations (4.50) ou (4.52) doivent être résolues. Nous proposons dans ce qui suit une méthode plus simple pour vérifier la contrainte d'invariance ainsi qu'une méthode systématique qui permet de déterminer, si elle existe, la fonction de découplage  $\phi$  qui répond aux conditions (4.59). C'est la contribution essentielle de l'algèbre des fonctions à la problématique du découplage des modèles comportementaux.

Considérons la contrainte d'invariance (4.42) :

$$(\phi, \phi) \in \Delta_f$$

Soit deux fonctions quelconques  $\xi, \zeta \in D_X$ . A partir des propriétés des opérateurs  $m$  et  $M$ , il est connu que

$$(M(\zeta), \zeta) \in \Delta_f \quad \text{et} \quad (\xi, m(\xi)) \in \Delta_f \quad (4.60)$$

Si  $(\phi, \phi)$  forme une paire algébrique, alors elle satisfait simultanément les deux relations (4.60)

$$(M(\phi), \phi) \in \Delta_f \quad \text{et} \quad (\phi, m(\phi)) \in \Delta_f \quad (4.61)$$

Ce qui est vrai si et seulement si les relations suivantes sont vraies<sup>15</sup> :

$$M(\phi) \geq \phi \text{ et } \phi \geq m(\phi) \quad (4.62)$$

Si les relations (4.62) sont vérifiées alors  $\phi$  est invariante par rapport à  $f$ .

Le problème du découplage se résume alors à trouver la fonction  $\phi \geq \phi^0$  qui remplit les conditions  $m(\phi) \leq \phi$  et  $M(\phi) \geq \phi$ . Cette formulation permet de simplifier la recherche de  $\phi$  puisque il n'est plus nécessaire de résoudre les relations (4.50) ou (4.52).

Le cas de l'invariance étendue est similaire, pour raccourcir l'exposé, seuls les résultats sont présentés :

**THÉORÈME 4.1** *Une fonction  $\phi$  de  $D_X$  est invariante par rapport à  $(f, h)$  si elle remplit les conditions suivantes :*

- $m(\phi \otimes h) \leq \phi$ ,
- $M(\phi) \geq \phi \otimes h$ .

Ce résultat sert de base à la synthèse de l'algorithme de décomposition.

Le théorème 4.2 introduit par A.Ye. Shumsky [Shu91] permet de déterminer la fonction  $\phi$  qui répond aux critères

- $\phi \geq \phi^0$ ,
- $m(\phi \otimes h) \leq \phi$ .

**THÉORÈME 4.2** *Soit la relation de récurrence suivante*

$$\phi^{i+1} = m(\phi^i \otimes h) \square \phi^i$$

avec  $i = 0, 1, \dots, n$ . Si  $\phi^{k+1} \equiv \phi^k$  pour une étape quelconque  $k$ , alors  $\phi^* = \phi^k$  est la fonction minimale<sup>16</sup> qui remplit

$$\begin{cases} \phi^0 \leq \phi^* \\ m(\phi^* \otimes h) \leq \phi^* \end{cases} \quad (4.63)$$

**Remarque 4.3.2** *Si le modèle  $\Sigma$  s'écrit sous la forme (4.49) :*

$$x(t+1) = f(x(t), x(t))$$

alors la fonction  $\phi^*$  fournie par le théorème 4.2, répond à la contrainte d'invariance et de découplage, car dans ce cas précis l'opérateur  $M$  est monotone et la relation

$$M(\phi) \geq \phi \otimes h \quad (4.64)$$

est vérifiée automatiquement. Par contre si  $\Sigma$  s'écrit sous la forme (4.51) :

$$\dot{x}(t) = f(x(t), x(t))$$

alors la relation (4.64) n'est plus garantie, car  $M$  n'est pas toujours monotone.



**Entrées :**  $\phi_*$

**Données :**  $h$

Initialisation :  $\phi^1 \geq \phi_*$  où  $\phi^1$  est une composante élémentaire de  $\phi_*$  ;

**début**

<b>tant que</b> $h \otimes \phi^1 \otimes \dots \otimes \phi^j \not\leq M(\phi^j)$ <b>faire</b>
Déterminer $\phi^{j+1}$ tel que
– $\phi^{j+1} \geq \phi_*$ ;
– $h \otimes \phi^1 \otimes \dots \otimes \phi^j \otimes \phi^{j+1} \leq M(\phi^j)$ ;
$j=j+1$ ;
Incrémenter $j$ ;
$\phi = \phi^1 \otimes \dots \otimes \phi^j$ ;

**fin**

**Sorties :**  $\phi$

**Algorithme 4 :** Détermination de  $\phi$  à partir de  $\phi_*$

L'étape suivante consiste à vérifier la relation (4.64)

$$M(\phi) \geq \phi \otimes h$$

L'algorithme suivant permet de déterminer la fonction  $\phi$ , telle que  $\phi \geq \phi_*$ , qui satisfait la relation (4.64) :

La fonction  $\phi$  ainsi obtenue est découplée du sous-ensemble d'entrées  $W'$  et invariante par rapport à la fonction d'état  $f$ .

L'algorithme complet de décomposition est obtenu en incorporant la contrainte de mesurabilité et en testant le sous-modèle obtenu pour vérifier le couplage. La mesurabilité intervient comme une étape intermédiaire entre le calcul des fonctions  $\phi^*$  et  $\phi$ . La mesurabilité de  $\phi^*$  est vérifiée et l'algorithme 4 est initialisé en choisissant  $\phi^1 = \phi^* \square h$

L'algorithme complet est décrit par le pseudo-code 5.

A l'issue de l'algorithme (5), si le critère de couplage n'est pas rempli, il est nécessaire de relâcher le critère de découplage. Dans le cas contraire les critères de couplage et de découplage ne peuvent pas être remplis simultanément.

**Remarque 4.3.3** *La décomposition d'un modèle linéaire  $\Sigma$  (4.33) peut-être réalisée avec plus de facilité que par la méthode décrite par l'algorithme 5 [BCC05]. En utilisant une procédure basée sur la mise sous forme canonique de Kronecker [FW89] et la méthode de synthèse de filtres présentée dans [Mir79], l'algorithme 5 dans le cas spécifique des systèmes linéaires s'écrit comme suit (algorithme 6).*

*La structure de l'algorithme 6 diffère de l'algorithme dans le cas général. En effet la première itération sert à déterminer la matrice d'injection de sorties. Ce résultat est ensuite utilisé dans la seconde itération pour déterminer la matrice  $\Phi$  selon la méthode générale.*

*Le sous-modèle  $\Sigma_*$  obtenu est découplé des entrées  $\gamma$  et couplé aux entrées  $\rho$ . Notons que l'algorithme original reste valable est donne un résultat équivalent.*

<sup>15</sup>voir propriétés de l'algèbre des fonctions au chapitre 3

<sup>16</sup>Rappelons que plus une fonction est de dimension réduite, plus le sous-ensemble qu'elle décrit est grand.

**Entrées :**  $\Sigma(X, W, Y, f, h)$

**Données :**  $i = 0, j = 0$  (compteurs)

**Sorties :**  $\phi$  (fonction de décomposition)

**début**

$\phi^0 = m_{W-X}(\gamma')$  où  $\gamma'$  est la fonction qui réalise  $\gamma'(W) = \gamma$  ;

$\beta^0 = m_{W-X}(\rho')$  où  $\rho'$  est la fonction qui réalise  $\rho'(W) = W - \rho$  ;

$\xi^0 = \phi^0$  **tant que**  $\xi^i \not\leq \xi^{i-1}$  **faire**

    |  $\xi^{i+1} = m(\xi^i \otimes h) \square \xi^i$  ;

**fin**

$\phi_* = \xi^i$  ;

$\phi^1 = \phi^* \square h = v \circ h$  ;

$j = 1$  ;

**tant que**  $h \otimes \phi^1 \otimes \dots \otimes \phi^j \not\leq M(\phi^j)$  **faire**

    | Prendre la plus grande fonction  $\phi^{j+1}$  telle que

        -  $\phi_* \leq \phi^{j+1}$  ;

        -  $h \otimes \phi^1 \otimes \dots \otimes \phi^{j+1} \leq M(\phi^j)$  ;

    | incrémenter  $j$  ;

**fin**

$\phi = \phi_1 \otimes \phi_2 \otimes \dots \otimes \phi_j$  ;

**si**  $\phi \geq \beta^0$  **alors**

    | **retourner** Critère de couplage non rempli par  $\phi$

**sinon**

    | **retourner** Critère de couplage rempli par  $\phi$

**fin**

**fin**

**Algorithme 5 :** Algorithme complet de découplage en termes d'algèbre des fonctions

**Données :** A, B, C, E, K

**Sorties :**  $\Phi, A_*, B_*, J, N$

**début**

```

 $\Phi_1 = E \square C = \Psi C ;$ 
 $k = 1 ;$ 
 $j = 1 ;$ 
tant que  $JCF_k \neq J_1 YF_{k-1} + J_2 CF_{k-2} + \dots + J_k C$  faire
  | incrémenter  $k$  ;
fin
tant que  $j \neq k - 1$  faire
  |  $\Phi_{j+1} = \Phi_j A - J_j C ;$ 
  | incrémenter  $j$  ;
fin
 $A_* = \Phi A ;$ 
 $B_* = \Phi B ;$ 
 $N = \Phi_1 ;$ 
retourner  $\Sigma_*$ 

```

**fin**

**Algorithme 6 :** Algorithme de décomposition pour un modèle temporel linéaire

#### 4.3.4 Exemple de décomposition de modèles temporels

##### Décomposition d'un modèle linéaire

Un exemple tiré de [CP99] est traité pour illustrer la méthode de décomposition. Sans entrer dans les détails de la modélisation, le processus est représenté par le modèle suivant

$$\begin{cases} \dot{x} = Ax + Bu + E\gamma + K\rho \\ y = Cx \end{cases} \quad (4.65)$$

avec

$$A = \begin{bmatrix} -3.6 & 0 & 0 & 0 \\ 0 & -3.6702 & 0 & 0.0702 \\ 0 & 0 & -36.2588 & 0.2588 \\ 0 & 0.6344 & 0.7781 & -1.4125 \end{bmatrix}, E = \begin{bmatrix} 1 \\ 20.758 \\ 0 \\ 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, K = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (4.66)$$

La première étape est de déterminer la fonction  $\Phi^0$  qui répond au critère de découplage par rapport à  $\gamma$ .

$$\Phi^0 \cdot \begin{bmatrix} 1 & 0 \\ 20.758 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} = [0 \ 0] \Rightarrow \Phi^0 = [-20.7588 \quad 1 \quad 0 \quad 1]$$

Pour la deuxième étape il s'agit de déterminer la matrice d'injection de sorties de  $J$  de dimension suffisante pour satisfaire la contrainte d'invariance (4.46). On pose  $\Phi_1 = \Phi^0 \square C = vCH$  où  $v$  est une fonction à déterminer.

$$\Phi_1 = [-20.7588 \quad 1 \quad 0 \quad 0]$$

-  $J = [j_1 \ j_2 \ j_3]$  : Est ce que  $\Phi_1 F = JC$  ?

$$[74.736 \quad -3.6702 \quad 0.0 \quad 0.0702] \stackrel{?}{=} [j_1 \ j_2 \ j_3 \ 0]$$

Ce qui est impossible. Une ligne supplémentaire est ajoutée à  $J$

-  $J = \begin{bmatrix} j_1 & j_2 & j_3 \\ j_4 & j_5 & j_6 \end{bmatrix}$  : Est ce que  $\Phi_1 F^2 = J_1 CA + J_2 C$  ?

Cette matrice  $J$  satisfait la contrainte d'invariance :

$$J = \begin{bmatrix} a & b & 3.86(-0.0702b - 0.0049) \\ 3.6a - 269.049 & 3.67b + 13.47 & 140.10(-0.0702b - 0.0049) \end{bmatrix}$$

où  $a, b$  sont des réels quelconques. En posant  $a = b = 1$ ,  $J$  devient

$$J = \begin{bmatrix} 1 & 1 & -0.29 \\ -265.45 & 17.14 & -10.52 \end{bmatrix}$$

La matrice de découplage  $\Phi$  est construite suivant l'algorithme 4.46 :

-  $\Phi_1 = [-20.7588 \ 1 \ 0 \ 0]$

-  $\Phi_2 = \Phi_1 A - J_1 C$

$$\Phi_2 = [73.736 \quad -4.67 \quad 0.29 \quad 0.07]$$

Le système découplé  $\Sigma_*$  est donné par

$$\begin{cases} \dot{x}_* &= A_* x_* + B_* u + J y \\ y_* &= C_* x_* = v y \end{cases}$$

avec

$$\begin{aligned} A_* &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} & B_* &= \begin{bmatrix} -20.76 & 1.0 & 0.0 \\ 73.736 & -4.67 & 0.29 \end{bmatrix} & J &= \begin{bmatrix} 1 & 1 & -0.29 \\ -265.45 & 17.1402 & -10.52 \end{bmatrix} \\ C_* &= \begin{bmatrix} 1 & 0 \end{bmatrix} & v &= \begin{bmatrix} -20.758 & 1 & 0 \end{bmatrix} \end{aligned}$$

### Décomposition d'un modèle non linéaire

Considérons le système suivant

$$\begin{cases} \dot{x}_1 = x_2 x_3 + u_1 + \gamma \\ \dot{x}_2 = x_2 x_3 - x_4^2 + u_1 - \rho \\ \dot{x}_3 = x_4^2 - x_4 x_5 + x_6^2 \\ \dot{x}_4 = x_1 x_4 x_5 + x_2 x_3 u_2 \\ \dot{x}_5 = x_4 + x_3 u_3 \\ \dot{x}_6 = x_2 x_3 + x_5 u_2 \end{cases} \quad \begin{cases} y_1 = x_2 \\ y_2 = x_3 \\ y_3 = x_5 \end{cases} \quad (4.67)$$

où  $\gamma$  représente la perturbation, et  $\rho$  représente la défaillance à détecter. En premier lieu, le critère de découplage (4.26) est vérifié

$$\frac{df}{d\gamma} = [1 \ 0 \ 0 \ 0 \ 0 \ 0]^T \quad (4.68)$$

Cela veut dire que les états  $x_2, \dots, x_6$  sont structurellement découplés de la défaillance. La fonction vectorielle  $\phi^0(x)$  est la fonction qui transpose le vecteur d'état initial vers une forme structurellement découplée de la perturbation et qui satisfait la condition (4.24)

$$\phi^0(x) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} x = [x_2 \ x_3 \ x_4 \ x_5 \ x_6]^T \quad (4.69)$$

Par analogie, les états découplés de la défaillance sont regroupés dans le vecteur  $\beta^0(x)$  (4.32)

$$\beta^0(x) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} x = [x_1 \ x_3 \ x_4 \ x_5 \ x_6]^T \quad (4.70)$$

Si la relation  $\beta^0 \not\preceq \phi^0$  est satisfaite, alors le sous-modèle  $\Sigma_*$  est couplé à  $\rho$ .

La détermination de la partie invariante de  $\phi^0$ , notée  $\phi^*$ , est réalisée en utilisant le théorème 4.2

La première itération donne

$$\begin{aligned} \phi^1 &= \phi^0 \square m(\phi^0 \otimes h) \\ &= [x_2 \ x_3 \ x_4 \ x_5 \ x_6]^T \square m([x_2 \ x_3 \ x_4 \ x_5 \ x_6]^T) \\ \phi^1 &= [x_2 \ x_3 \ x_5 \ x_6]^T \end{aligned}$$

Puisque  $\phi^1 \not\equiv \phi^0$ , une seconde itération est nécessaire

$$\begin{aligned}\phi^2 &= \phi^1 \square m(\phi^1 \otimes h) \\ &= \begin{bmatrix} x_2 & x_3 & x_5 & x_6 \end{bmatrix}^T \square m\left(\begin{bmatrix} x_2 & x_3 & x_5 & x_6 \end{bmatrix}^T\right) \\ \phi^2 &= \begin{bmatrix} x_2 + x_3 + \frac{1}{2}x_5^2 & x_6 \end{bmatrix}^T\end{aligned}$$

On remarque que  $\phi^2 \not\equiv \phi^1$ , une troisième itération est nécessaire

$$\begin{aligned}\phi^3 &= \phi^2 \square m(\phi^2 \otimes h) \\ &= \begin{bmatrix} x_2 & x_3 & x_5 & x_6 \end{bmatrix}^T \square m\left(\begin{bmatrix} x_2 & x_3 & x_5 & x_6 \end{bmatrix}^T\right) \\ \phi^3 &= \begin{bmatrix} x_2 + x_3 + \frac{1}{2}x_5^2 & x_6 \end{bmatrix}^T\end{aligned}$$

Puisque  $\phi^2 \equiv \phi^3$ , alors la fonction

$$\phi^* = \phi^3 = \begin{bmatrix} x_2 + x_3 + \frac{1}{2}x_5^2 & x_6 \end{bmatrix}^T$$

La dernière étape du processus de décomposition consiste à vérifier la relation

$$M(\phi) \geq h \otimes \phi$$

avec  $\phi \geq \phi_*$ , en passant par les itérations de l'algorithme 4 :

1. A la première étape, l'algorithme itératif est initialisé avec une fonction qui satisfait la contrainte de mesurabilité :

$$\phi_1 = \phi^* \square h = \begin{bmatrix} x_2 + x_3 + \frac{1}{2}x_5^2 \end{bmatrix}$$

2. La relation  $M(\phi) \geq \phi \otimes h$  est vérifiée

$$h \otimes \phi_1 = \begin{bmatrix} x_2 & x_3 & x_5 \end{bmatrix}^T \not\leq M(\phi_1)$$

La condition n'est pas vérifiée puisque

$$M(\phi_1) = \begin{bmatrix} x_2x_3 & x_5x_3 & x_6^2 \end{bmatrix}^T$$

3. La fonction  $\phi_2$  est choisie telle que

$$\phi_2 = \begin{bmatrix} x_6 \end{bmatrix}$$

pour satisfaire la relation

$$\phi_2 \leq \phi^* \text{ et } h \otimes \phi_1 \otimes \phi_2 \leq M(\phi_1)$$

4. La condition  $M(\phi) \geq \phi \otimes h$  est vérifiée une seconde fois

$$h \otimes \phi_1 \otimes \phi_2 = \begin{bmatrix} x_2 & x_3 & x_5 & x_6 \end{bmatrix}^T \leq M(\phi_2)$$

Cette fois-ci la condition est remplie puisque

$$M(\phi_2) = \begin{bmatrix} x_2 x_3 & x_5 \end{bmatrix}^T$$

5. La fonction de décomposition  $\phi$  est donnée par la relation

$$\phi = \phi_1 \otimes \phi_2 = \begin{bmatrix} x_2 + x_3 + \frac{1}{2}x_5^2 & x_6 \end{bmatrix}^T$$

Finalement, la condition de couplage est vérifiée

$$\phi \square \beta^0 = x_6 \neq \phi$$

puisque  $\beta^0 \not\leq \phi$ , alors le sous-modèle  $\Sigma_*$  est couplé à l'entrée  $\rho$ . La fonction de découplage  $\phi$  est donnée par

$$\phi = \begin{bmatrix} x_2 + x_3 + \frac{1}{2}x_5^2 \\ x_6 \end{bmatrix} \quad (4.71)$$

Le sous-modèle découplé  $\Sigma_*$  est obtenu en appliquant la fonction  $\phi$  sur le système  $\Sigma$  :

$$\Sigma \xrightarrow{\phi} \Sigma_*$$

$$x_* = \phi(x) \longrightarrow \dot{x}_* = \frac{d\phi}{dx} \dot{x}$$

Le sous-modèle  $\Sigma_*$  est donné par

$$\Sigma_* \begin{cases} \dot{x}_{*1} &= \dot{x}_2 + \dot{x}_3 + x_5 \dot{x}_5 \\ &= x_{*2}^2 + y_1 y_2 + u_1 + y_3 y_2 u_3 - \rho \\ \dot{x}_{*2} &= \dot{x}_6 = y_1 y_2 + y_3 u_2 \end{cases} \quad (4.72)$$

avec  $y_* = x_{*1}$  et

$$y_* = c(y) = \phi \square h = y_1 + y_2 + \frac{1}{2}y_3^2$$

### 4.3.5 Aspects calculatoires

#### Objectif

Cette section est consacrée aux aspects calculatoires de l'algorithme 5. Elle est composée de deux parties indépendantes :

1. la première partie détaille le calcul des éléments de l'algèbre des fonctions. Tous les opérateurs et opérations sont revus en détail et des idées nouvelles sont présentées, en particulier pour le calcul de l'opérateur «  $m$  » [BCC06b],
2. la seconde partie est consacrée à l'extension de l'étape d'initialisation de la méthode de décomposition. Des idées pour optimiser le choix de l'ensemble des fonctions candidates  $\phi^0$  sont proposées. Les optimisations sont rendues possibles grâce à l'utilisation des techniques d'élimination de variables [BCCJ06, BCC06a].

Ces deux points permettent d'étendre la méthodologie de décomposition, et de proposer des solutions aux cas qui n'étaient pas éligibles à la décomposition en utilisant l'approche originelle proposée dans [Zhi97]. Par ailleurs, les opérateurs de l'algèbre des fonctions étant relativement nouveaux et peu connus, aucune étude de leur utilisation concrète n'existe. Les méthodes de calcul proposées ici permettent d'utiliser les opérateurs de l'algèbre des fonctions sans rentrer dans les détails théoriques et de construire des algorithmes de calcul faciles à implémenter en langage informatique, en utilisant par exemple un logiciel de calcul symbolique tel que MAXIMA, avec lequel les calculs de ce mémoire ont été effectués.

### Calcul des opérateurs de l'algèbre des fonctions

Les méthodes de calcul des opérateurs de l'algèbre des fonctions sont données dans cette section. Les idées de base sont inspirées de [ZS93]. En outre, des propositions sont faites pour le calcul des différents opérateurs, notamment une méthode alternative proposée pour le calcul de l'opérateur  $m$ .

#### a. Relations d'ordre $\leq$ et $\equiv$

Soit les fonctions indépendantes  $\alpha : X \rightarrow S$  et  $\beta : X \rightarrow T$ . La fonction  $\alpha$  est inférieure ou égale à la fonction  $\beta$  :  $\alpha \leq \beta$  si la relation suivante est satisfaite

$$\text{rang} \left[ \frac{d\alpha}{dx} \right] = \text{rang} \left[ \begin{array}{c} \frac{d\alpha}{dx} \\ \frac{d\beta}{dx} \end{array} \right] \quad (4.73)$$

où  $\frac{d\alpha}{dx}$ ,  $\frac{d\beta}{dx}$  sont les Jacobiennes des fonctions vectorielles  $\alpha$  et  $\beta$ , et rang représente le rang générique.

**Illustration** Soit les fonctions vectorielles  $\alpha(x) = [x_1 x_2 \mid x_3]^T$  et  $\beta(x) = [x_1 x_2 x_3]$  avec  $x = [x_1 \mid x_2 \mid x_3]^T$ . On constate que  $\alpha \leq \beta$  puisque

$$\text{rang} \left( \begin{bmatrix} x_2 & x_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) = \text{rang} \left( \begin{bmatrix} x_2 & x_1 & 0 \\ 0 & 0 & 1 \\ x_2 x_3 & x_1 x_3 & x_1 x_2 \end{bmatrix} \right) = 2 \quad (4.74)$$

Considérons à présent la fonction  $\alpha'(x) = [x_1 x_2 + x_3 \mid x_1 x_2]^T$ . A partir de la relation qui suit



$$\begin{aligned} \text{rang} \left( \begin{bmatrix} x_2 & x_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) &= \text{rang} \left( \begin{bmatrix} x_2 & x_1 & 0 \\ 0 & 0 & 1 \\ x_2 & x_1 & 1 \\ x_2 & x_1 & 0 \end{bmatrix} \right) \\ &= \text{rang} \left( \begin{bmatrix} x_2 & x_1 & 1 \\ x_2 & x_1 & 0 \end{bmatrix} \right) = 2 \end{aligned}$$

il apparaît que  $\alpha \equiv \alpha'$ .

**b. L'opération  $\otimes$**

$\alpha$  et  $\beta$  sont deux fonctions vectorielles telles que  $\alpha : X \rightarrow S$  et  $\beta : X \rightarrow T$ . Si  $\alpha$  et  $\beta$  sont indépendantes alors le résultat de l'opération  $\alpha \otimes \beta$  est donné par

$$\alpha \otimes \beta = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{4.75}$$

Si  $\alpha$  et  $\beta$  sont *dépendantes i.e.*

$$\text{rang} \left[ \frac{d\alpha}{dx} \right] + \text{rang} \left[ \frac{d\beta}{dx} \right] \neq \text{rang} \begin{bmatrix} \frac{d\alpha}{dx} \\ \frac{d\beta}{dx} \end{bmatrix} \tag{4.76}$$

alors le résultat  $\alpha \otimes \beta$  est constitué des composantes de  $\alpha$  et de  $\beta$  qui sont indépendantes (toutes les lignes redondantes sont éliminées).

**c. L'opération  $\square$**

Soit une fonction  $g : X \rightarrow S \cap T$  telle que  $g = \alpha \square \beta$ . Toutes les composantes de  $g$ , notées  $g_i$ , satisfont les relations

$$\text{rang} \begin{bmatrix} \frac{d\alpha}{dx} \\ \frac{dg_i}{dx} \end{bmatrix} = \text{rang} \left[ \frac{d\alpha}{dx} \right] \text{ et } \text{rang} \begin{bmatrix} \frac{d\beta}{dx} \\ \frac{dg_i}{dx} \end{bmatrix} = \text{rang} \left[ \frac{d\beta}{dx} \right] \tag{4.77}$$

où les  $\frac{d}{dx}$  représentent des Jacobiennes. Des équations différentielles partielles sont déduites de ces égalités, pour le calcul de chaque composante  $g_i$  de  $g$  :

$$\begin{cases} N \begin{bmatrix} \frac{d\alpha}{dx} \\ \frac{dg_i}{dx} \end{bmatrix} = \begin{bmatrix} \frac{dg_i}{dx} \\ \frac{d\alpha}{dx} \end{bmatrix} \\ N' \begin{bmatrix} \frac{d\beta}{dx} \\ \frac{dg_i}{dx} \end{bmatrix} = \begin{bmatrix} \frac{dg_i}{dx} \\ \frac{d\beta}{dx} \end{bmatrix} \end{cases} \tag{4.78}$$

où  $N$  et  $N'$  sont des matrices non nulles. La fonction  $g = \alpha \square \beta$  est alors donnée par

$$g = g_1 \otimes \dots \otimes g_i$$

**Illustration** Soit les fonctions  $\alpha = [x_1 x_2 | x_3]^T$  et  $\beta = [x_1 | x_2]^T$ . Le résultat  $\alpha \otimes \beta$  est donné par

$$\alpha \otimes \beta = \delta = [x_1 x_2 | x_3]^T$$

puisque

$$\delta \leq \alpha, \delta \leq \beta \text{ et } [x_1 | x_2]^T \leq [x_1 x_2] \quad (4.79)$$

Le calcul de  $\alpha \square \beta$  passe par la résolution des équations suivantes

$$\begin{cases} N \begin{bmatrix} x_2 & x_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \frac{\partial g_1}{\partial x_3} \\ \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \frac{\partial g_1}{\partial x_3} \end{bmatrix} \\ N' \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \frac{\partial g_1}{\partial x_3} \\ \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \frac{\partial g_1}{\partial x_3} \end{bmatrix} \end{cases} \quad (4.80)$$

et le résultat est donné par

$$\frac{dg_1}{dx} = [x_2 | x_1 | 0] \Rightarrow g = \alpha \square \beta = [x_1 x_2] \quad (4.81)$$

Nous pouvons vérifier l'exactitude de ce résultat en utilisant la relation (4.78)

$$\text{rang} \begin{bmatrix} x_2 & x_1 & 0 \\ 0 & 0 & 1 \\ x_2 & x_1 & 0 \end{bmatrix} = \text{rang} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_2 & x_1 & 0 \end{bmatrix} = 2 \quad (4.82)$$

#### d. Calcul de l'opérateur maximal M

Considérons  $\beta(x) = [\beta_1(x) | \dots | \beta_l(x)]^T$  et  $f(x, u)$  deux fonctions vectorielles où les  $\beta_i(x)$  sont des fonctions scalaires.  $M(\beta)$  est déterminé par la relation

$$M(\beta) = M(\beta_1) \otimes \dots \otimes M(\beta_j) \otimes \dots \otimes M(\beta_l)$$

Pour calculer les différents  $M(\beta_i)$ , la définition (3.34) est utilisée pour obtenir la relation suivante

$$\frac{d\beta_i}{dx} f(x, u) = v(\alpha_1(x), \dots, \alpha_p(x), u) \quad (4.83)$$

Les fonctions scalaires  $\alpha_1(x), \dots, \alpha_p(x)$  avec  $p \in \mathbb{N}$  sont déterminées en utilisant l'équation (4.83).

Supposons que  $v = \sum \alpha_j(x) b_j(u)$  où les  $b_j$  sont des fonctions linéaires. La relation (4.83) devient

$$\frac{d\beta_i}{dx} f(x, u) = \sum \alpha_j(x) b_j(u) \quad (4.84)$$

Pour des fonctions  $b_j$  indépendantes, la fonction vectorielle  $\alpha(x) = \alpha_1(x) \otimes \dots \otimes \alpha_p(x)$  permet d'exprimer l'expression  $\frac{d\beta_i}{dx} f(x, u)$  i.e

$$\alpha \leq \frac{d\beta_i}{dx} f(x, u) \quad (4.85)$$

A partir des relations (4.85) et (4.85),  $\alpha$  est la plus grande fonction qui forme une paire algébrique à gauche avec  $\alpha$  et par conséquent

$$M(\beta_i) = \alpha \quad (4.86)$$

Si les différents  $\beta_j$  sont dépendants alors les fonctions  $\alpha_j$  qui correspondent aux  $\beta_j$  dépendants, ne participent pas au calcul de la fonction  $\alpha$ .

**Remarque 4.3.4** Si la fonction  $v$  de l'équation (4.83) a une forme plus générale, les différentes fonctions  $\alpha_i$  de la relation (4.84) doivent remplir une condition supplémentaire [ZS93]. S'il existe  $r$  valeurs d'entrée  $u = w_1, u = w_2, \dots, u = w_r$  telles que

$$\beta_i(f(x, w_1)) \otimes \beta_i(f(x, w_2)) \otimes \dots \otimes \beta_i(f(x, w_r)) \leq \alpha_i \quad (4.87)$$

pour chaque  $i = 1..p$ , alors

$$M(\beta_i) = \alpha_1(x) \otimes \dots \otimes \alpha_p(x) \quad (4.88)$$

#### e. Calcul de l'opérateur minimal $m$

Le calcul de  $m(\alpha)$  pour une fonction vectorielle  $\alpha(x)$  donnée, se fait en utilisant la définition (3.33) :

$$v(\alpha(x), u) \leq \frac{d m(\alpha)(x)}{dx} f(x, u) \quad (4.89)$$

où  $v$  est une fonction à déterminer. En utilisant la relation (4.73), la relation (4.89) devient

$$\text{rang} \left[ \begin{array}{c} \frac{d\alpha}{dx} \\ \frac{d m_i(\alpha)}{dx} \frac{df}{dx} + f^T \left( \frac{d}{dx} \left( \frac{d m_i(\alpha)}{dx} \right) \right)^T \end{array} \right] = \text{rang} \left[ \frac{d\alpha}{dx} \right] \quad (4.90)$$

où les  $m_i(\alpha)$  sont les composantes ligne de la fonction vectorielle  $m(\alpha)(x)$ . Par analogie avec (4.73), des équation différentielles partielles sont déduites de la relation (4.90), et la détermination de  $m(\alpha)$  passe par leur résolution.

Cet ensemble d'équations peut se révéler difficile à résoudre dans le cas général. D'un autre coté, une solution particulière serait satisfaisante pour calculer  $m$ , et l'hypothèse

$$\frac{d}{dx} \left( \frac{d m_i(\alpha)(x)}{dx} \right) = 0 \quad (4.91)$$

est faite pour obtenir une solution simple (si cette solution simple existe). L'équation à résoudre s'écrit maintenant sous la forme

$$\frac{dm_i(\alpha)}{dx} \frac{df}{dx} = N \frac{d\alpha}{dx} \quad (4.92)$$

pour une matrice  $N$  non nulle à déterminer. La relation (4.92) représente un ensemble d'équations linéaires en  $\frac{\partial m_i(\alpha)}{\partial x_i}$

#### f. Calcul alternatif de l'opérateur $m$

Le cas où la relation (4.92) ne possède pas de solution est délicat, puisque le calcul de l'opérateur «  $m$  » passe alors par la résolution de l'équation (4.90). Pour remédier à cette situation, une solution alternative est proposée.

D'après les propriétés de l'opérateur «  $m$  », pour deux fonctions quelconques  $\alpha$  et  $\alpha'$  de  $D_X$  la relation suivante est vérifiée

$$\alpha \equiv \alpha' \Rightarrow m(\alpha) \equiv m(\alpha') \quad (4.93)$$

Ce qui veut dire que pour deux fonctions équivalentes<sup>17</sup>  $\alpha$  et  $\alpha'$ , les résultats  $m(\alpha)$  et  $m(\alpha')$  sont également équivalents. En utilisant cette propriété, le calcul de l'opérateur «  $m$  » pour un argument qui impose des calculs complexes, peut être substitué par le calcul de «  $m$  » pour un autre argument  $\alpha_{eq}$  équivalent au premier. La complexité des équations (4.90) est alors évitée.

Le choix de la fonction de substitution  $\alpha_{eq}$  doit être réalisé avec soin. Le critère de choix à prendre en compte est de faciliter la résolution des équations (4.92) ou (4.90). A cet effet, la connaissance de la fonction  $f$  ou de sa structure permettent de faciliter ce choix.

L'idée proposée ici est inspirée des méthodes de résolution des équations différentielles. La fonction de substitution  $\alpha_{eq}$  est déterminée non pas sur  $D_X$  mais sur  $D_{f(X)}$ <sup>18</sup>. Par exemple, si la fonction  $f$  a la forme

$$f_i(x, u) = a_{(i,1)}(x)b_{(i,1)}(u) + \dots + a_{(i,w)}(x)b_{(i,w)}(u) \quad (4.94)$$

où les fonctions  $f_i(x, u)$  sont les composantes lignes  $f(x, u)$ , alors la fonction de substitution  $\alpha_{eq}$  est donnée par

$$\alpha_{eq} = a_{(1,1)}(x) \otimes \dots \otimes a_{(i,v)}(x) \quad (4.95)$$

où les différentes fonctions  $a_{(1,1)}(x), \dots, a_{(i,v)}(x)$  sont indépendantes et la relation  $\alpha \leq a_{(i,j)}(x)$  est satisfaite pour tous les  $i = 1..n$  et  $j = 1..v$ .

**Remarque 4.3.5** Si aucune fonction équivalente à  $\alpha$  ne peut être trouvée (ou n'apporte pas la simplification de calculs escomptée) et dans le cadre de la décomposition avec critère de découplage,  $\alpha_{eq}$  peut être choisie telle que

$$\alpha_{eq} \geq \alpha$$

<sup>17</sup> dans le sens de la relation d'ordre  $\leq$  sur  $D_X$

<sup>18</sup>  $D_{f(X)}$  est ensemble des fonctions sur l'ensemble (resp. l'espace vectoriel) image de (resp. espace)  $X$  par la fonction  $f$

puisque cela implique que

$$m(\alpha_{eq}) \geq m(\alpha)$$

La fonction  $\alpha_{eq}$  ainsi choisie doit, si possible, vérifier la relation

$$\alpha_{eq} \square h \neq \text{cste}$$

sous peine d'échouer à la vérification de la mesurabilité du sous-modèle obtenu.

Cette substitution ne perturbe pas le déroulement de l'algorithme de décomposition 5 et permet en particulier de mener à terme la procédure itérative du théorème 4.2.

**Illustration** Soit la fonction  $f$  suivante

$$f(x, u) = [x_2x_3 + u \mid x_2 + x_2x_3 \mid x_1]^T, x = [x_1 \mid x_2 \mid x_3]^T \quad (4.96)$$

Posons  $\beta(x) = I_{3 \times 3}x = [x_1 \mid x_2 \mid x_3]^T$ . Pour calculer  $M(\beta(x))$ , les relations (4.83) et (4.88) sont utilisées :

$$M(\beta)(x) = [x_2x_3 \mid x_2 \mid x_1]^T \quad (4.97)$$

Posons  $\alpha(x) = I_{3 \times 3}x$ . La détermination de  $m(\alpha(x))$  en utilisant la relation (4.92) ne permet pas d'obtenir une solution. Prenons la fonction

$$\alpha_{eq}(x) = [x_2x_3 \mid x_1 \mid x_2]^T$$

La fonction  $\alpha_{eq}$  vérifie la relation  $\alpha \equiv \alpha_{eq}$  et  $m(\alpha_{eq})$  est donnée par

$$m(\alpha_{eq})(x) = [x_3 \mid -x_1 + x_2 \mid x_1]^T \quad (4.98)$$

Ce résultat peut être subsitué à  $m(\alpha)$  puisque  $m(\alpha) \equiv m(\alpha_{eq})$ .

### Extension de l'algorithme de décomposition par l'utilisation des techniques d'élimination de variables

Si l'emploi de l'algorithme 5 ne permet pas de déterminer un sous-modèle découplé, il est possible que c'est la conséquence de la taille insuffisante de l'ensemble des fonctions candidates à la décomposition représenté par la fonction  $\phi^0$ . Dans cette section une transformation préalable du modèle est proposée, afin que l'ensemble de fonctions candidates soit le plus important possible. Cette transformation est obtenue en utilisant des techniques d'élimination de variables. La transformation présentée dans cette section permet de traiter le cas où les entrées à découpler sont injectées linéairement. Elle est obtenue en utilisant l'algorithme d'élimination de Gauss. Le traitement du cas où l'injection est non linéaire, par exemple une injection multiplicative d'entrées, reste une perspective en cours de développement.

L'avantage de cette approche est la facilité d'utilisation : des algorithmes efficaces et rapides existent, et peuvent être implémentés sur la plupart des logiciels de calcul symbolique tels que MAPLE©, MUPAD©, ou MAXIMA.

### Détermination de la transformation préalable pour une injection linéaire d'entrées à découpler

Si la fonction  $\phi^0$  obtenue par le biais de la relation (4.25) ne satisfait pas l'un des critères de décomposition, et si la relation (4.99)

$$\text{rang} \left[ \frac{\partial f(x, u, \rho, \gamma)}{\partial \gamma} \right] < n \quad (4.99)$$

où  $n$  représente la dimension de  $f$ , est vérifiée, alors une transformation préalable du modèle à décomposer est susceptible d'améliorer (ou de faciliter) la décomposition. Cette transformation est obtenue en éliminant les variables à découpler  $\gamma$  du modèle à décomposer.

L'ordre d'élimination est donné par la séquence suivante

$$\gamma_1 > \dots > \gamma_{k1} > x_1 > \dots > x_n > u_1 > \dots > u_l > \rho_1 > \dots > \rho_{k2} \quad (4.100)$$

les variables les plus à gauche sont à éliminer en priorité. Puisque l'objectif de la décomposition est d'obtenir un sous-modèle découplé de  $\gamma$  et couplé à  $\rho$ , les entrées  $\gamma$  sont à éliminer en priorité et les entrées  $\rho$  ne doivent pas être éliminées. Si l'entrée à découpler intervient sous forme linéaire (injection linéaire), le modèle apparaît sous la forme suivante

$$\Sigma \begin{cases} \dot{x} = f(x, u, \rho) + E\gamma \\ y = h(x) \end{cases} \quad (4.101)$$

où  $E$  est une matrice linéaire. La relation (4.24) stipule que l'ensemble des fonctions de décomposition possibles  $\phi^0$  serait plus grand si l'influence de  $\gamma$  sur le modèle venait à diminuer. La transformation  $T$  permet de réduire cette influence. Puisque l'injection est linéaire, la transformation  $T$  est une matrice qui transforme la matrice d'injection de la perturbation  $E$ .

$$TE = \hat{E} = \begin{bmatrix} D \\ 0 \end{bmatrix} \quad (4.102)$$

où  $D$  est une matrice diagonale. Le modèle (4.101) est transformé en conséquence

$$f(x, u) + K\rho + E\gamma \xrightarrow{T} \tilde{\Sigma} \begin{cases} \tilde{f}_1(\tilde{x}, u, \rho) + D\gamma \\ \tilde{f}_2(\tilde{x}, u, \rho) \end{cases} \quad (4.103)$$

Il n'est pas nécessaire de déterminer le résultat de la transformation des équations d'état ( $\tilde{f}_1, \tilde{f}_2$ ). En réalité, seules les équations qui éliminent le signal  $\gamma$  sont utiles. Puisque  $\text{rang}[E] = \text{rang}[D] = r$ , la transformation  $T$  génère  $(n - r)$  états découplés formellement par rapport à l'influence des entrées  $\gamma$ . La matrice de transformation  $T$  est déterminée à partir de  $E$  par élimination gaussienne.

Le calcul de la matrice  $T$  n'est pas toujours évident. En appliquant l'algorithme d'élimination gaussienne sur  $E$ , une matrice  $\hat{E}$  est obtenue. Si  $E$  n'est pas carrée mais rectangulaire, alors  $T$  peut pas être obtenue par la relation

$$T = \hat{E}E^{-1} \quad (4.104)$$

La solution vient de l'utilisation de la pseudo inverse de E

$$E^+ = (E^T E)^{-1} E^T \quad (4.105)$$

avec la contrainte que E soit de rang plein colonne. La matrice de transformation T est alors donnée par

$$T = \hat{E}E^+ \quad (4.106)$$

L'objectif visé par la détermination de T est l'obtention de nouvelles fonctions candidates afin d'augmenter la fonction d'initialisation  $\phi^0$  ce qui améliore les performances de l'algorithme de décomposition.

$$T = \left[ \begin{array}{cccc} t_{1,1} & \dots & t_{1,n} \\ t_{r,1} & \dots & t_{r,n} \\ t_{r+1,1} & \dots & t_{r+1,n} \\ t_{n,1} & \dots & t_{n,n} \end{array} \right] \phi^0 \quad (4.107)$$

où  $t_{j,k}$  sont les composantes de T. Les éléments de  $\phi^0$  sont les combinaisons des états du modèle qui éliminent le signal de perturbation.

### Exemple de décomposition pour une injection linéaire de défaillances

Considérons le système suivant :

$$\begin{cases} \dot{x}_1 = x_2 x_3 + u_1 + \gamma \\ \dot{x}_2 = x_4^2 + \rho + \gamma \\ \dot{x}_3 = x_4 x_5 + \gamma \\ \dot{x}_4 = x_1 x_4 x_5 + x_2 x_3 u_2 \\ \dot{x}_5 = x_4 + x_3 u_3 \end{cases} \quad \begin{cases} y_1 = x_1 \\ y_2 = x_4 \\ y_3 = x_5 \end{cases} \quad (4.108)$$

où  $\gamma$  représente la perturbation et  $\rho$  représente la défaillance à détecter. La fonction d'initialisation  $\phi^0$  est déterminée en utilisant la relation 5

$$\phi^0(x) = [x_4 \mid x_5]^T \quad (4.109)$$

La fonction  $\phi^0(x)$  sélectionne les états qui ne sont pas influencés par la perturbation  $\gamma$ . Par analogie, les états qui ne sont pas influencés par la défaillance forment la fonction  $\beta^0(x)$  (4.32) :

$$\beta^0 = [x_1 \ x_3 \ x_4 \ x_5]^T \quad (4.110)$$

L'étape suivante consiste à déterminer  $\phi^*$  qui est la partie de  $\phi^0$  qui satisfait

$$m(\phi_* \otimes h) \leq \phi_*$$

Le théorème 4.2 est utilisé pour déterminer  $\phi_*$ .

La première itération donne

$$\begin{aligned}\phi^1 &= \phi^0 \square m(\phi^0 \otimes h) \\ &= [x_4 \ x_5]^T \square m([x_1|x_4 \ x_5]^T) \\ &= [x_4]\end{aligned}\tag{4.111}$$

Puisque  $\phi^1 \not\equiv \phi^0$ , alors une seconde itération est nécessaire

$$\begin{aligned}\phi^2 &= \phi^1 \square m(\phi^1 \otimes h) \\ &= [x_4] \square m([x_1 \ x_4 \ x_5]^T) \\ &= [x_4]\end{aligned}\tag{4.112}$$

Puisque  $\phi^1 \equiv \phi^2$ , alors  $\phi^* = [x_4]$ .

Le critère de couplage n'est pas satisfait, puisque  $\beta^0 \leq \phi^*$ . Cela implique que le sous-modèle découplé généré à partir de  $\phi_*$  ne sera pas couplé à la défaillance.

Une solution peut être obtenue en utilisant la transformation du modèle proposée précédemment. La méthode de Gauss est utilisée pour éliminer la perturbation  $\gamma$  d'un maximum d'équations.

Le système devient

$$\begin{cases} \dot{\tilde{x}}_1 = \dot{x}_1 = (\tilde{x}_1 - \tilde{x}_2)(\tilde{x}_1 - \tilde{x}_2 - \tilde{x}_3) + u_1 + \gamma \\ \dot{\tilde{x}}_2 = \dot{x}_1 - \dot{x}_2 = (\tilde{x}_1 - \tilde{x}_2)(\tilde{x}_1 - \tilde{x}_2 - \tilde{x}_3) - \tilde{x}_4^2 + u_1 - \rho \\ \dot{\tilde{x}}_3 = \dot{x}_2 - \dot{x}_3 = \tilde{x}_4(\tilde{x}_4 - \tilde{x}_5) + \rho \\ \dot{\tilde{x}}_4 = \dot{x}_4 = \tilde{x}_1\tilde{x}_4\tilde{x}_5 + (\tilde{x}_1 - \tilde{x}_2)(\tilde{x}_1 - \tilde{x}_2 - \tilde{x}_3)u_2 \\ \dot{\tilde{x}}_5 = \dot{x}_5 = \tilde{x}_4 + (\tilde{x}_1 - \tilde{x}_2 - \tilde{x}_3)u_3 \end{cases}\tag{4.113}$$

$$\begin{cases} \tilde{y}_1 = \tilde{x}_1 \\ \tilde{y}_2 = \tilde{x}_4 \\ \tilde{y}_3 = \tilde{x}_5 \end{cases}$$

La fonction  $\phi^0$  est donnée par

$$\phi^0 = [\tilde{x}_2 \ \tilde{x}_3 \ \tilde{x}_4 \ \tilde{x}_5]^T\tag{4.114}$$

Par analogie, les états qui ne sont pas influencés par la défaillance forment  $\beta^0(x)$

$$\beta^0 = [\tilde{x}_1 \ \tilde{x}_4 \ \tilde{x}_5]^T\tag{4.115}$$

Nous constatons que  $\beta^0 \not\leq \phi^0$ .

A présent la fonction  $\phi_*$  est recherchée. La première itération du théorème 4.2 donne



$$\begin{aligned}
 \phi^1 &= \phi^0 \square m(\phi^0 \otimes h) \\
 &= [\tilde{x}_2 \ \tilde{x}_3 \ \tilde{x}_4 \ \tilde{x}_5]^T \square m([\tilde{x}_1 \ \tilde{x}_2 \ \tilde{x}_3 \ \tilde{x}_4 \ \tilde{x}_5]^T) \\
 &= [\tilde{x}_2 \ \tilde{x}_3 \ \tilde{x}_4]^T
 \end{aligned} \tag{4.116}$$

Puisque  $\phi^1 \neq \phi^0$ , une seconde itération est nécessaire

$$\begin{aligned}
 \phi^2 &= \phi^1 \square m(\phi^1 \otimes h) \\
 &= [\tilde{x}_2 \ \tilde{x}_3 \ \tilde{x}_4]^T
 \end{aligned} \tag{4.117}$$

Puisque  $\phi^1 \equiv \phi^2$ , alors  $\phi^* = [\tilde{x}_2 \ \tilde{x}_3 \ \tilde{x}_4]^T$ . Par ailleurs,  $\beta_0 \not\leq \phi^*$ , ce qui implique que le sous-modèle généré pourra éventuellement être sensible à la défaillance.

L'étape suivante est la détermination du modèle  $\Sigma_*$  en sélectionnant les fonctions qui satisfont la contrainte  $M(\phi) \geq (\phi \otimes h)$  en utilisant l'algorithme 4

1. La première itération initialise l'algorithme avec une fonction qui satisfait la contrainte de mesurabilité :

$$\phi_1 = \phi^* \square h = [\tilde{x}_4]$$

2. La condition  $M(\phi) \geq (\phi \otimes h)$  est vérifiée une première fois pour  $\phi_1$  :

$$h \otimes \phi_1 = [\tilde{x}_1 \ | \ \tilde{x}_4 \ | \ \tilde{x}_5]^T \not\leq M(\phi_1)$$

La condition n'est pas satisfaite puisque

$$M(\phi_1) = [\tilde{x}_1 \tilde{x}_4 \tilde{x}_5 \ | \ \tilde{x}_2 \tilde{x}_3 \ | \ \tilde{x}_1 \tilde{x}_3 \ | \ \tilde{x}_1 \tilde{x}_2 \ | \ \tilde{x}_1^2 \ | \ \tilde{x}_2^2]^T$$

3. Une seconde itération est initiée, en choisissant la fonction

$$\phi_2 = [\tilde{x}_2 \ | \ \tilde{x}_3]^T$$

qui satisfait

$$\phi_2 \leq \phi^* \text{ et } h \otimes \phi_1 \otimes \phi_2 \leq M(\phi_1)$$

4. La condition  $M(\phi) \geq (\phi \otimes h)$  est vérifiée une seconde fois

$$h \otimes \phi_1 \otimes \phi_2 = [\tilde{x}_1 \ | \ \tilde{x}_2 \ | \ \tilde{x}_3 \ | \ \tilde{x}_4 \ | \ \tilde{x}_5]^T \leq M(\phi_2)$$

Il apparaît que la condition est satisfaite pour cette fonction  $\phi$  puisque

$$M(\phi_2) = [\tilde{x}_2 \tilde{x}_3 \ | \ \tilde{x}_1 \tilde{x}_3 \ | \ \tilde{x}_1 \tilde{x}_2 \ | \ \tilde{x}_1^2 \ | \ \tilde{x}_2^2 \ | \ \tilde{x}_4^2 \ | \ \tilde{x}_4 \tilde{x}_5]^T$$

5. La fonction de décomposition  $\phi$  est donnée par

$$\phi = \phi_1 \otimes \phi_2 = [\tilde{x}_2 \ | \ \tilde{x}_3 \ | \ \tilde{x}_4]^T$$

Puisque  $\phi \square \beta^0 = \tilde{x}_4 \neq \text{cste} \Leftrightarrow \beta^0 \not\leq \phi$ , alors le système  $\Sigma_*$  est sensible à la défaillance. La description de  $\Sigma_*$  est donnée par

$$\begin{cases}
 \dot{x}_{*1} = \dot{\tilde{x}}_2 = (y_1 - x_{*1})(y_1 - x_{*1} - x_{*2}) - \tilde{y}_2^2 + u_1 \\
 \dot{x}_{*2} = \dot{\tilde{x}}_3 = \tilde{y}_2(\tilde{y}_2 - \tilde{y}_3) \\
 \dot{x}_{*3} = \dot{\tilde{x}}_4 = \tilde{y}_1 \tilde{y}_2 \tilde{y}_3 + (y_1 - x_{*1})(y_1 - x_{*1} - x_{*2})u_2
 \end{cases} \tag{4.118}$$

avec  $y_* = x_{*3}$  et  $c(\tilde{y}) = \tilde{y}_2$

### 4.3.6 Conclusion

Le principe de la décomposition de modèles temporels avec des critères de couplage et de découplage a été présentée et illustrée par des exemples. La méthode de découplage présentée dans [Shu91, ZS93] a été tout d'abord développée et discutée. Des solutions ont ensuite été proposées pour améliorer l'algorithme de décomposition :

- une approche alternative de calcul pour certains opérateurs de l'algèbre des fonctions est proposée, ainsi que le détail du calcul des autres opérateurs,
- une optimisation de l'étape d'initialisation est suggérée. La recherche du plus grand ensemble de fonctions candidates est réalisée en minimisant au préalable l'influence formelle de l'entrée à découpler grâce à l'utilisation de techniques d'élimination de variables. Le cas de l'injection linéaire des entrées est traité en utilisant l'élimination Gaussienne. Le cas non linéaire est une perspective en développement.

Le résultat de la décomposition se présente sous la forme de sous-modèles découplés qui sont par la suite utilisés pour la génération de résidus structurés (voir chapitre 2).

## 4.4 Décomposition des modèles comportementaux événementiels

### 4.4.1 Introduction et objectifs

Cette section est consacrée à la décomposition des modèles événementiels. La méthodologie décrite pour le cas général est reformulée en utilisant les outils de l'algèbre des paires. Les critères et les contraintes de la décomposition sont traduits pour correspondre aux modèles représentés par des machines séquentielles. La réécriture des contraintes de décomposition permet de montrer les analogies entre des notions liées aux modèles temporels et celles liées aux modèles événementiels, notamment en ce qui concerne le critère d'existence du sous-modèle. L'algorithme général de décomposition est ensuite formulé et un exemple didactique est présenté pour l'illustration.

Considérons la machine séquentielle  $\Sigma$  décrite par le quintuplet  $(S, I, O, \delta, \lambda)$ , où  $S$ ,  $I$  et  $O$  sont respectivement les ensembles finis d'états, d'entrées et de sorties et où  $\delta$  et  $\lambda$  sont respectivement l'application de transition d'état et l'application de sortie décrites par

$$\delta : S \times I \rightarrow S \text{ et } \lambda : S \rightarrow O \quad (4.119)$$

L'ensemble d'entrées est composé de deux sous-ensembles :

- sous-ensemble d'événements connus noté  $I_c$
- sous-ensemble d'événements inconnus  $I_{nc} = I_\gamma \cup I_\rho$ , où  $I_\rho$  est l'ensemble d'événements représentant des défaillances et  $I_\gamma$  est l'ensemble d'événements représentant des perturbations.

Pour un sous-ensemble d'entrées  $I_\gamma$  donné, la décomposition de la machine séquentielle  $\Sigma$  donne deux machines séquentielles partielles  $\Sigma'$  et  $\Sigma_*$  :

- $\Sigma_*$  est la machine séquentielle partielle découplée du sous-ensemble d'entrées  $I_\gamma$  et couplée au sous-ensemble  $I_\rho$ . La dimension de la machine découplée doit être maximale.
- $\Sigma'$  est la machine séquentielle partielle couplée au sous-ensemble d'entrées  $I_\gamma$ . C'est le modèle complémentaire à  $\Sigma_*$ .

Pour obtenir une machine séquentielle partielle découplée à partir de  $\Sigma$ , un partitionnement de l'ensemble d'états est fait, suivant une partition notée  $\pi_*$ . La manipulation de partitions d'ensembles finis est plus simple d'un point de vue calculatoire que la manipulation de partitions d'ensembles infinis, et c'est pour cette raison que nous n'exprimerons plus le problème de décomposition à l'aide de fonctions, mais directement en termes de partitions.

Considérons la machine séquentielle partielle  $\Sigma_*(S_*, I_*, O_*, \delta_*, \lambda_*)$  obtenue en partitionnant l'ensemble d'états  $S$  de la machine séquentielle  $\Sigma$ , alors les ensembles du quintuplet qui décrivent  $\Sigma_*$  sont donnés par :

- $S_* = \pi_*$  : une partition de  $S$ ,
- $I_* = \pi_I$  : une partition de  $I$ ,
- $O_* = \pi_O$  : une partition de  $O$ ,

où  $\pi_I$  et  $\pi_O$  sont respectivement les partitions de  $I$  et de  $O$  qui sont associées à  $\pi_*$ . Les applications  $\delta_*$  et  $\lambda_*$  sont respectivement les restrictions de  $\delta$  et de  $\gamma$  :

$$\delta_* : \pi_* \times \pi_I \rightarrow \pi_* \text{ et } \lambda_* : \pi_* \rightarrow \pi_O \quad (4.120)$$

Le problème de décomposition consiste à déterminer la partition  $\pi_*$  de l'ensemble d'états  $S$  qui satisfait les critères de décomposition. La machine séquentielle partielle complémentaire  $\Sigma'$  est déterminée à partir de la partition  $\pi'$  qui satisfait

$$\pi' \cdot \pi_* = S \quad (4.121)$$

Les deux machines séquentielles partielles permettent de restituer l'évolution du modèle complet  $\Sigma$ .

#### 4.4.2 Méthodologie de décomposition d'un modèle événementiel

L'ensemble des critères et des contraintes de décomposition est repris dans cette section en utilisant les outils de l'algèbre des paires. La possibilité de manipuler directement les partitions permet de s'affranchir de certaines étapes qui sont nécessaires pour la décomposition des modèles temporels.

##### Critères de décomposition

Le critère de décomposition est double :

- le couplage par rapport à un sous-ensemble d'événements  $I_\rho$ ,
- le découplage par rapport à un sous-ensemble d'événements  $I_\gamma$

Le problème est formulé de sorte à obtenir la plus grande machine séquentielle partielle découplée de  $I_0$  en gardant le couplage à  $I_\gamma$ . Un partitionnement de l'ensemble d'entrées est fait en fonction du critère de décomposition. Cette partition est la plus fine partition possible pour satisfaire la condition de dimension.

### Critère de découplage

Le critère de découplage défini précédemment pour le cas général, est repris en termes d'algèbre des paires.

Soit un événement  $a \in I$ . La partition  $\pi_{I_*}$  découplée du sous-ensemble  $I_* = \{a\}$  est définie suivant la procédure suivante :

Un événement virtuel  $i_0$  est défini tel que :

$$\forall s \in S : \delta(s, i_0) = s \quad (4.122)$$

Cet élément représente l'élément neutre de l'application  $\delta$ .  $i_0$  est ajouté à l'ensemble d'entrées  $I$  de la machine séquentielle  $\Sigma$ . La partition triviale  $\mathbb{O}$  de  $I$ , qui est la partition la plus fine possible de  $I$ , est par définition découplée par rapport à  $i_0$ .

La plus fine partition de l'ensemble d'entrées découplée par rapport à  $I'$  est la partition qui regroupe dans un seul bloc les événements  $i_0$  et  $a$ . Cette partition est notée  $\pi_{I_*}^0$  (4.123).

$$\pi_{I_*}^0 = \{\overline{i_0, a}, \overline{i_1}, \dots, \overline{i_l}\} \quad (4.123)$$

où les différents  $i_j$  représentent les événements d'entrée. La partition  $\pi_{I_*}^0$  entraîne une partition de l'ensemble d'états notée  $\pi_*^0$ . La partition  $\pi_*^0$  est définie comme la plus fine partition de l'ensemble d'états découplée de  $I'$ . Cette partition est obtenue en utilisant l'opérateur  $m$  de l'algèbre des paires :

$$\pi_*^0 = m_{I-S}(\pi_{I_*}^0) \quad (4.124)$$

Considérons  $\Sigma_*$  une machine séquentielle partielle extraite de  $\Sigma(S, I, O, \delta, \lambda)$ .  $\Sigma_*$  est découplée de  $I'$  si son ensemble d'états  $S_*$  peut être exprimé en utilisant seulement des éléments de la partition  $\pi_*^0$ . Ce qui s'écrit en termes d'algèbre des paires :

$$\pi_*^0 \leq \pi_* \quad (4.125)$$

où  $\pi_*$  est la partition de décomposition de  $S$ . La relation (4.125) exprime le critère que doit satisfaire la partition  $\pi_*$  pour obtenir un sous-modèle découplé de  $I'$ .

### Critère de couplage

Le critère de couplage est défini de la même manière que pour le cas des modèles temporels (4.32). Soit un événement  $b \in I$  tel que  $b \notin I'$ . Un critère est donné pour vérifier qu'une machine séquentielle partielle découplée d'un sous-ensemble d'entrées  $I'$  reste couplée à l'événement  $b$ .

La partition  $\pi_{I_*}$  découplée de l'événement  $b$  est définie par la relation

$$\bar{\pi}_{I_*} = \{\overline{i_0}, \overline{b}, \overline{i_1}, \dots, \overline{i_l}\}$$

et la partition d'état correspondante est donnée par

$$\bar{\pi}_*^0 = m_{I-S}(\bar{\pi}_{I_*}^0) \quad (4.126)$$

La machine séquentielle partielle  $\Sigma_*$ , obtenue par décomposition de  $\Sigma$  par rapport à une partition  $\pi_*$ , est couplée à  $b$  si la relation suivante est satisfaite

$$\bar{\pi}_*^0 \not\leq \pi_* \quad (4.127)$$

où  $\bar{\pi}_*^0$  est déterminée par la relation (4.126)

### Les contraintes de décomposition

Les contraintes de décomposition pour le cas général sont transposées dans le cadre des machines séquentielles. Rappelons qu'un modèle  $\Sigma_*$  décrit une partie d'un autre modèle  $\Sigma$  si les conditions suivantes sont respectées :

- L'ensemble d'états de  $\Sigma_*$  est un sous-ensemble (ou une partition) des états de  $\Sigma$  :  $S_* \subseteq S$  ou  $S_* = \pi(S)$  où  $\pi$  est une partition de  $S$ . La restriction de l'application d'état  $\delta$  à l'ensemble  $S_*$  doit exister.
- Les sorties de  $\Sigma_*$  et de  $\Sigma$  sont cohérentes entre elles : Pour les états similaires des deux modèles, les sorties doivent être similaires également. L'ensemble de sorties  $O_*$  doit être un sous-ensemble ou une partition de  $O$  et la fonction de sortie  $\lambda_*$  doit être une restriction de  $\lambda$  sur un sous-ensemble de  $Y_*$ .

Les contraintes d'invariance et de mesurabilité sont satisfaites si  $\Sigma_*$  est une image *homomorphe* de  $\Sigma$  (voir annexe A et [HS66]). Dans la suite de cette section, la propriété de substitution décrite dans le chapitre précédent à la section 3.3.2 est utilisée pour démontrer l'homomorphisme de machines séquentielles.

#### a. Contrainte d'existence et propriété de substitution

Considérons la machine d'état  $\Sigma^E : (S, I, \delta)$  extraite de la machine séquentielle  $\Sigma$  définie par  $(S, I, O, \delta, \lambda)$ . Soit une partition  $\pi$  de  $S$  qui possède la propriété de substitution par rapport à  $\delta$ . La propriété de substitution implique que chaque bloc de  $\pi$  a pour image selon  $\delta$  un autre bloc de  $\pi$ .

$$\forall \alpha \in \pi, i \in I : \delta(\alpha, i) \in \pi \quad (4.128)$$

L'équation (4.128) implique qu'il existe une restriction de  $\delta$  sur  $\pi$  notée  $\delta_*$  telle que :

$$\begin{aligned} \delta_* : \pi \times I &\rightarrow \pi \\ \alpha &\mapsto \delta_*(\alpha) \end{aligned} \quad (4.129)$$

La relation suivante permet de déterminer la partition de l'ensemble d'entrées  $I$  qui est associée à  $\pi$  :

$$\pi_I = M_{I-S}(\pi) \quad (4.130)$$

tous les éléments de  $\pi_I$  qui sont dans le même bloc que l'élément neutre  $i_0$  de  $I$  sont également neutres par rapport à l'application  $\delta_*$  :

$$\forall \alpha \in B_{\pi_I}(i_0), \forall s_* \in \pi : \delta_*(s_*, \alpha) = s_*$$

La machine d'état  $\Sigma_*^E$  définie par le triplet  $(S_*, I_*, \delta_*)$  où  $S_* = \pi$  et  $I_* = \pi_I$  est une image *homomorphe* de la machine séquentielle  $\Sigma$  suivant la définition donnée dans l'annexe A. Par ailleurs,  $\Sigma_*$  est découplée de toutes les entrées  $\alpha \in B_{\pi_I}(i_0)$ .

La condition d'existence d'une machine séquentielle partielle peut être formulée par rapport à sa partition de définition  $\pi$ . En généralisant cette notion pour les machines de Moore, une partition  $\pi$  qui possède la propriété de substitution définit une machine séquentielle partielle de  $\Sigma(S, I, O, \delta, \lambda)$  réalisable. Cette machine séquentielle partielle est décrite par le quintuplet

$$(\pi, M_{I-S}(\pi), O_*, \delta_*, \lambda_*)$$

où  $\delta_*$  est une restriction de  $\delta$  sur  $\pi$  et  $O_*$  et  $\lambda_*$  sont quelconques. La contrainte d'existence d'une machine séquentielle partielle déterminée par  $\pi$  est alors formulée en termes d'algèbre des paires par

$$(\pi, \pi) \in \Delta_\delta \tag{4.131}$$

La relation (4.131) est satisfaite si :

$$\pi \leq M(\pi) \tag{4.132}$$

$$\pi \geq m(\pi) \tag{4.133}$$

Si les relations (4.132) et (4.133) sont satisfaites, alors la partition  $\pi$  permet de déterminer une machine séquentielle partielle de  $\Sigma$ .

**Illustration** Considérons la machine séquentielle  $\Sigma$  décrite par la table (4.1)

	a	b	o
1	2	3	N
2	1	3	O
3	3	2	O

TAB. 4.1 – Table de transition d'états de  $\Sigma$

et la figure 4.3.

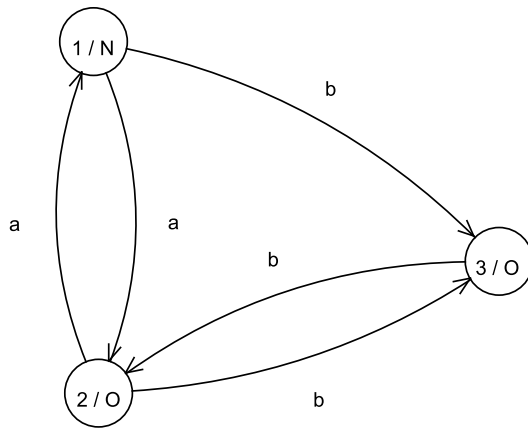


FIG. 4.3 – Graphe représentant la machine séquentielle  $\Sigma$

La partition  $\pi = \{\overline{1}, \overline{2}, \overline{3}\}$  satisfait la contrainte d'existence puisque

$$m(\{\overline{1}, \overline{2}, \overline{3}\}) = \{\overline{1}, \overline{2}, \overline{3}\}$$

et que la propriété «  $\pi \geq m(\pi)$  » (4.133) est satisfaite.  $\pi$  possède alors la propriété de substitution et  $(\pi, \pi)$  forme une paire algébrique. La machine  $\Sigma_\pi$  obtenue par le biais de  $\pi$  est représentée par la figure 4.4. Le symbole X désigne ici une sortie « O ou N », ce qui implique que cette machine séquentielle partielle ne satisfait pas la contrainte de mesurabilité traitée par la suite dans ce chapitre.

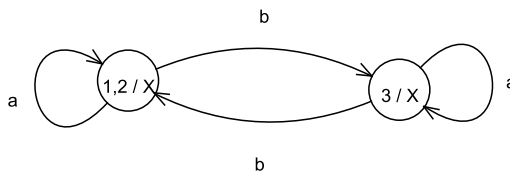
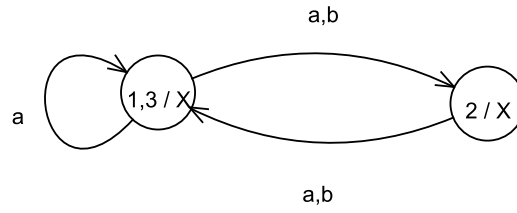


FIG. 4.4 – Graphe représentant la machine séquentielle partielle  $\Sigma_\pi$

Notons au passage que la partition  $\pi' = \{\overline{1}, \overline{3}, \overline{2}\}$  ne satisfait pas la contrainte d'existence puisque

$$m(\{\overline{1}, \overline{3}, \overline{2}\}) = \{\overline{1}, \overline{2}, \overline{3}\}$$

et que  $m(\pi') \not\leq \pi'$ . Cela implique que la partition  $\pi'$  ne possède pas la propriété de substitution. La figure 4.5 illustre le résultat de la décomposition. Il apparaît clairement que le modèle événementiel  $\Sigma'_\pi$  n'est pas une machine séquentielle déterministe et de ce fait ne peut pas représenter une machine séquentielle partielle de  $\Sigma$ .

FIG. 4.5 – Graphe représentant  $\Sigma_{\pi'}$ 

### Contrainte d'existence étendue et injection de sorties pour une machine séquentielle

Dans la section précédente, une analogie a été montrée entre la contrainte d'existence formulée dans le cas général (et dans le cas temporel) et la propriété de substitution des partitions pour le cas événementiel. Cette analogie permet de généraliser la formulation de la contrainte d'existence en utilisant la notion de paire algébrique. Nous avons défini par ailleurs un critère d'existence étendu pour le cas général, en utilisant une injection de sortie. L'injection de sorties est une démarche relativement courante dans le cadre des modèles temporels [DI01], mais demeure inexploitée dans le cadre des machines séquentielles.

Nous présenterons dans cette section un critère d'existence étendu basé sur cette notion d'injection de sorties pour une machine séquentielle. Rappelons que le critère d'existence étendu établi pour les modèles temporels s'exprime algébriquement par

$$(\phi \otimes h, \phi) \text{ est une paire algébrique}$$

ce qui est vérifié si

$$(\phi \otimes h) \leq M(\phi) \text{ et } m(\phi \otimes h) \leq \phi$$

où  $m$ ,  $M$  et  $\otimes$  sont des opérateurs de l'algèbre des fonctions,  $\phi$  est la fonction de décomposition et  $h$  est la fonction de sortie du modèle temporel à décomposer. En utilisant une approche analogique, un critère d'existence étendu est proposé pour la décomposition de machines séquentielles.

Soit une partition  $\pi$  qui ne possède pas la propriété de substitution vis-à-vis d'une machine séquentielle  $\Sigma(S, I, O, \delta, \lambda)$ . Elle définit une partie de  $\Sigma$  qui ne peut être réalisée sous la forme d'une machine séquentielle partielle. La partition  $\pi$  n'est pas assez fine. En *affinant*  $\pi$  à l'aide d'une injection de sorties, il est possible de satisfaire la propriété de substitution, puisque par définition la partition minimale  $\mathbb{O}$  possède la propriété de substitution.

Considérons  $\pi_\lambda$ , la partition de l'ensemble d'états  $S$  induite par la fonction de sortie  $\lambda$ . Cette partition regroupe tous les états qui génèrent une sortie similaire et se calcule en utilisant l'opérateur  $M$  :

$$\pi_\lambda = M_{S-\mathbb{O}}(\mathbb{O}) \quad (4.134)$$

où  $\mathbb{O}$  est la partition minimale triviale de l'ensemble  $O$ . Considérons la partition

$$\pi \cdot \pi_\lambda$$



qui représente la partition de l'ensemble d'états en fonction de  $\pi$  et affiné par une injection de sorties. Si  $\pi \geq (\pi, \pi_\lambda)$  alors il est possible que la partition  $\pi, \pi_\lambda$  possède la propriété de substitution.

Par analogie au cas général, le critère d'existence étendu est formulé par la relation suivante

$$(\pi, \pi_\lambda, \pi) \text{ forme une paire algébrique} \quad (4.135)$$

Cette condition est satisfaite si les deux relations suivantes sont vraies <sup>19</sup>

$$M(\pi) \geq (\pi, \pi_\lambda) \quad (4.136)$$

$$\pi \geq m(\pi, \pi_\lambda) \quad (4.137)$$

Si le critère (4.135) est satisfait, alors une décomposition de  $\Sigma$  suivant  $\pi$  est possible à l'aide d'une *injection de sorties*. Nous obtenons alors la machine séquentielle partielle  $\Sigma_\pi$ . L'injection de sorties est réalisée en augmentant l'ensemble d'entrées de  $\Sigma_\pi$  avec l'ensemble de sortie de  $\Sigma$ . Le nouvel ensemble d'entrées  $I_\pi$  est donné par

$$I_\pi = M_{I-S}(\pi) \times O$$

où  $\times$  représente le produit cartésien des ensembles  $I$  et  $O$ .

Pour résumer, si une partition  $\pi$  satisfait le critère d'existence étendu vis-à-vis d'une machine séquentielle  $\Sigma$  alors elle permet de déterminer une machine séquentielle partielle  $\Sigma_*$  décrite par le quintuplet

$$(S_*, I_*, O_*, \delta_*, \lambda_*)$$

où  $S_* = \pi$ ,  $I_* = M_{I-S}(\pi) \times O$ ,

$$\begin{aligned} \delta_* : S_* \times I_* &\rightarrow S_* \\ (s, (i, o)) &\mapsto \delta(s \cap o, i) \end{aligned}$$

où  $O_*$  et  $\lambda_*$  restent à définir.

Rigoureusement parlant,  $\Sigma_*$  n'est pas une partie de  $\Sigma$ , mais une machine séquentielle qui *estime* l'évolution de la partie de  $\Sigma$  qui est déterminée par  $\pi$ .

**Remarque 4.4.1** Une conséquence de l'utilisation du critère étendu est une augmentation conséquente du nombre d'entrées ce qui peut poser un problème pour des systèmes avec ensemble de sorties de cardinalité importante. Ce problème est strictement calculatoire et la solution peut être optimisée en fonction du cas traité, en utilisant uniquement une partition particulière de l'ensemble  $O$  au lieu de la partition minimale  $\mathbb{O}$ .

<sup>19</sup>Suivant les propriétés de l'algèbre des paires, et pour des modèles de type machines séquentielles, si l'une de ces conditions est satisfaite, alors l'autre condition est satisfaite également.

**Illustration** Considérons la machine séquentielle  $\Sigma$  de l'exemple précédent, et la partition  $\pi' = \{\overline{1,3}, \overline{2}\}$ . En utilisant la contrainte d'existence étendue, nous allons déterminer la machine séquentielle partielle  $\Sigma_{\pi'}$ . La partition de  $S$  induite par  $\lambda$  est donnée par

$$\pi_{\lambda} = \{\overline{2,3}, \overline{1}\}$$

la contrainte d'existence est satisfaite puisque

$$\pi \cdot \pi_{\lambda} = \mathbb{O}$$

et

$$m(\mathbb{O}) = \mathbb{O} \leq \pi$$

La machine séquentielle partielle  $\Sigma_{\pi'}$  est montrée par la figure 4.6. L'ensemble d'entrées est donné par

$$I_{\pi'} = \{(a, \mathbb{O}), (a, N), (b, \mathbb{O}), (b, N)\}$$

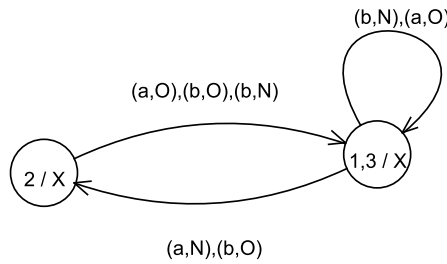


FIG. 4.6 – Graphe représentant la machine séquentielle partielle  $\Sigma_{\pi'}$

### Contrainte de cohérence des sorties : la propriété de mesurabilité

La contrainte de mesurabilité est démontrée dans le cas général par l'existence d'une restriction de la fonction de sortie sur les sous-ensembles d'états et de sorties obtenus par la décomposition. Nous avons vu dans la section précédente qui traitait de la contrainte d'existence, que l'existence d'une restriction peut être démontrée en utilisant la propriété de substitution.

Par analogie, si la partition de décomposition  $\pi$  possède la propriété de substitution par rapport à l'application  $\lambda$  alors la restriction de  $\lambda$  sur la partition  $\pi$  existe et le critère de mesurabilité est satisfait. Ce qui s'écrit mathématiquement par

$$(\pi, \pi) \in \Delta_{\lambda} \quad (4.138)$$

ce qui signifie que  $(\pi, \pi)$  est une paire algébrique par rapport à  $\lambda$ . Cette propriété est satisfaite si la relation

$$M_{S-\mathbb{O}}(\mathbb{O}) \geq \pi \quad (4.139)$$

est vérifiée. Si la relation (4.139) est satisfaite alors la machine séquentielle partielle obtenue par  $\pi$  est cohérente avec toutes les sorties de  $\Sigma$  ce qui réduit considérablement les possibilités de succès de la décomposition.

Si une restriction de  $\lambda$  sur au moins une partition non triviale de  $O$  existe, alors la contrainte de mesurabilité est satisfaite. Cette contrainte est suffisante pour vérifier que la machine séquentielle partielle est cohérente en sorties avec le modèle complet. En utilisant une formulation algébrique, cette contrainte s'écrit par

$$M_{S-O}(O) + \pi \neq \mathbb{I} \tag{4.140}$$

où  $\mathbb{I}$  et  $O$  sont respectivement les partitions triviales maximale et minimale de  $O$ . La partition de l'ensemble d'états induite par l'ensemble de sorties doit avoir une intersection nulle avec  $\pi$ .

**Illustration** Considérons la machine séquentielle  $\Sigma$  (figure 4.7) décrite par la table 4.2 :

	a	b	o
1	2	3	O
2	1	3	O
3	3	2	N

Tab. 4.2 – table de transition d'état de la machine séquentielle  $\Sigma$

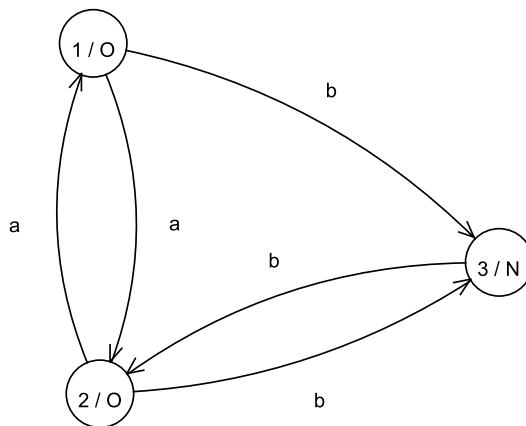


FIG. 4.7 – Graphe illustrant la machine séquentielle  $\Sigma$

Déterminons en premier lieu la partition  $\pi_\lambda$

$$\pi_\lambda = M_{S-O}(O) = \{\overline{1}, \overline{2}, \overline{3}\}$$

La partition  $\pi = \{\overline{1}, \overline{2}, \overline{3}\}$  satisfait la contrainte de mesurabilité puisque

$$\pi_\lambda + \pi = \pi \neq \mathbb{I}$$

Il s'avère que  $\pi_\lambda \geq \pi$ , ce qui implique que  $\pi$  possède la propriété de substitution par rapport à  $\lambda$ . Les sorties du sous-modèle obtenu sont donc cohérentes avec toutes les sorties de  $\Sigma$ .

La partition  $\pi' = \{\overline{1,3}, \overline{2}\}$  par contre ne satisfait pas la contrainte de mesurabilité puisque

$$\pi_\lambda + \pi' = \{\overline{1,2,3}\} = \mathbb{I}$$

Les sorties de la machine séquentielle partielle obtenue par la partition de décomposition  $\pi'$  ne pourront pas être comparées au modèle complet.

### Implémentation de l'algorithme de décomposition généralisé en utilisant les opérateurs de l'algèbre des paires

L'intégralité des critères et des contraintes de décomposition ont été traduits dans le formalisme de l'algèbre des paires tout en tenant compte des spécificités des machines séquentielles. Ces critères et ces contraintes nous permettent de réaliser l'algorithme 2, qui, rappelons le, comporte les phases suivantes :

1. une phase d'initialisation : l'ensemble des solutions candidates est déterminé à partir du critère de la décomposition,
2. une phase itérative de recherche de solutions qui répondent aux contraintes de la décomposition :
  - une solution qui satisfait la contrainte de mesurabilité est choisie,
  - une procédure itérative est initiée pour retrouver la partition qui satisfait la contrainte d'existence,
  - si aucune partition ne satisfait la contrainte d'existence, une autre solution mesurable est sélectionnée et la phase itérative recommencée.

A l'issue de l'itération, un sous-modèle découplé est obtenu. Ce sous-modèle satisfait le critère de découplage et les deux contraintes de décomposition. La dimension du sous-modèle est la plus grande possible, puisque l'algorithme effectue une élimination à partir d'un ensemble de solutions.

3. une phase de vérification de couplage : Le sous-modèle découplé obtenu est confronté au critère de couplage. Si ce critère n'est pas satisfait, les sorties du sous-modèle ne seront pas sensibles à l'événement à détecter.

L'algorithme de décomposition proposé dans cette section prend en considérations les spécificités des machines séquentielles. Comme il a été mentionné précédemment, les modèles comportementaux décrits sous la forme

$$\Sigma : x^+ = f(x, w)$$

garantissent la monotonie des opérateurs algébriques  $m$  et  $M$ <sup>20</sup>.

La monotonie implique que si  $m(\pi) \leq \pi$  alors la relation  $\pi \leq M(\pi)$  est vérifiée, puisque  $M(m(\pi)) \leq M(\pi)$  et que  $\pi \leq M(m(\pi))$ . En outre, si  $m(\pi.\pi_\lambda) \leq \pi$ , alors en suivant la même démarche  $(\pi.\pi_\lambda) \leq M(\pi)$  est satisfaite. Cette propriété permet de simplifier l'algorithme de décomposition en recherchant uniquement les solutions qui satisfont

$$m(\pi) \leq \pi \text{ ou } m(\pi.\pi_\lambda) \leq \pi$$

<sup>20</sup>voir les propriétés des opérateurs de l'algèbre des paires au chapitre 3

au lieu de la double contrainte qui s'impose dans le cas général. L'algorithme comporte alors une seule itération qui détermine la partition avec propriété de substitution suivant le théorème 4.2. En outre puisque la seconde itération est inutile, la phase de vérification comporte un test supplémentaire, celui de la mesurabilité.

La structure de l'algorithme simple est donnée par le pseudo code 7.

**Entrées :**  $\Sigma(S, I, O, \delta, \lambda)$

**Données :**  $i = 0, j = 0$  (compteurs)

**Sorties :**  $\pi$  (partition de décomposition)

**début**

$\pi_*^0 = m_{I-S}(\pi_\gamma)$  où  $\pi_\gamma$  est la partition qui découple le sous-ensemble  $I_\gamma$  ;

$\bar{\pi}_*^0 = m_{I-S}(\pi_\rho)$  où  $\pi_\rho$  est la partition qui découple le sous-ensemble  $I_\rho$  ;

$\pi_\lambda = M_{S-O}(\mathbb{O})$  ;

$\xi^0 = \pi_*^0$  ;

**tant que**  $\xi^i \neq \xi^{i-1}$  **faire**

  |  $\xi^{i+1} = m(\xi^i) + \xi^i$  ;

**fin**

$\pi = \xi^i$  ;

**si**  $\pi = \mathbb{I}$  **alors**

  | **retourner** *Décomposition impossible*

**fin**

**si**  $\pi + \pi_\lambda = \mathbb{I}$  **alors**

  | **retourner** *Critère de mesurabilité non rempli par  $\pi$*

**sinon**

  | **retourner** *Critère de mesurabilité rempli par  $\pi$*

**fin**

**si**  $\pi \geq \bar{\pi}_*^0$  **alors**

  | **retourner** *Critère de couplage non rempli par  $\pi$*

**sinon**

  | **retourner** *Critère de couplage rempli par  $\pi$*

**fin**

**fin**

**Algorithme 7 :** Algorithme de décomposition simple en termes d'algèbre des paires

La structure de l'algorithme étendu est donnée par le pseudo code 8.

Le résultat de l'algorithme de décomposition est une partition  $\pi$  qui permet d'obtenir une machine séquentielle partielle découplée d'un sous-ensemble d'entrées  $I_\gamma$  et couplée à un sous-ensemble d'entrées  $I_\rho$ .

### Illustration de la décomposition d'un modèle événementiel

Considérons le modèle événementiel  $\Sigma$  décrit par la figure 4.8 et la table 4.3

**Entrées :**  $\Sigma(S, I, O, \delta, \lambda)$

**Données :**  $i = 0, j = 0$  (compteurs)

**Sorties :**  $\pi$  (partition de décomposition)

**début**

$\pi_*^0 = m_{I-S}(\pi_\gamma)$  où  $\pi_\gamma$  est la partition qui découple le sous-ensemble  $I_\gamma$  ;

$\bar{\pi}_*^0 = m_{I-S}(\pi_\rho)$  où  $\pi_\rho$  est la partition qui découple le sous-ensemble  $I_\rho$  ;

$\pi_\lambda = M_{S-O}(\mathbb{O})$  ;

$\xi^0 = \pi_*^0$  ;

**tant que**  $\xi^i \not\cong \xi^{i-1}$  **faire**

  |  $\xi^{i+1} = m(\xi^i \cdot \pi_\lambda) + \xi^i$  ;

**fin**

$\pi = \xi^i$  ;

**si**  $\pi = \mathbb{I}$  **alors**

  | **retourner** *Décomposition impossible*

**fin**

**si**  $\pi + \pi_\lambda = \mathbb{I}$  **alors**

  | **retourner** *Critère de mesurabilité non rempli par  $\pi$*

**sinon**

  | **retourner** *Critère de mesurabilité rempli par  $\pi$*

**fin**

**si**  $\pi \geq \bar{\pi}_*^0$  **alors**

  | **retourner** *Critère de couplage non rempli par  $\pi$*

**sinon**

  | **retourner** *Critère de couplage rempli par  $\pi$*

**fin**

**fin**

**Algorithme 8 :** Algorithme de décomposition étendu en termes d'algèbre des paires

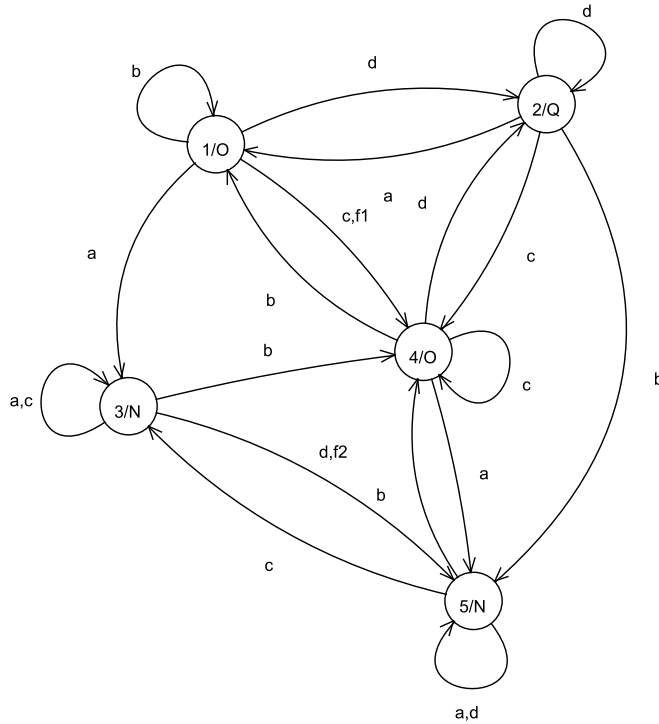


FIG. 4.8 – Représentation graphique de modèle  $\Sigma$

	a	b	c	d	$f_1$	$f_2$	o
1	3	-	4	2	4	-	O
2	1	5	4	-	-	-	Q
3	-	4	-	5	-	5	N
4	5	1	-	2	-	-	O
5		4	3	-	-	-	N

TAB. 4.3 – Table de transitions de la machine  $\Sigma$

$\Sigma$  est un modèle à 5 états, 4 événements connus et deux événements inconnus  $f_1$  et  $f_2$ . Nous allons illustrer la méthode de décomposition à un critère de découplage par rapport à  $I_\gamma = \{f_2\}$  et nous vérifierons le couplage par rapport à  $I_\rho = \{f_1\}$ .

La première étape consiste à calculer l'ensemble des solutions candidates à la décomposition qui satisfont le critère de découplage représenté par la partition  $\pi_*^0$ .

La partition de l'ensemble d'entrées  $I$  qui découple  $I_\gamma$  est donnée par

$$\pi_\gamma = \{\overline{i_0}, \overline{f_2}, \overline{a}, \overline{b}, \overline{c}, \overline{d}, \overline{f_1}\}$$

La partition de l'ensemble d'états correspondante est donnée par

$$\pi_*^0 = m_{I-S}(\pi_\gamma) = \{\overline{3}, \overline{5}, \overline{1}, \overline{2}, \overline{4}\}$$

A présent, la plus grande partition  $\pi$  satisfait la contrainte d'existence

$$\xi^0 = \pi_*^0 \xi^1 = \xi^0 + m(\xi^0) = \{\bar{1}, \bar{2}, \bar{4}, \bar{3}, \bar{5}\} = \xi^1$$

Puisque  $\xi^0 = \xi^1$  alors  $\pi = \{\bar{1}, \bar{4}, \bar{2}, \bar{3}, \bar{5}\}$ .

La phase de vérification consiste à tester la mesurabilité et le couplage par rapport à  $I_\rho$ .

La partition de S induite par la sortie est donnée par la relation

$$\pi_\lambda = \{\bar{1}, \bar{4}, \bar{2}, \bar{3}, \bar{5}\}$$

La contrainte de mesurabilité est vérifiée par  $\pi$  puisque

$$\pi + \pi_\lambda = \pi_\lambda \neq \mathbb{I}$$

La vérification du critère de couplage consiste à déterminer la partition d'entrées  $\pi_\rho$  qui découple  $I_\rho$

$$\pi_\rho = \{\bar{i}_0, \bar{f}_1, \bar{a}, \bar{b}, \bar{c}, \bar{d}, \bar{f}_2\}$$

La partition de l'ensemble d'états correspondante est donnée par

$$\bar{\pi}_*^0 = m_{I-S}(\pi_\rho) = \{\bar{1}, \bar{4}, \bar{2}, \bar{3}, \bar{5}\}$$

Le critère de couplage par rapport à  $I_\rho$  est vérifié puisque

$$\bar{\pi}_*^0 \not\leq \pi$$

Finalement, la machine séquentielle partielle  $\Sigma_{f_1}$  obtenue par la partition de décomposition  $\pi$  est donnée par la table 4.4

	a	b	c	d	$f_1$	o
$1' = \bar{1}$	3'	-	4'	2'	4'	O
$2' = \bar{2}$	1'	3'	4'	2'	-	Q
$3' = \bar{3}, \bar{5}$	-	4'	-	-	-	N
$4' = \bar{4}$	3'	1'	-	2'	-	O

Tab. 4.4 – Table de transitions de la machine séquentielle partielle  $\Sigma_{f_1}$

La figure 4.9 montre le graphe représentant le sous-modèle découplé par rapport à  $f_2$ .



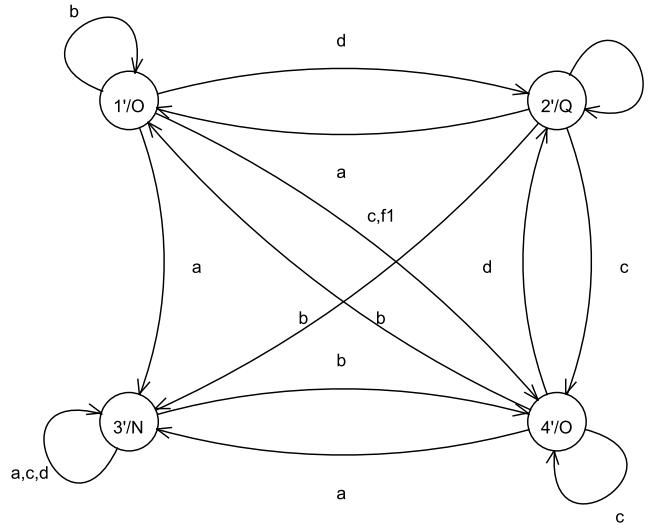


FIG. 4.9 – Graphe d'état de la machine séquentielle partielle  $\Sigma_{f_1}$

A présent nous allons tenter de résoudre le problème inverse en posant  $I_\gamma = \{f_1\}$  et  $I_\rho = \{f_2\}$  et en recherchant la partition de décomposition qui permet le découplage de  $I_\gamma$  et le couplage à  $I_\rho$  :

La première étape consiste à calculer l'ensemble des solutions candidates représenté par la partition  $\pi'_*$ .

La partition de l'ensemble d'entrées  $I$  qui découple de  $I_\gamma$  est donnée par

$$\pi'_\gamma = \{\overline{i_0}, \overline{f_1}, \overline{a}, \overline{b}, \overline{c}, \overline{d}, \overline{f_2}\}$$

La partition de l'ensemble d'états correspondante est donnée par

$$\pi'_* = m_{I-S}(\pi'_\gamma) = \{\overline{1,4}, \overline{2,3}, \overline{5}\}$$

A présent, la plus grande partition  $\pi$  qui satisfait la contrainte d'existence va maintenant être calculée (boucle *tant que* de l'algorithme 8)

$$\xi^0 = \pi'_*$$

$$\xi^1 = \xi^0 + m(\xi^0) = \{\overline{2,3,5}, \overline{1,4}\} + \{\overline{3,5}, \overline{1,2,4}\} = \{\overline{1,4,2,3,5}\}$$

Puisque  $\xi^1 \neq \xi^0$  une seconde itération est nécessaire

$$\xi^2 = \xi^1 + m(\xi^1) = \{\overline{2,3,5}, \overline{1,4}\} + \{\overline{3,5}, \overline{1,2,4}\} = \{\overline{1,4,2,3,5}\} = \xi^1$$

Puisque  $\xi^2 = \xi^1$  alors  $\pi' = \{\overline{1,4,2,3,5}\}$ .

La phase de vérification consiste à tester la mesurabilité et le couplage par rapport à  $I_\rho$ . Rappelons que  $\pi_\lambda$  est la partition de  $S$  induite par la sortie :

$$\pi_\lambda = \{\overline{1,4,2,3,5}\}$$

La contrainte de mesurabilité est vérifiée par  $\pi'$  puisque

$$\pi' + \pi_\lambda = \pi_\lambda = \pi' \neq \mathbb{I}$$

La vérification du critère de couplage consiste à déterminer la partition d'entrées  $\pi_\rho$  qui découple  $I_\rho$ .

$$\pi_\rho = \{\overline{i_0}, \overline{f_2}, \overline{a}, \overline{b}, \overline{c}, \overline{d}, \overline{f_1}\}$$

La partition de l'ensemble d'états correspondante est donnée par

$$\overline{\pi'}^0_* = m_{I-S}(\pi'_\rho) = \{\overline{3}, \overline{5}, \overline{1}, \overline{2}, \overline{4}\}$$

Le critère de couplage par rapport à  $I_\rho$  **n'est pas vérifié** puisque

$$\overline{\pi'}^0_* \leq \pi'$$

Finalement, la machine séquentielle partielle  $\Sigma_{f_2}$  obtenue par la partition de décomposition  $\pi'$  est donnée par la table 4.5

	a	b	c	d	$f_2$	o
$1'' = \overline{1, 4}$	$3''$	-	-	$2''$	-	O
$2'' = \overline{2}$	$1''$	$3''$	$1''$	-	-	Q
$3'' = \overline{3, 5}$	-	$1''$	-	-	-	N

TAB. 4.5 – Table de transitions  $\Sigma_{f_2}$

La figure 4.10 montre le graphe de  $\Sigma_{f_2}$ . La machine séquentielle partielle découplée de  $f_1$  existe mais malheureusement elle est également découplée de  $f_2$ . Elle ne peut donc pas servir à détecter l'événement  $f_1$ .

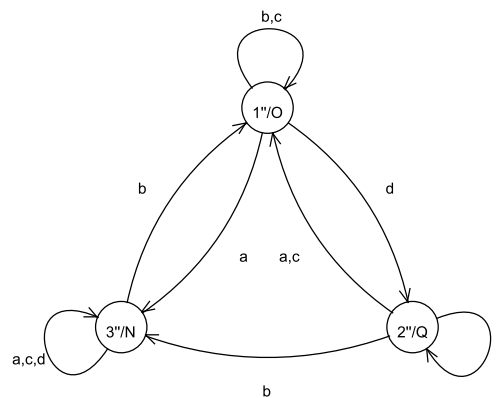


FIG. 4.10 – Graphe d'état de la machine séquentielle partielle  $\Sigma_{f_2}$

### 4.4.3 Conclusion

La décomposition des systèmes représentés par des machines séquentielles a été abordée dans cette section. La méthodologie de décomposition présentée pour le cas général a été reprise et formulée en utilisant les outils de l'algèbre des paires.

La méthodologie en elle-même n'est pas nouvelle, puisque le principe de décomposition de machines séquentielles qui utilise la propriété de substitution a déjà été présenté dans l'ouvrage [HS66]. Par contre, l'objectif poursuivi, c'est-à-dire l'obtention de machines séquentielles découplées, et les critères de décomposition sont nouveaux dans le cadre événementiel. En outre, une extension de la méthode de décomposition classique a été proposée, basée sur le principe de « l'injection de sorties » qui est relativement connu dans le domaine des modèles temporels, mais qui est nouveau dans le domaine des modèles événementiels. Par ailleurs, une analogie entre la propriété de substitution des machines séquentielles et la propriété d'invariance des fonctions temporelles a été mise en évidence.

Un dernier point mérite par ailleurs d'être souligné : la méthodologie de décomposition est strictement identique au cas temporel, y compris dans ses aspects mathématiques, puisque l'algèbre des paires et l'algèbre des fonctions sont similaires sauf en ce qui concerne les aspects calculatoires des différents opérateurs.

## 4.5 Conclusion générale

Dans ce chapitre, les contributions majeures de cette thèse ont été présentées

1. Une méthodologie générale de décomposition avec critères de couplage et de découplage par rapport aux entrées a été présentée. Les idées classiques de décomposition d'automates formulées dans [HS66] et son extension aux modèles temporels présentée dans [Shu91, ZS93] et complétée dans [Zhi01, SZ05, Zhi06, ZM04] ont été utilisées pour généraliser la méthodologie de décomposition aux modèles comportementaux analytiques. La méthodologie a été formulée avec un haut degré d'abstraction en utilisant des notions de la théorie des ensembles, pour proposer des équivalents aux conditions de décomposition existantes. Les conditions de décomposition proposées s'appliquent à tout les types de modèles comportementaux déterministes.
2. La méthodologie de décomposition proposée dans [Shu91],..., [Zhi06], a été étudiée et des améliorations sur divers aspects, en particulier calculatoires, ont été proposés. Ces améliorations ont pour vocation de compléter les travaux déjà existants :
  - Le premier aspect développé est le calcul des différents opérateurs de l'algèbre des fonctions qui n'est pas abordé dans la littérature. En outre, une idée nouvelle a été proposée pour le calcul de l'opérateur  $m$  en utilisant la notion d'équivalence de fonctions.
  - Le deuxième aspect développé est une proposition qui concerne la première étape de l'algorithme de décomposition. Cette proposition a pour objectif l'optimisation, en terme de dimension, de l'ensemble des solutions candidates à

la décomposition. Des techniques d'élimination de variables sont utilisées à cet effet. Le cas de l'injection linéaire de défaillances a été traité en utilisant l'algorithme d'élimination de Gauss. L'injection non linéaire polynomiale reste pour l'instant une perspective en phase de réalisation. Les idées utilisées proviennent des techniques d'élimination de variables par bases de Groebner.

3. La partie théorique de la décomposition des machines séquentielles [HS66] propose une approche de décomposition avec des critères de découplage. Cette approche vient compléter l'autre contribution de cette thèse qui est une nouvelle approche de détection de défauts par génération d'indicateurs de défaillances. L'algorithme de décomposition a été écrit de manière à reprendre les étapes de la décomposition de modèles temporels. Cette approche a permis d'étendre des méthodes connues dans le domaine des modèles temporels à celui des modèles événementiels. Par exemple, l'injection de sorties du modèle à décomposer pour relâcher la contrainte d'existence est une notion qui existe dans la littérature [Isi95, DI01], mais qui n'est pas utilisée dans le cadre des machines séquentielles. Ce point a été discuté et une proposition concrète a été faite pour utiliser l'injection de sorties dans les machines séquentielles pour le relâchement de la contrainte d'existence.



# Chapitre 5

## Illustration sur le système à trois cuves

### 5.1 Introduction

Cette partie est dédiée à l'étude d'un exemple illustratif. Le processus présenté figure (5.1) est décrit en utilisant deux représentations de niveau d'abstraction différent :

- Une représentation continue par un modèle non linéaire
- Une représentation événementielle par automate à état fini

Il s'agit d'un système à trois cuves classique [KSGF99, DJDZ99, JPS03], représenté par la figure (5.1)

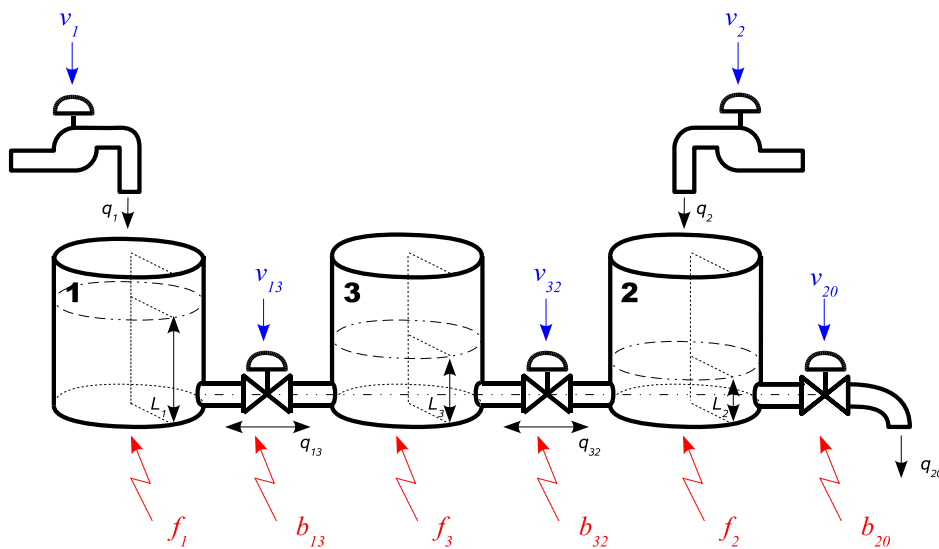


FIG. 5.1 – Représentation du processus à trois cuves

Le système est constitué de cuves reliées par des conduites. Les cuves 1 et 2 sont alimentées par deux arrivées de liquide et le débit est contrôlé par les vannes  $v_1$  et  $v_2$ . Les conduites reliant les cuves sont munies de vannes commandées  $v_{13}$  et  $v_{32}$ . La vanne  $v_{20}$  sert à évacuer le liquide. Les cuves 1 et 3 sont munies de capteurs de niveau. Le diagramme fonctionnel du processus est présenté par la figure (5.2).

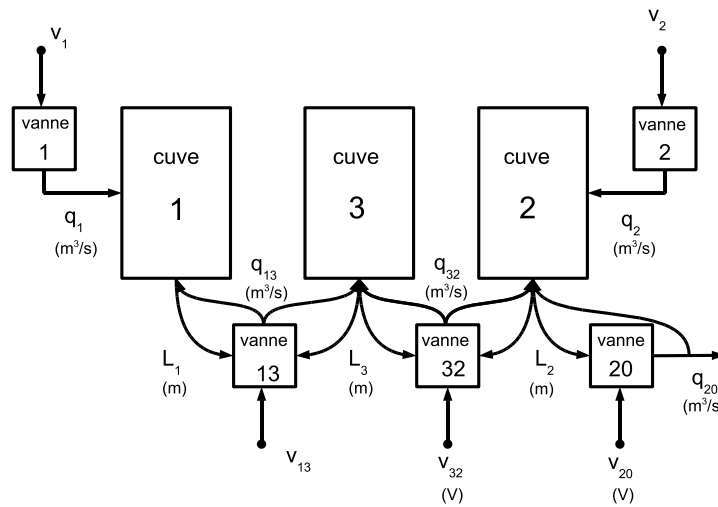


FIG. 5.2 – Diagramme fonctionnel du processus 3 cuves

## 5.2 Modélisation temporelle

### 5.2.1 Modèle de bon fonctionnement

Le théorème de conservation des masses permet d'écrire

(Masse du liquide dans la cuve) = (Somme des masses de liquide entrant) - (Somme des masses de liquide sortant)

c'est-à-dire

$$\rho LS = \rho \sum_i V e_i - \rho \sum_j V s_j$$

où  $L$  est le niveau du liquide,  $S$  la surface de la cuve,  $V e$  et  $V s$  sont les volumes entrants et sortants, et  $\rho$  la masse volumique du liquide. La variation du volume du liquide dans une cuve est donnée par

$$\frac{dL}{dt} S = \sum_i \frac{dV e_i}{dt} - \sum_j \frac{dV s_j}{dt}$$

Notons que  $\frac{dV}{dt} = q$  représente un débit,

$$S \frac{dL}{dt} = \sum q_e - \sum q_s$$

Cette relation permet d'obtenir les relations (5.1) qui décrivent les échanges de liquide entre les trois cuves

$$\begin{cases} S \frac{dL_1}{dt} = q_1 - q_{13} \\ S \frac{dL_2}{dt} = q_2 + q_{32} - q_{20} \\ S \frac{dL_3}{dt} = q_{13} - q_{32} \end{cases} \quad (5.1)$$

où  $L_i$  représente le niveau de la cuve  $i$ ,  $S$  la surface d'une cuve et  $q_i$  et  $q_{ij}$  représentent les flux du liquide d'alimentation à travers les vannes  $i$  ou  $ij$  respectivement. La valeur des débits est calculée en utilisant le théorème de Bernoulli. Le débit  $q_i$  est donné par

$$q_i = \frac{1}{S} K v_i$$

où  $K$  représente le gain de la vanne en  $\left[\frac{l}{s}\right]$  et  $v_i$  la tension de commande des vannes. Le débit  $q_{ij}$  est donné par

$$q_{ij} = \frac{S_p}{S} K v_{ij} \sqrt{2g(L_i - L_j)} \text{ avec } L_i > L_j$$

et

$$q_{ij} = -\frac{S_p}{S} K v_{ij} \sqrt{2g(L_j - L_i)} \text{ avec } L_i < L_j$$

où  $S_p$  représente la section de la conduite,  $g$  l'accélération gravitationnelle terrestre et  $v_{ij}$  représente la commande des vannes (discrète ou continue). Le modèle d'état de bon fonctionnement s'écrit dans l'espace d'état sous la forme

$$\begin{cases} \dot{x}_1 = \frac{1}{S} K v_1 - K v_{13} \frac{S_p}{S} \text{sign}(x_1 - x_3) \sqrt{2g} \sqrt{|x_1 - x_3|} \\ \dot{x}_2 = \frac{1}{S} K v_2 + K v_{32} \frac{S_p}{S} \text{sign}(x_3 - x_2) \sqrt{2g} \sqrt{|x_3 - x_2|} - K v_{20} \frac{S_p}{S} \sqrt{2g} \sqrt{x_2} \\ \dot{x}_3 = K v_{13} \frac{S_p}{S} \text{sign}(x_1 - x_3) \sqrt{2g} \sqrt{|x_1 - x_3|} \dots \\ \dots - K v_{32} \frac{S_p}{S} \text{sign}(x_3 - x_2) \sqrt{2g} \sqrt{|x_3 - x_2|} \end{cases} \quad \begin{cases} y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x \end{cases} \quad (5.2)$$

Les variables d'état  $x_1$ ,  $x_2$  et  $x_3$  correspondent aux niveaux  $L_1$ ,  $L_2$  et  $L_3$  respectivement. Les niveaux des cuves 1 et 2 sont mesurés.

### 5.2.2 Modèle de fonctionnement défaillant

Les défauts considérés dans le processus sont des fuites dans les cuves, notées  $f_i$ , et des bouchages, partiels ou totaux, des conduites  $c_{ij}$ , que l'on note  $b_{ij}$ . Le vecteur des défaillances est donné par

$$\rho = [f_1 \quad f_2 \quad f_3 \quad b_{13} \quad b_{32} \quad b_{20}]^T$$

où  $f_1$ ,  $f_2$  et  $f_3$  sont des fuites dans les cuves 1, 2 et 3 respectivement, et  $b_{13}$ ,  $b_{32}$  et  $b_{20}$  sont des bouchages dans les conduites  $c_{13}$ ,  $c_{32}$  et  $c_{20}$ . L'influence des fuites sur le modèle est représentée par un débit sortant  $q_{f_i}$  tel que

$$q_{f_i} = \frac{f_i}{S} \sqrt{2g} \sqrt{L_i}$$



où  $S$  est la surface de la cuve  $i$  qui fuit et  $L_i$  le niveau du liquide dans cette cuve. L'encrassement d'une conduite  $c_{ij}$  est représenté par la diminution de la surface du tube  $S'_p = S_p(1 - b_{ij})$  avec  $b_{ij} \in [0, 1]$ ; si la conduite est totalement bouchée alors  $b_{ij} = 1$  et si elle est encrassée  $0 < b_{ij} < 1$ . L'expression du flux dans la conduite qui relie les cuves  $i$  et  $j$  devient

$$q_{ij} = (1 - b_{ij}) \frac{S_p}{S} \text{sign}(L_i - L_j) \sqrt{2g} \sqrt{|L_i - L_j|}$$

Le modèle d'état de fonctionnement défaillant est donné par les équations

$$\left\{ \begin{array}{l} \dot{x}_1 = \frac{1}{S} K v_1 - K v_{13} (1 - b_{13}) \frac{S_p}{S} \text{sign}(x_1 - x_3) \sqrt{2g} \sqrt{|x_1 - x_3|} - \frac{f_1}{S} \sqrt{2g} \sqrt{x_1} \\ \dot{x}_2 = \frac{1}{S} K v_2 + (1 - b_{32}) K v_{32} \frac{S_p}{S} \text{sign}(x_3 - x_2) \sqrt{2g} \sqrt{|x_3 - x_2|} \cdots \\ \quad \cdots - \frac{f_2}{S} \sqrt{2g} \sqrt{x_2} - K v_{20} (1 - b_{20}) \frac{S_p}{S} \sqrt{2g} \sqrt{x_2} \\ \dot{x}_3 = K v_{13} (1 - b_{13}) \frac{S_p}{S} \text{sign}(x_1 - x_3) \sqrt{2g} \sqrt{|x_1 - x_3|} \cdots \\ \quad \cdots - (1 - b_{32}) K v_{32} \frac{S_p}{S} \text{sign}(x_3 - x_2) \sqrt{2g} \sqrt{|x_3 - x_2|} - \frac{f_3}{S} \sqrt{2g} \sqrt{x_3} \end{array} \right. \quad (5.3)$$

Bien entendu, dans le cas non défaillant ( $f_i = 0$  et  $b_{ij} = 0$ ) nous retrouvons le modèle de fonctionnement normal (5.2).

### 5.2.3 Décomposition en sous-modèles découplés

Conformément à la méthodologie présentée au chapitre précédent, nous allons construire un banc de sous-systèmes découplés des sous-ensembles de pannes qui permettra de calculer des résidus robustes. L'objectif est de synthétiser six sous-modèles découplés respectivement de  $f_1$ ,  $f_2$ ,  $f_3$ ,  $b_{13}$ ,  $b_{32}$ ,  $b_{20}$ . Ces sous-modèles sont notés respectivement :

$$\Sigma_{f_1}, \Sigma_{f_2}, \Sigma_{f_3}, \Sigma_{b_{13}}, \Sigma_{b_{32}}, \Sigma_{b_{20}}$$

Chaque sous-modèle  $\Sigma_{def_i}$  devra être découplé de la défaillance  $def_i$  et pourra être couplé aux autres défaillances selon le tableau 5.1.

	$f_1$	$f_2$	$f_3$	$b_{13}$	$b_{32}$	$b_{20}$
$\Sigma_{f_1}$	0	X	X	X	X	X
$\Sigma_{f_2}$	X	0	X	X	X	X
$\Sigma_{f_3}$	X	X	0	X	X	X
$\Sigma_{b_{13}}$	X	X	X	0	X	X
$\Sigma_{b_{32}}$	X	X	X	X	0	X
$\Sigma_{b_{20}}$	X	X	X	X	X	0

TAB. 5.1 – Table représentant les couplages des sous-modèles

où les 0 signifient un découplage et les X signifient indifféremment un couplage ou un découplage. Il est nécessaire que la matrice de signature effectivement obtenue à partir des sous-modèles soit de rang plein structurellement pour pouvoir localiser toutes les défaillances.

### Analyse de robustesse et de sensibilité

Nous allons déterminer ici les fonctions qui découplent formellement les différentes défaillances selon la relation

$$\frac{\partial}{\partial \gamma} \frac{d\phi_\gamma^0}{dx} f(x, u, \gamma, \rho) = 0 \quad (5.4)$$

où  $\gamma$  est le vecteur de défaillances à découpler et  $f$  la fonction d'état. Pour des raisons de clarté nous allons détailler la recherche de la fonction de décomposition qui permet d'obtenir le sous-modèle découplé de  $b_{13}$  et donner les résultats pour les autres sous-modèles. En appliquant directement la relation (5.4) qui permet de sélectionner les états formellement découplés de  $b_{13}$ , nous obtenons

$$\phi_{b_{13}}^0 = [x_2] \quad (5.5)$$

Pour augmenter la dimension de la fonction  $\phi_{b_{13}}^0$ , l'élimination de variables est utilisée pour déterminer les combinaisons des fonctions d'état qui éliminent formellement la défaillance  $b_{13}$ . Nous obtenons alors

$$\phi_{b_{13}}^0 = \begin{bmatrix} x_2 \\ x_1 + x_3 \end{bmatrix} \quad (5.6)$$

Pour vérifier que les sous-modèles sont couplés aux défaillances à détecter, la condition (5.7) est nécessaire

$$\frac{\partial}{\partial \rho} \frac{d\phi_{def}}{dx} f(x, u, \gamma, \rho) \neq 0 \quad (5.7)$$

où  $\phi_{def}$  est la fonction de décomposition qui permet d'obtenir le sous-modèle  $\Sigma_{def}$ . En appliquant (5.7), les vecteurs de couplage  $\beta_{def}$  sont déterminés

$$\beta_{b_{13}}^0 = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} \quad (5.8)$$

Le sous-modèle  $\Sigma_{b_{13}}$  est couplé à la défaillance  $b_{13}$  si  $\phi_{b_{13}}^0 \not\perp \beta_{b_{13}}^0$ .

#### 5.2.4 Synthèse de sous-modèles découplés

Nous allons déterminer à présent les fonctions  $\phi_{def}^*$  qui vérifient les propriétés d'invariance selon le théorème 4.2 en utilisant l'itération

$$\alpha_{i+1} = \alpha_i \square m(\alpha_i \otimes h)$$

avec  $\alpha_0 = \phi_{b_{13}}^0$  et  $\phi_{b_{13}}^* = \alpha_k$  tel que  $\alpha_k = \alpha_{k+1}$ .

Appliqué à notre cas, la première itération de l'algorithme nous donne

$$\begin{aligned}\alpha_1 &= \begin{bmatrix} x_2 \\ x_1 + x_3 \end{bmatrix} \square m \left( \begin{bmatrix} x_2 \\ x_1 + x_3 \end{bmatrix} \otimes \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right) = \begin{bmatrix} x_2 \\ x_1 + x_3 \end{bmatrix} \square m \left( \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \right) \\ &= \begin{bmatrix} x_2 \\ x_1 + x_3 \end{bmatrix} \square \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_1 + x_3 \end{bmatrix}\end{aligned}$$

Nous obtenons  $\alpha_1 = \alpha_0$  alors  $\phi_{b_{13}}^* = \begin{bmatrix} x_2 \\ x_1 + x_3 \end{bmatrix}$

En répétant ces étapes pour le reste des défaillances, nous obtenons les fonctions  $\phi^*$  suivantes

$$\begin{array}{ll} \Sigma_{f_1} : & \phi_{f_1}^* = \begin{bmatrix} x_2 \\ x_3 \end{bmatrix} & \Sigma_{f_2} : & \phi_{f_2}^* = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} \\ \Sigma_{f_3} : & \phi_{f_3}^* = [\emptyset] & \Sigma_{b_{13}} : & \phi_{b_{13}}^* = \begin{bmatrix} x_2 \\ x_1 + x_3 \end{bmatrix} \\ \Sigma_{b_{32}} : & \phi_{b_{32}}^* = \begin{bmatrix} x_1 \\ x_2 + x_3 \end{bmatrix} & \Sigma_{b_{20}} : & \phi_{b_{20}}^* = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}\end{array}$$

Notons que cinq fonctions sont obtenues au total, puisqu'il n'existe pas de fonction qui découple de  $f_3$ . Par ailleurs, deux des fonctions obtenues  $\phi_{f_2}^*$  et  $\phi_{b_{20}}^*$  sont identiques ce qui implique que  $\Sigma_{f_2}$  est identique à  $\Sigma_{b_{20}}$ . Par conséquent, quatre sous-modèles peuvent être obtenus dans le meilleurs des cas.

L'obtention de chaque modèle se fait en utilisant l'algorithme 5 qui permet de construire la fonction de décomposition finale. Nous détaillons le cas de  $\Sigma_{b_{13}}$  et donnons les résultats pour les trois sous-modèles restants.

**a. Synthèse de  $\Sigma_{b_{13}}$**  Les étapes pour obtenir la fonction de décomposition  $\phi$  sont énumérées ci-dessous

1. Sélectionner une fonction  $\phi^1$  qui satisfait le critère de mesurabilité :

$$\phi^1 = \phi_{b_{13}}^* \square h = [x_2]$$

2. Vérifier le critère d'invariance pour  $\phi^1$  :

$$\phi^1 \otimes h = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \stackrel{?}{\leq} M([x_2]) = \begin{bmatrix} \text{sign}(x_1 - x_3) \sqrt{|x_1 - x_3|} \\ \sqrt{|x_2|} \end{bmatrix}$$

Le critère n'est pas vérifié.

3. Une deuxième composante  $\phi^2$  est choisie telle que  $\phi^2 \otimes \phi^1 \otimes h \leq M(\phi^1)$ .  $\phi^2$  est donnée par  $\phi^2 = [x_1 + x_3]$
4. Le critère d'invariance est vérifié pour  $\phi^1 \otimes \phi^2$  puisque la relation

$$\phi^2 \otimes \phi^1 \otimes h = \begin{bmatrix} x_1 \\ x_2 \\ x_1 + x_3 \end{bmatrix} \stackrel{?}{\leq} M([x_1 + x_3]) = [\text{sign}(x_3 - x_2) \sqrt{|x_3 - x_2|}]$$

est vérifiée.

5. La fonction de décomposition est donnée par

$$\phi = \phi^1 \otimes \phi^2 = \begin{bmatrix} x_2 \\ x_1 + x_3 \end{bmatrix}$$

Le sous-modèle  $\Sigma_{b_{13}}$  est donné par les relations

$$\Sigma_{b_{13}} \begin{cases} \dot{z}_1 = \frac{1}{S} K v_2 + K v_{32} \frac{S_p}{S} \text{sign}(z_2 - y_1 - y_2) \sqrt{2g} \sqrt{|z_2 - y_1 - y_2|} \cdots \\ \quad \cdots - K v_{20} \frac{S_p}{S} \sqrt{2g} \sqrt{|y_2|} \\ \dot{z}_2 = \frac{1}{S} K v_1 - K v_{32} \frac{S_p}{S} \text{sign}(z_2 - y_1 - y_2) \sqrt{2g} \sqrt{|z_2 - y_1 - y_2|} \end{cases}$$

$$\{y_{b_{13}} = z_1$$

**b. Synthèse de  $\Sigma_{f_1}$**  Le sous-modèle  $\Sigma_{f_1}$  est donné par les relations

$$\Sigma_{f_1} \begin{cases} \dot{z}_1 = \frac{1}{S} K v_2 + K v_{32} \frac{S_p}{S} \text{sign}(z_2 - y_2) \sqrt{2g} \sqrt{|z_2 - y_2|} - K v_{20} \frac{S_p}{S} \sqrt{2g} \sqrt{|y_2|} \\ \dot{z}_2 = K v_{12} \frac{S_p}{S} \text{sign}(y_1 - z_2) \sqrt{2g} \sqrt{|y_1 - z_2|} \cdots \\ \quad \cdots - K v_{32} \frac{S_p}{S} \text{sign}(z_2 - y_2) \sqrt{2g} \sqrt{|z_2 - y_2|} \end{cases}$$

$$\{y_{f_1} = z_1$$

**c. Synthèse de  $\Sigma_{f_2, b_{20}}$**  Puisque les sous-modèles  $\Sigma_{f_2}$  et  $\Sigma_{b_{20}}$  sont identiques, ils seront remplacés par un seul sous-modèle  $\Sigma_{f_2, b_{20}}$  qui est donné par les relations

$$\Sigma_{f_2, b_{20}} \begin{cases} \dot{z}_1 = \frac{1}{S} K v_1 - K v_{13} \frac{S_p}{S} \text{sign}(y_1 - z_2) \sqrt{2g} \sqrt{|y_1 - z_2|} \\ \dot{z}_2 = K v_{12} \frac{S_p}{S} \text{sign}(y_1 - z_2) \sqrt{2g} \sqrt{|y_1 - z_2|} \cdots \\ \quad \cdots - K v_{32} \frac{S_p}{S} \text{sign}(z_2 - y_2) \sqrt{2g} \sqrt{|z_2 - y_2|} \end{cases}$$

$$\{y_{f_2, b_{20}} = z_1$$

d. **Synthèse de  $\Sigma_{b_{32}}$**  Le sous-modèle  $\Sigma_{b_{32}}$  est donné par les relations

$$\Sigma_{b_{32}} \begin{cases} \dot{z}_1 &= \frac{1}{S}Kv_1 - Kv_{13} \frac{S_p}{S} \text{sign}(y_1 - z_2 - y_2) \sqrt{2g} \sqrt{|y_1 - z_2 - y_2|} \\ \dot{z}_2 &= \frac{1}{S}Kv_2 + Kv_{13} \frac{S_p}{S} \text{sign}(y_1 - z_2 - y_2) \sqrt{2g} \sqrt{|y_1 - z_2 - y_2|} - Kv_{20} \frac{S_p}{S} \sqrt{2g} \sqrt{|y_2|} \end{cases}$$

$$\{y_{b_{32}} = z_1$$

La matrice finale des couplages est donnée par la table (5.2)

	$f_1$	$f_2$	$f_3$	$b_{13}$	$b_{32}$	$b_{20}$
$\Sigma_{f_1}$	0	1	1	1	1	1
$\Sigma_{f_2, b_{20}}$	1	0	1	1	1	0
$\Sigma_{b_{13}}$	1	1	1	0	1	1
$\Sigma_{b_{32}}$	1	1	1	1	0	1

TAB. 5.2 – Matrice de couplage du système à trois cuves

**Remarque 5.2.1** Il n'existe pas de sous-modèle découplé de la défaillance  $f_3$  selon la table (5.2). Si  $f_3$  se produit, alors les sorties de tous les sous-modèles vont réagir.

Par ailleurs, les sous-modèles découplés des défaillances  $f_2$  et  $b_{20}$  sont identiques, et les défaillances ne sont donc pas discernables a priori. Puisque  $f_2$  est une fuite dans la deuxième cuve, elle ne dépend pas des commandes de vanne. Par contre la défaillance de la conduite  $b_{20}$  dépend directement de l'état de la vanne  $v_{20}$  : Si la vanne est fermée, alors  $b_{20}$  n'est pas détectable puisqu'il n'y a pas de flux de liquide. Nous pourrions discerner  $f_2$  de  $b_{20}$  si la vanne  $v_{20}$  est fermée.

### 5.2.5 Génération de résidus

Un banc de générateurs de résidus est construit à partir des sous-modèles :

$$\Sigma_{f_1}, \Sigma_{f_2, b_{20}}, \Sigma_{b_{13}}, \Sigma_{b_{32}}$$

Pour obtenir la convergence en présence de conditions initiales inconnues, nous avons choisi des générateurs de résidus à base d'observateurs d'Utkin [Utk81]. Ces observateurs sont une variante des observateurs discontinus construits selon la théorie des modes glissants. Ce choix est fait par rapport aux équations qui régissent le fonctionnement des sous-modèles, et qui font intervenir des non linéarités non différentiables (fonctions « signe »). La structure d'un observateur d'Utkin est donnée par l'équation 5.9

$$\begin{cases} \dot{\hat{x}}_1 &= \hat{f}_1(\hat{x}_1, \hat{x}_2) + LM \text{sign}(r) \\ \dot{\hat{y}} &= \hat{x}_2 = \hat{f}_2(\hat{x}_1, \hat{x}_2) - M \text{sign}(r) \end{cases} \quad \begin{cases} \hat{y} &= \hat{h}(x) \\ r &= \hat{y} - y \end{cases} \quad (5.9)$$

où L et M sont des coefficients de réglage. Les expressions des quatre générateurs de résidus sont donnés par les équations suivantes :

– Le générateur du résidu robuste à  $f_1$  est donné par :

$$\hat{\Sigma}_{f_1} \begin{cases} \dot{z}_1 = \frac{1}{S}Kv_2 + Kv_{32} \frac{S_p}{S} \text{sign}(z_2 - y_2) \sqrt{2g} \sqrt{|z_2 - y_2|} \cdots \\ \quad \cdots - Kv_{20} \frac{S_p}{S} \sqrt{2g} \sqrt{|y_2|} - M_1 \text{sign}(r_{f_1}) \\ \dot{z}_2 = Kv_{12} \frac{S_p}{S} \text{sign}(y_1 - z_2) \sqrt{2g} \sqrt{|y_1 - z_2|} \cdots \\ \quad \cdots - Kv_{32} \frac{S_p}{S} \text{sign}(z_2 - y_2) \sqrt{2g} \sqrt{|z_2 - y_2|} + L_1 M_1 \text{sign}(r_{f_1}) \end{cases}$$

$$r_{f_1} = z_1 - y_2$$

– Le générateur du résidu robuste à  $f_2$  et à  $b_{20}$  est donné par :

$$\hat{\Sigma}_{f_2, b_{20}} \begin{cases} \dot{z}_1 = \frac{1}{S}Kv_1 - Kv_{13} \frac{S_p}{S} \text{sign}(y_1 - z_2) \sqrt{2g} \sqrt{|y_1 - z_2|} - M_2 \text{sign}(r_{f_2}) \\ \dot{z}_2 = Kv_{12} \frac{S_p}{S} \text{sign}(y_1 - z_2) \sqrt{2g} \sqrt{|y_1 - z_2|} \cdots \\ \quad \cdots - Kv_{32} \frac{S_p}{S} \text{sign}(z_2 - y_2) \sqrt{2g} \sqrt{|z_2 - y_2|} + L_2 M_2 \text{sign}(r_{f_2}) \end{cases}$$

$$r_{f_2, b_{20}} = z_1 - y_1$$

– Le générateur du résidu robuste à  $b_{13}$  est donné par :

$$\hat{\Sigma}_{b_{13}} \begin{cases} \dot{z}_1 = \frac{1}{S}Kv_2 + Kv_{32} \frac{S_p}{S} \text{sign}(z_2 - y_1 - y_2) \sqrt{2g} \sqrt{|z_2 - y_1 - y_2|} \cdots \\ \quad \cdots - Kv_{20} \frac{S_p}{S} \sqrt{2g} \sqrt{|y_2|} - M_3 \text{sign}(r_{b_{13}}) \\ \dot{z}_2 = \frac{1}{S}Kv_1 - Kv_{32} \frac{S_p}{S} \text{sign}(z_2 - y_1 - y_2) \sqrt{2g} \sqrt{|z_2 - y_1 - y_2|} + L_3 M_3 \text{sign}(r_{b_{13}}) \end{cases}$$

$$r_{b_{13}} = z_1 - y_2$$

– Le générateur du résidu robuste à  $b_{32}$  est donné par :

$$\hat{\Sigma}_{b_{32}} \begin{cases} \dot{z}_1 = \frac{1}{S}Kv_1 - Kv_{13} \frac{S_p}{S} \text{sign}(y_1 - z_2 - y_2) \sqrt{2g} \sqrt{|y_1 - z_2 - y_2|} - M_4 \text{sign}(r_{b_{32}}) \\ \dot{z}_2 = \frac{1}{S}Kv_2 + Kv_{13} \frac{S_p}{S} \text{sign}(y_1 - z_2 - y_2) \sqrt{2g} \sqrt{|y_1 - z_2 - y_2|} \cdots \\ \quad \cdots - Kv_{20} \frac{S_p}{S} \sqrt{2g} \sqrt{|y_2|} + L_4 M_4 \text{sign}(r_{b_{32}}) \end{cases}$$

$$r_{b_{32}} = z_1 - y_1$$

## 5.2.6 Simulation

Les simulations sont réalisées avec le logiciel MATLAB/SIMULINK avec les paramètres suivants :

- $K = 1 \left[ \frac{l}{s} \right]$
- $S = 0.5\pi^2 = \pi [m^2]$
- $S_p = 0.05\pi^2 = \frac{\pi}{10} [m^2]$
- $g = 9.81 \left[ \frac{m}{s^2} \right]$

Les coefficients de réglage des générateurs de résidus sont donnés par

$$L_i = 0.7 \text{ et } M_i = 0.3 \quad \forall i = 1..4$$

La figure 5.3 représente le schéma de simulation.

Les figures ci-dessous illustrent le comportement des générateurs de résidus pour différents scénarios avec des sorties bruitées par un bruit blanc d'amplitude 0.15. La simulation dure 300 secondes avec un pas d'échantillonnage de 0,5 seconde.

- En fonctionnement normal (figure 5.4);

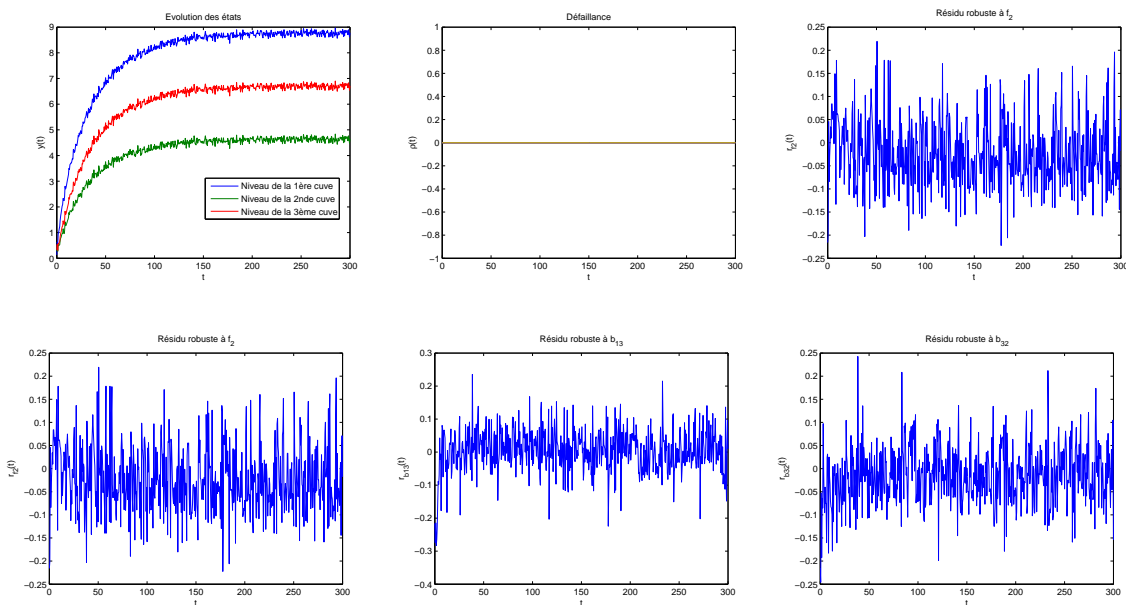


FIG. 5.4 – Résidus dans le cas normal

- Pour une défaillance permanente  $b_{13}$  (figure 5.5);

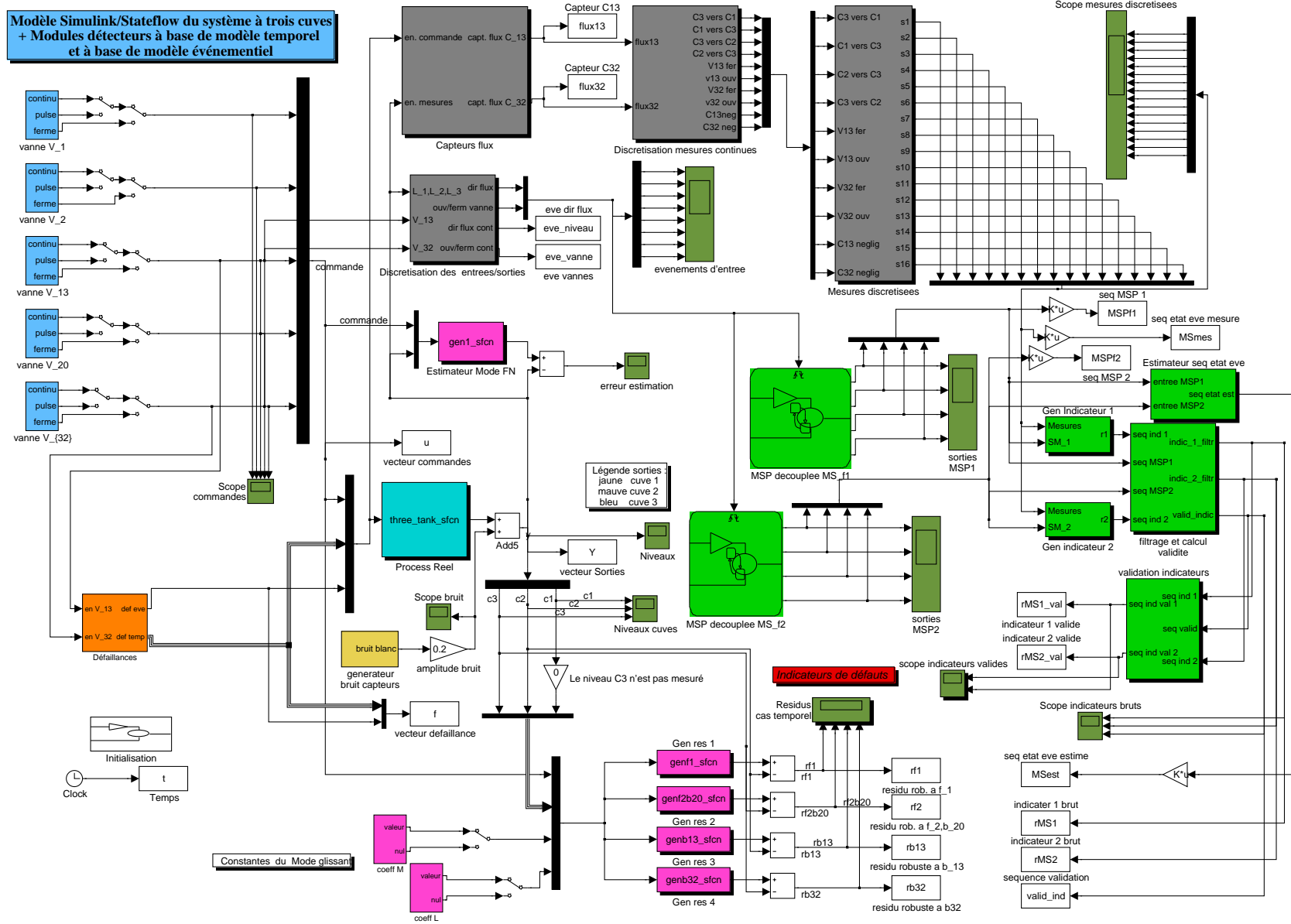


FIG. 5.3 – Schéma de simulation Simulink



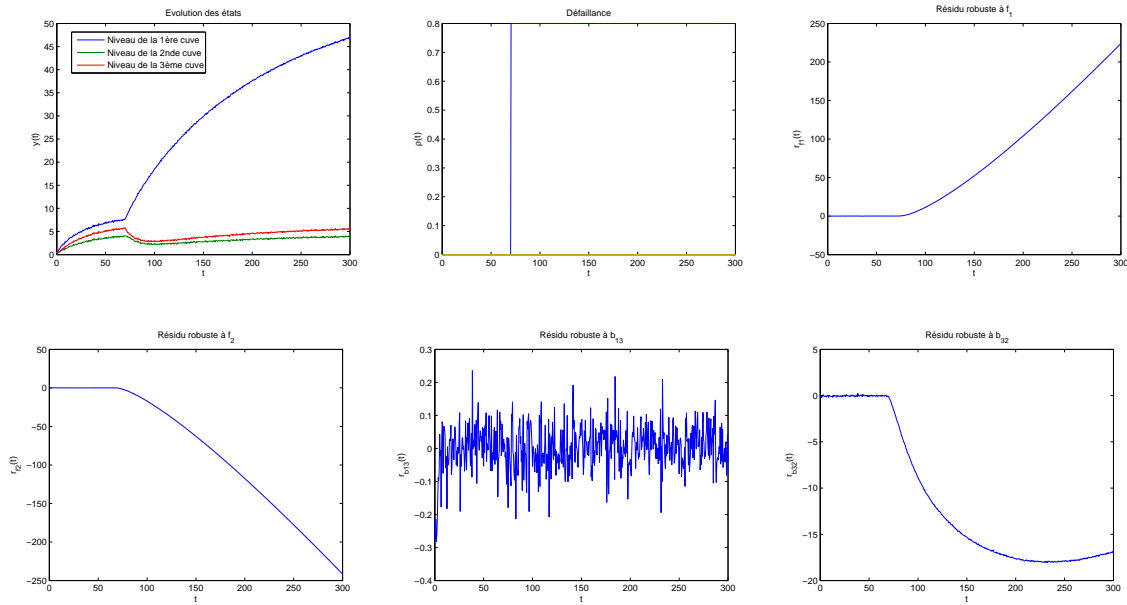


FIG. 5.5 – Résidus dans le cas d’une défaillance  $b_{13}$  permanente

– Pour une défaillance intermittente  $b_{13}$  (figure 5.6) ;

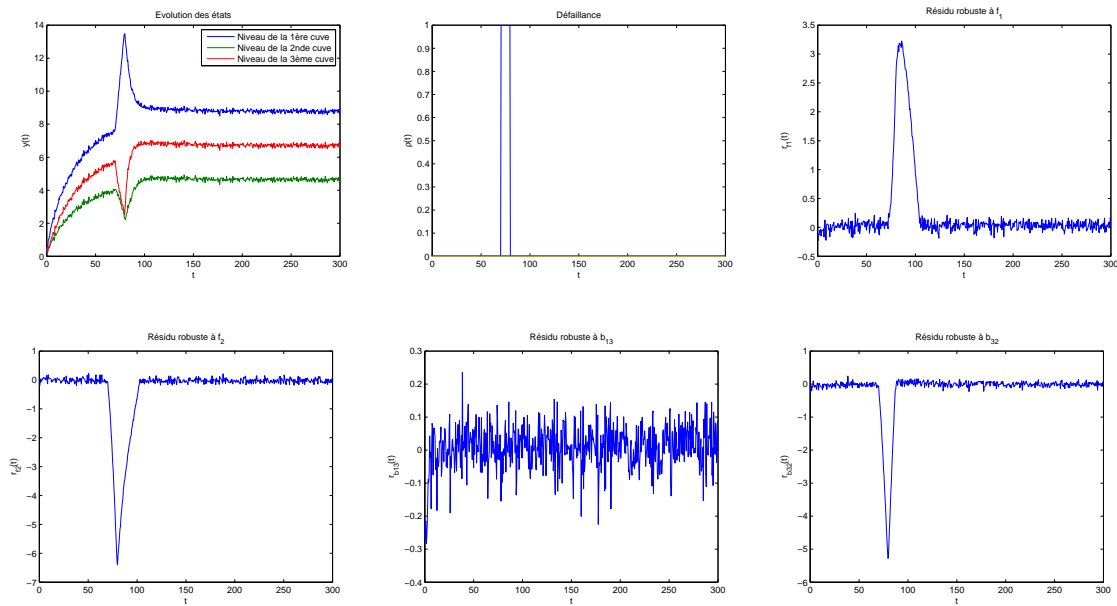


FIG. 5.6 – Résidus dans le cas d’une défaillance  $b_{13}$  temporaire

– Les autres simulations sont regroupées en annexe C.

Nous remarquons que dans chaque cas, les résidus sensibles à une défaillance réagissent dès que la défaillance en question est différente de zéro, et cela pour les deux cas de figure, pour des défaillances permanentes et pour des défaillances temporaires.

## 5.3 Modélisation événementielle

### 5.3.1 Description du modèle événementiel de bon fonctionnement

Pour illustrer l'approche de décomposition dans le cadre événementiel proposée dans le chapitre précédent, nous allons représenter l'évolution des directions des débits dans les conduites du processus « 3 cuves » par une machine séquentielle pour détecter et localiser les défaillances des deux vannes  $V_{13}$  et  $V_{32}$ . Deux capteurs sont placés sur les conduites entre les cuves 1 et 3 et les cuves 2 et 3. Ces capteurs indiquent la direction du flux dans la conduite, par exemple le capteur placé dans  $c_{13}$  peut indiquer que le flux circule de la cuve 1 vers la cuve 3, ou de la cuve 3 vers la cuve 1, ou qu'il n'y a pas de flux, c'est à dire qu'il n'y a pas de mouvement de liquide. Si le débit dans une conduite est nul, alors la vanne correspondante est considérée comme fermée. Si le débit n'est pas nul, la vanne est considérée comme ouverte.

La représentation choisie décrit les quatre principaux modes de fonctionnement du système en fonction de la direction des flux dans les conduites d'eau :

- Etat 1 : La direction du flux va de la cuve 1 à la cuve 3 et de la cuve 3 vers la cuve 2 :  $1 \rightarrow 3$  et  $3 \rightarrow 2$
- Etat 2 : La direction du flux va de la cuve 3 à la cuve 1 et de la cuve 3 vers la cuve 2 :  $1 \leftarrow 3$  et  $3 \rightarrow 2$
- Etat 3 : La direction du flux va de la cuve 1 à la cuve 3 et de la cuve 2 vers la cuve 3 :  $1 \rightarrow 3$  et  $3 \leftarrow 2$
- Etat 4 : La direction du flux va de la cuve 3 à la cuve 1 et de la cuve 2 vers la cuve 3 :  $1 \leftarrow 3$  et  $3 \leftarrow 2$

En incorporant l'état des vannes  $V_{13}$  et  $V_{32}$  posées sur les conduites qui relient les cuves 1 et 3 et les cuves 3 et 2, deux états supplémentaires sont ajoutés :

- Un état noté  $-|-$  qui représente l'absence de flux entre les cuves à cause de la fermeture de la vanne correspondante.
- Un état noté  $-X-$  qui représente le régime transitoire dans les instants qui suivent l'ouverture de la vanne correspondante.

Ce qui nous donne au total un système à 16 états :

- Etat 5 :  $1 \rightarrow 3$  et  $2 -|- 3$
- Etat 6 :  $1 \leftarrow 3$  et  $2 -|- 3$
- Etat 7 :  $3 \rightarrow 2$  et  $1 -|- 3$
- Etat 8 :  $3 \leftarrow 2$  et  $1 -|- 3$
- Etat 9 :  $1 -|- 3$  et  $2 -|- 3$
- Etat 10 :  $1 \rightarrow 3$  et  $2 -X- 3$
- Etat 11 :  $1 \leftarrow 3$  et  $2 -X- 3$
- Etat 12 :  $3 \rightarrow 2$  et  $1 -X- 3$
- Etat 13 :  $3 \leftarrow 2$  et  $1 -X- 3$
- Etat 14 :  $1 -X- 3$  et  $2 -|- 3$
- Etat 15 :  $1 -|- 3$  et  $2 -X- 3$
- Etat 16 :  $1 -X- 3$  et  $2 -X- 3$

Les états 1,  $\dots$ , 9 sont des états stables. Les états 10,  $\dots$ , 16 sont des états transitoires de courte durée (états instables).

Le fonctionnement de la machine est décrit par la table de transitions (5.10) et la figure (5.7).

	$a$	$b$	$c$	$d$	$e$	$f$	$g$	$h$
1	2	-	3	-	7	-	5	-
2	-	1	4	-	7	-	6	-
3	4	-	-	1	8	-	5	-
4	-	3	-	2	8	-	6	-
5	6	-	-	-	9	-	-	10
6	-	5	-	-	9	-	-	11
7	-	-	8	-	-	12	9	-
8	-	-	-	7	-	13	9	-
9	-	-	-	-	-	14	-	15
10	-	11	3	1	15	-	5	-
11	10	-	4	2	15	-	6	-
12	2	1	13	-	7	-	14	-
13	4	3	-	12	8	-	14	-
14	6	5	-	-	9	-	-	16
15	-	-	8	7	-	16	9	-
16	11	10	13	12	15	-	14	-

(5.10)

Où les tirets « - » représentent la persistance de l'état présent. Les actions considérées sont les suivantes :

- $a : L_1 < L_3$  Le niveau de la cuve centrale devient supérieur à celui de la première cuve.
- $b : L_1 > L_3$  Le niveau de la cuve centrale devient inférieur à celui de la première cuve.
- $c : L_2 > L_3$  Le niveau de la cuve centrale devient inférieur à celui de la dernière cuve.
- $d : L_2 < L_3$  Le niveau de la cuve centrale devient supérieur à celui de la dernière cuve.
- $e : V_{13} = 0$  La première vanne  $V_{13}$  est fermée.
- $f : V_{13} = 1$  La première vanne est ouverte.
- $g : V_{32} = 0$  La seconde vanne  $V_{32}$  est fermée.
- $h : V_{32} = 1$  La seconde vanne est ouverte.

La correspondance entre les sorties de  $\Sigma$  et les mesures des capteurs installés sur le processus est donnée par la table (5.3).  $V_{13}$  et  $V_{32}$  représentent les vannes et  $C_1, C_2, C_3$  représentent les cuves.

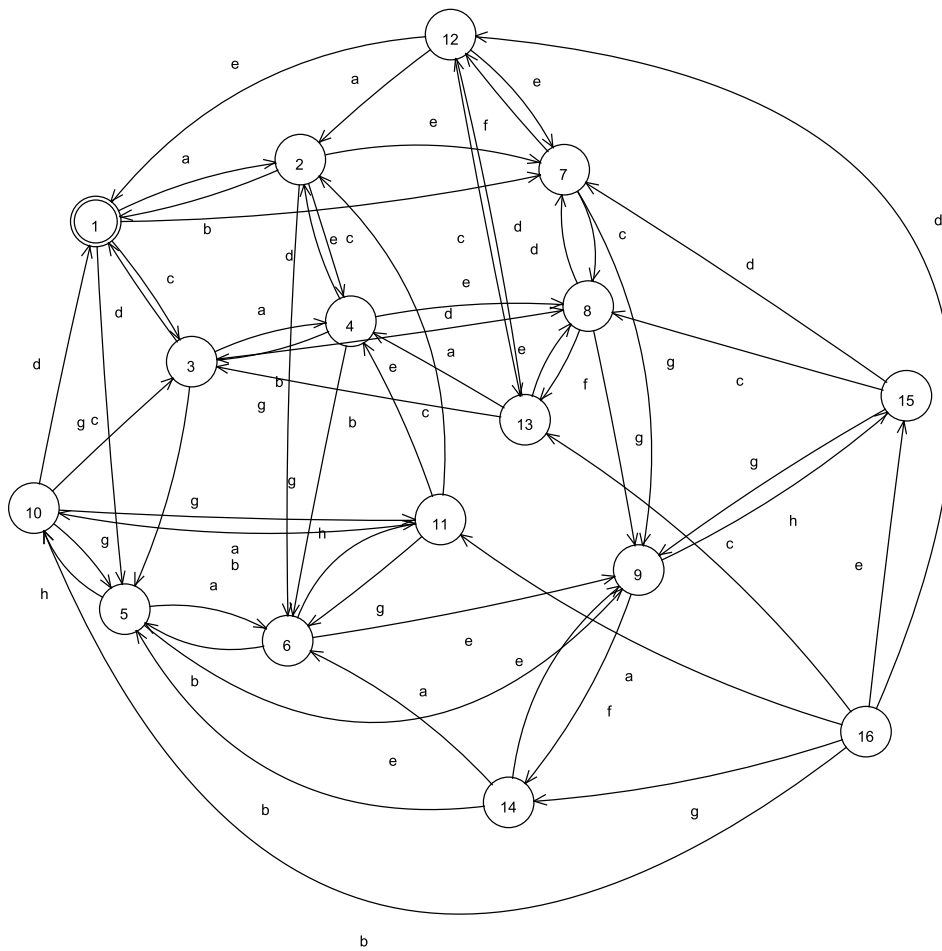


FIG. 5.7 – Représentation du modèle de fonctionnement normal du processus 3 cuves

$\Sigma$	
Etats	Sorties
1	$V_{13}$ et $V_{32}$ ouvertes, flux de $C_1$ vers $C_3$ et de $C_3$ vers $C_2$
2	$V_{13}$ et $V_{32}$ ouvertes, flux de $C_3$ vers $C_1$ et de $C_3$ vers $C_2$
3	$V_{13}$ et $V_{32}$ ouvertes, flux de $C_1$ vers $C_3$ et de $C_2$ vers $C_3$
4	$V_{13}$ et $V_{32}$ ouvertes, le flux de $C_3$ vers $C_1$ et $C_2$ vers $C_3$
5	$V_{32}$ fermée $V_{13}$ ouverte, flux de $C_1$ vers $C_3$
6	$V_{32}$ fermée $V_{13}$ ouverte, flux de $C_3$ vers $C_1$
7	$V_{13}$ fermée, $V_{32}$ ouverte, flux de $C_3$ vers $C_2$
8	$V_{13}$ fermée, $V_{32}$ ouverte, flux $C_2$ vers $C_3$
9	$V_{13}$ et $V_{32}$ fermées
10	$V_{13}$ et $V_{32}$ ouvertes, flux inconnu entre $C_2$ et $C_3$ , flux de $C_1$ vers $C_3$
11	$V_{13}$ et $V_{32}$ ouvertes, flux inconnu entre $C_2$ et $C_3$ , flux de $C_3$ vers $C_1$
12	$V_{13}$ et $V_{32}$ ouvertes, flux inconnu entre $C_1$ et $C_3$ , flux de $C_3$ vers $C_2$
13	$V_{13}$ et $V_{32}$ ouvertes, flux inconnu entre $C_1$ et $C_3$ , flux $C_2$ vers $C_3$
14	$V_{13}$ ouverte, $V_{32}$ fermée, flux inconnu entre $C_1$ et $C_3$
15	$V_{32}$ ouverte, $V_{13}$ fermée, flux inconnu entre $C_2$ et $C_3$
16	$V_{13}$ et $V_{32}$ ouvertes, flux inconnu entre toutes les cuves

TAB. 5.3 – Correspondance entre les états du modèle et les sorties capteurs

### 5.3.2 Description du modèle de fonctionnement défaillant

Nous considérons deux défaillances  $f_{V_{13}}$  et  $f_{V_{32}}$  qui correspondent à un dysfonctionnement des vannes  $V_{13}$  et  $V_{32}$ . Ces dysfonctionnements correspondent à des fermetures ou à des ouvertures intempestives. Si une défaillance est amenée à se produire sur la vanne, nous en verrions l'influence en analysant la trajectoire que prend le modèle en réponse à une sollicitation de commande. Nous ajoutons un événement  $f$  qui représente la défaillance d'une vanne. Cet événement est placé en parallèle à toutes les transitions qui concernent la vanne (figure 5.8(b)).

**Remarque 5.3.1** Une autre manière possible de représenter le fonctionnement d'une vanne serait de représenter la défaillance par un état défaillant non mesuré (figure 5.9(b)).

L'état vanne bloquée ouverte possède la même sortie que l'état vanne ouverte ; par conséquent, ces deux états peuvent être distingués si l'action fermeture de vanne est demandée. Notons cependant que la description par transitions est plus adaptée pour les défauts du type « ouverture ou fermeture intempestives de la vanne » et que la description par état est plus adaptée pour décrire un blocage de vanne. En règle générale les deux descriptions ne sont pas équivalentes mais complémentaires.

En ajoutant le modèle de fonctionnement défaillant des vannes au modèle de bon fonctionnement du processus représenté figure 5.7, nous obtenons une machine séquentielle de 16 états et de 10 entrées (figure 5.10) :

La table 5.3.2 illustre l'évolution du modèle de fonctionnement défaillant.

Nous proposons de construire un réseau de machines séquentielles partielles qui décrivent le fonctionnement du système en étant chacune découplées de chaque dé-

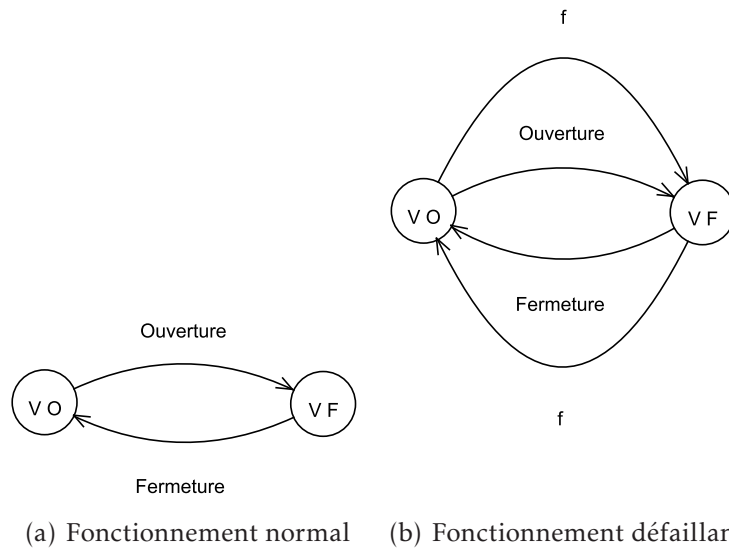


FIG. 5.8 – Modèle événementiel d’une vanne (approche *transitions*)

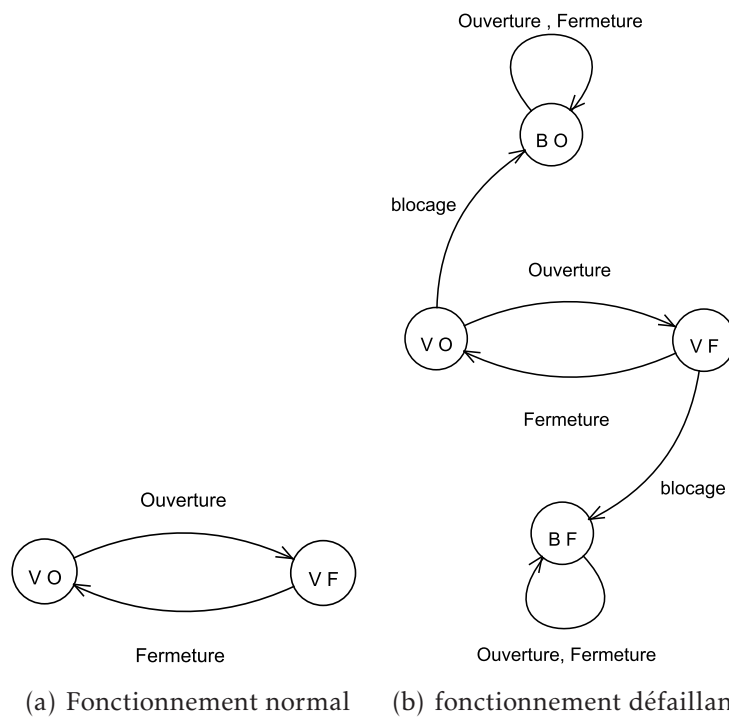


FIG. 5.9 – Modèle événementiel d’une vanne (approche *état*)

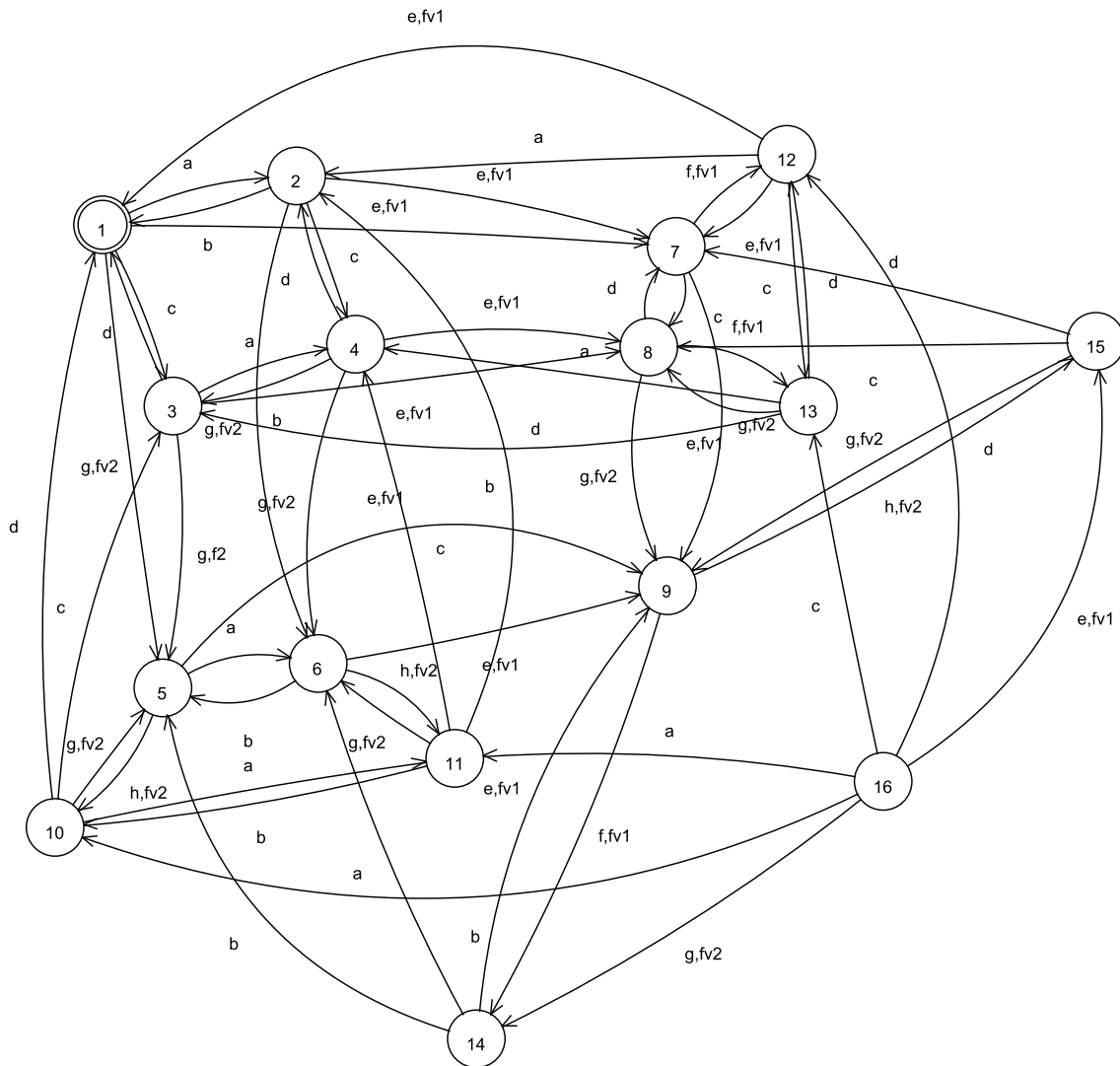


FIG. 5.10 – Représentation du modèle de fonctionnement défaillant du processus à 3 cuves

	a	b	c	d	e	f	g	h	$f_{V_{13}}$	$f_{V_{32}}$
1	2	-	3	-	7	-	5	-	7	5
2	-	1	4	-	7	-	6	-	7	6
3	4	-	-	1	8	-	5	-	8	5
4	-	3	-	2	8	-	6	-	8	6
5	6	-	-	-	9	-	-	10	9	10
6	-	5	-	-	9	-	-	11	9	11
7	-	-	8	-	-	12	9	-	12	9
8	-	-	-	7	-	13	9	-	13	9
9	-	-	-	-	-	14	-	15	14	15
10	-	11	3	1	15	-	5	-	15	5
11	10	-	4	2	15	-	6	-	15	6
12	2	1	13	-	7	-	14	-	7	14
13	4	3	-	12	8	-	14	-	8	14
14	6	5	-	-	9	-	-	16	9	16
15	-	-	8	7	-	16	9	-	16	9
16	11	10	13	12	15	-	14	-	15	14

TAB. 5.4 – Table de transitions du modèle événementiel de fonctionnement défaillant du système à 3 cuves

faillance. Un ensemble de  $k$  machines séquentielles partielles est obtenu pour  $k$  défaillances.

### 5.3.3 Décomposition du modèle événementiel

Nous allons déterminer deux machines séquentielles partielles, la première  $\Sigma_{V_{13}}$  découplée de la défaillance  $f_{V_{13}}$  et la seconde  $\Sigma_{V_{32}}$  découplée de la défaillance  $f_{V_{32}}$ . Les chemins générés par  $\Sigma_{V_{13}}$  sont robustes à la défaillance  $f_{V_{13}}$ . Les chemins générés par  $\Sigma_{V_{32}}$  sont robustes à la défaillance  $f_{V_{32}}$ . En comparant les sorties mesurées sur le processus et les sorties simulées des deux machines séquentielles partielles, nous obtenons une indication sur l'état du système et en analysant les chemins de  $\Sigma_{V_{13}}$  et de  $\Sigma_{V_{32}}$ , nous pouvons déduire l'occurrence ou l'absence des défaillances  $f_{V_{13}}$  et  $f_{V_{32}}$ .

#### Détermination de partitions découplées

Pour déterminer la partition de l'ensemble d'états découplée de  $f_{V_{13}}$  et  $f_{V_{32}}$ , nous allons définir une transition  $i_0$  qui représente l'élément neutre tel que  $\forall s \in S : \delta(s, i_0) = s$ . L'événement  $i_0$  n'induit pas de modification de la trajectoire de fonctionnement. Maintenant nous allons créer une partition de l'ensemble d'entrées qui regroupe  $i_0$  et les événements à découpler.



$$\pi_{I_{f_{V_{13}}}} = \{a, b, c, d, e, f, g, h, \overline{i_0, f_{V_{13}}, f_{V_{32}}}\} \quad (5.11)$$

$$\pi_{I_{f_{V_{32}}}} = \{a, b, c, d, e, f, g, h, \overline{i_0, f_{V_{32}}, f_{V_{13}}}\} \quad (5.12)$$

Ces partitions nous permettent d'obtenir la partition de l'ensemble d'états S par

$$\pi_{f_{V_{13}}} = m_{I-S}(\pi_{I_{f_{V_{13}}}}) = \{\overline{1, 2, 7, 12}, \overline{3, 4, 8, 13}, \overline{5, 6, 9, 14}, \overline{10, 11, 15, 16}\} \quad (5.13)$$

$$\pi_{f_{V_{32}}} = m_{I-S}(\pi_{I_{f_{V_{32}}}}) = \{\overline{1, 3, 5, 10}, \overline{2, 4, 6, 11}, \overline{7, 8, 9, 15}, \overline{12, 13, 14, 16}\} \quad (5.14)$$

Les partitions de l'ensemble d'états  $\pi_{f_{V_{13}}}$  et  $\pi_{f_{V_{32}}}$ , sous réserve qu'elles possèdent la propriété de substitution par rapport au modèle initial, définissent deux machines séquentielles partielles,  $\Sigma_1$  et  $\Sigma_2$ , dont les trajectoires de fonctionnement sont robustes aux influences de  $f_{V_{13}}$  et  $f_{V_{32}}$  respectivement. Rappelons qu'une partition  $\pi$  possède la propriété de substitution si elle remplit la condition suivante

$$(\pi, \pi) \in \Delta_\delta$$

qui est satisfaite si les relations

$$m(\pi) \leq \pi$$

ou

$$M(\pi) \geq \pi$$

sont satisfaites. Ici les opérateurs M et m font référence à  $M_{S-S}$  et  $m_{S-S}$ .

$$m(\pi_{f_{V_{13}}}) = \{\overline{1, 2, 7, 12}, \overline{3, 4, 8, 13}, \overline{5, 6, 9, 14}, \overline{10, 11, 15, 16}\} = \pi_{f_{V_{13}}} \quad (5.15)$$

$$m(\pi_{f_{V_{32}}}) = \{\overline{1, 3, 5, 10}, \overline{2, 4, 6, 11}, \overline{7, 8, 9, 15}, \overline{12, 13, 14, 16}\} = \pi_{f_{V_{32}}} \quad (5.16)$$

$$M(\pi_{f_{V_{13}}}) = \{\overline{1, 2, 7, 12}, \overline{3, 4, 8, 13}, \overline{5, 6, 9, 14}, \overline{10, 11, 15, 16}\} = \pi_{f_{V_{13}}} \quad (5.17)$$

$$M(\pi_{f_{V_{32}}}) = \{\overline{1, 3, 5, 10}, \overline{2, 4, 6, 11}, \overline{7, 8, 9, 15}, \overline{12, 13, 14, 16}\} = \pi_{f_{V_{32}}} \quad (5.18)$$

et par conséquent

$$(\pi_{f_{V_{13}}}, \pi_{f_{V_{13}}}) \in \Delta_\delta \quad \text{et} \quad (\pi_{f_{V_{32}}}, \pi_{f_{V_{32}}}) \in \Delta_\delta$$

Nous remarquons que les deux partitions  $\pi_{f_{V_{13}}}$  et  $\pi_{f_{V_{32}}}$  possèdent la propriété de substitution. Elles sont aptes à générer des machines séquentielles.

Nous allons vérifier maintenant que la partition  $\pi_{f_{V_{13}}}$  ne découple pas de la défaillance  $f_{V_{32}}$  et que  $\pi_{f_{V_{32}}}$  ne découple pas de  $f_{V_{13}}$ . Cette propriété est vérifiée en calculant la partition de l'ensemble d'entrées induite par  $\pi_{f_1}$  et  $\pi_{f_2}$  avec la relation suivante

$$M_{I-S}(\pi_{f_{V_{13}}}) = \{\overline{c}, \overline{d}, \overline{g}, \overline{h}, \overline{i}, \overline{a, b, e, f}, \overline{f_{V_{13}}, f_{V_{32}}}\}$$

Nous constatons que  $\pi_{f_{V_{13}}}$  ne découple pas de  $f_{V_{32}}$ . Par analogie, puisque

$$M_{I-S}(\pi_{f_{V_{32}}}) = \{\overline{c}, \overline{d}, \overline{g}, \overline{h}, \overline{i}, \overline{c}, \overline{d}, \overline{g}, \overline{h}, \overline{f_{V_{32}}}, \overline{f_{V_{13}}}\}$$

alors la partition  $\pi_{f_{V_{32}}}$  ne découple pas de  $f_{V_{13}}$ .

Les deux partitions  $\pi_{f_{V_{13}}}$  et  $\pi_{f_{V_{32}}}$  possèdent les propriétés de couplage et de découplage souhaitées. Par ailleurs, nous remarquons que

$$\pi_{f_{V_{13}}} \cdot \pi_{f_{V_{32}}} = \{\overline{1}, \overline{2}, \overline{3}, \overline{4}, \overline{5}, \overline{6}, \overline{7}, \overline{8}, \overline{9}, \overline{10}, \overline{11}, \overline{12}, \overline{13}, \overline{14}, \overline{15}, \overline{16}\} = \mathbb{O}$$

Cela indique que  $\pi_{f_{V_{13}}}$  et  $\pi_{f_{V_{32}}}$  sont des partitions complémentaires, c'est-à-dire qu'elles permettent de reconstituer la partition initiale.

### 5.3.4 Génération d'indicateurs de défaillances

Dans la section précédente nous avons déterminé les partitions  $\pi_{f_{V_{13}}}$  et  $\pi_{f_{V_{32}}}$  qui possèdent les propriétés de couplage et de découplage souhaitées. Nous allons à présent utiliser ces partitions pour la synthèse de machines séquentielles partielles qui réalisent une partie du fonctionnement du modèle initial. Les deux machines partielles, notées  $\Sigma_{f_{V_{13}}}$  et  $\Sigma_{f_{V_{32}}}$ , sont synthétisées à partir du modèle du système complet et des partitions  $\pi_{f_{V_{13}}}$  et  $\pi_{f_{V_{32}}}$ .

Rappelons qu'une partition de l'espace d'état définit une machine séquentielle partielle en regroupant les états contenus dans un bloc de la partition en un *macro-état*. Les machines obtenues  $\Sigma_{f_{V_{13}}}$  et  $\Sigma_{f_{V_{32}}}$  sont décrites par la table 5.3.4 et la figure (5.11).

	a	b	c	d	e	f	g	h
1''	2''	-	-	-	-	-	3''	-
2''	-	1''	-	-	-	-	3''	-
3''	-	-	-	-	-	-	-	4''
4''	2''	1''	-	-	-	-	3''	-

	a	b	c	d	e	f	g	h
1'	-	-	2'	-	-	-	-	-
2'	-	-	-	1'	-	-	3'	-
3'	-	-	-	-	-	-	-	4'
4'	-	-	2'	1'	-	-	3'	-

TAB. 5.5 – Table de transitions de  $\Sigma_{f_{V_{32}}}$  et de  $\Sigma_{f_{V_{13}}}$

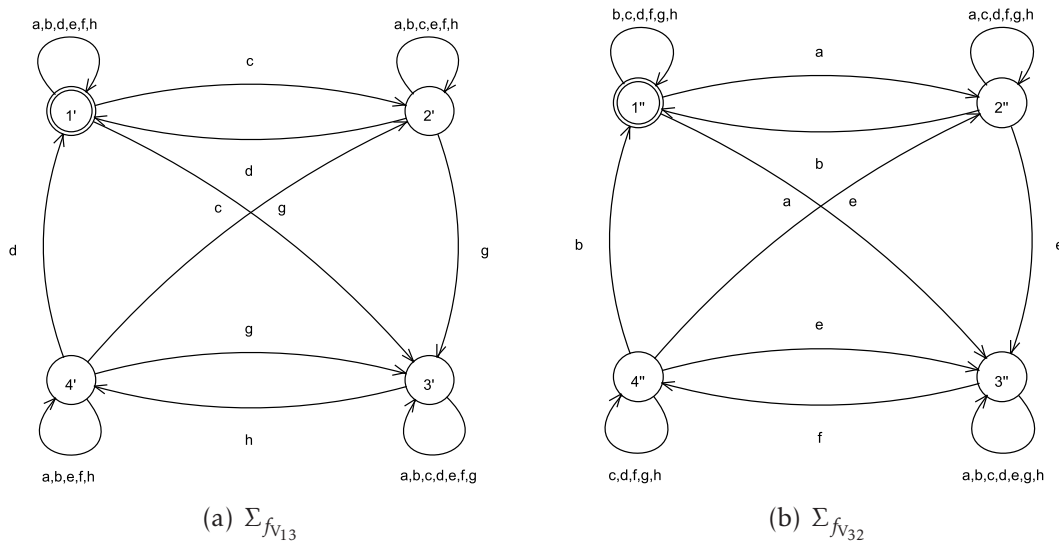


FIG. 5.11 – Représentation graphique de la machine séquentielle partielle  $\Sigma_{f_{V13}}$  et de  $\Sigma_{f_{V32}}$

La table de correspondance entre les états des machines séquentielles partielles  $\Sigma_{f_{V13}}$  et  $\Sigma_{f_{V32}}$  et les partitions de l'ensemble d'états de la machine séquentielle  $\Sigma$  est donnée (table 5.6)

$\Sigma_{f_{V13}}$		$\Sigma_{f_{V32}}$	
Etat	Bloc	Etat	Bloc
1'	1, 2, 7, 12	1''	1, 3, 5, 10
2'	3, 4, 8, 13	2''	2, 4, 6, 11
3'	5, 6, 9, 14	3''	7, 8, 9, 15
4'	10, 11, 15, 16	4''	12, 13, 14, 16

TAB. 5.6 – Correspondance des états et des partitions

Les deux machines séquentielles ainsi définies possèdent les propriétés de couplage et de découplage héritées des partitions  $\pi_{f_{V13}}$  et  $\pi_{f_{V32}}$ . Ceci garantit que les chemins qu'elles génèrent sont sensibles par rapport aux événements auxquels elles sont couplées, et robustes par rapport aux événements desquels elles sont découplées.

Une séquence indicatrice de défaillances est construite à partir du sous-modèle  $\Sigma_{f_{V13}}$ . La sortie de  $\Sigma$  (lorsqu'elle est disponible) est comparée à la sortie de  $\Sigma_{f_{V13}}$  et une séquence  $r_1$  est générée comme suit

$$r_1 = \begin{cases} 1 & \text{si } o_\Sigma \notin o_{\Sigma_{f_{V13}}} \\ 0 & \text{si } o_\Sigma \in o_{\Sigma_{f_{V13}}} \end{cases}$$

où  $o_\Sigma$  est la sortie mesurée et  $o_{\Sigma_{f_{V13}}}$  correspond à un bloc de la partition  $\pi_{f_{V13}}$ . L'indicateur de défaillance  $r_2$  est calculé de manière analogue.

Pour pouvoir comparer les sorties produites avec les sorties mesurées, la table suivante est utilisée

$\Sigma_{f_{V_{13}}}$		$\Sigma_{f_{V_{32}}}$	
E	Sortie	E	Sortie
1'	$V_{32}$ ouverte, flux de $C_3$ vers $C_2$	1''	$V_{13}$ ouverte, flux de $C_1$ vers $C_3$
2'	$V_{32}$ ouverte, flux de $C_2$ vers $C_3$	2''	$V_{13}$ ouverte, flux de $C_3$ vers $C_1$
3'	$V_{32}$ fermée	3''	Vanne 1 fermée
4'	Régime transitoire dans $c_{32}$	4''	Régime transitoire dans $c_{13}$

TAB. 5.7 – Tableau de correspondance entre les états des machines séquentielles partielles et les sorties mesurées

### 5.3.5 Simulations

Cette section est divisée en deux parties :

- Le module de surveillance composé des deux machines séquentielles partielles  $\Sigma_{f_{V_{13}}}$  et  $\Sigma_{f_{V_{32}}}$  est utilisé pour surveiller le modèle événementiel du système à trois cuves (figure 5.10, page 140). Les événements d'entrée sont générés artificiellement. Toutes les simulations et les calculs ont été réalisés à l'aide du logiciel MAXIMA qui est un solveur symbolique. L'objectif de cette simulation est de vérifier que les indicateurs de défaillances générés à partir de  $\Sigma_{f_{V_{13}}}$  et de  $\Sigma_{f_{V_{32}}}$  possèdent les propriétés désirées de robustesse et de sensibilité.
- Le module de surveillance validé dans la première partie est utilisé pour surveiller le modèle temporel du système à 3 cuves décrit à la page 126. Les sorties continues du modèle temporel sont discrétisées et comparées aux sorties des machines séquentielles partielles. Les machines séquentielles partielles sont excitées par des événements obtenus par discrétisation des entrées et des sorties continues. Toute la simulation est réalisée à l'aide du module STATEFLOW du logiciel MATLAB/SIMULINK. L'objectif est de valider l'approche de détection et de localisation de défaillances sur un modèle proche du système réel.

#### Détection de défaillances sur le modèle événementiel

Considérons la séquence d'entrée arbitraire suivante :

$$[1, 3, 2, 9, 4, 5, 3, 6, 2, 7, 1, 8, 4, 10, 2, 5, 7, 8, 4, 6, 2, 1, 3, 2, 4]$$

où les entrées sont représentées par leurs numéros respectifs

$$\left[ a = 1, b = 2, c = 3, d = 4, e = 5, f = 6, g = 7, h = 8, f_{V_{13}} = 9, f_{V_{32}} = 10 \right]$$

Les défaillances apparaissent aux instants 4 et 14 (figure (5.12))

La réponse du système  $\Sigma$  et des machines partielles  $\Sigma_{f_{V_{13}}}$  et  $\Sigma_{f_{V_{32}}}$  est donnée par

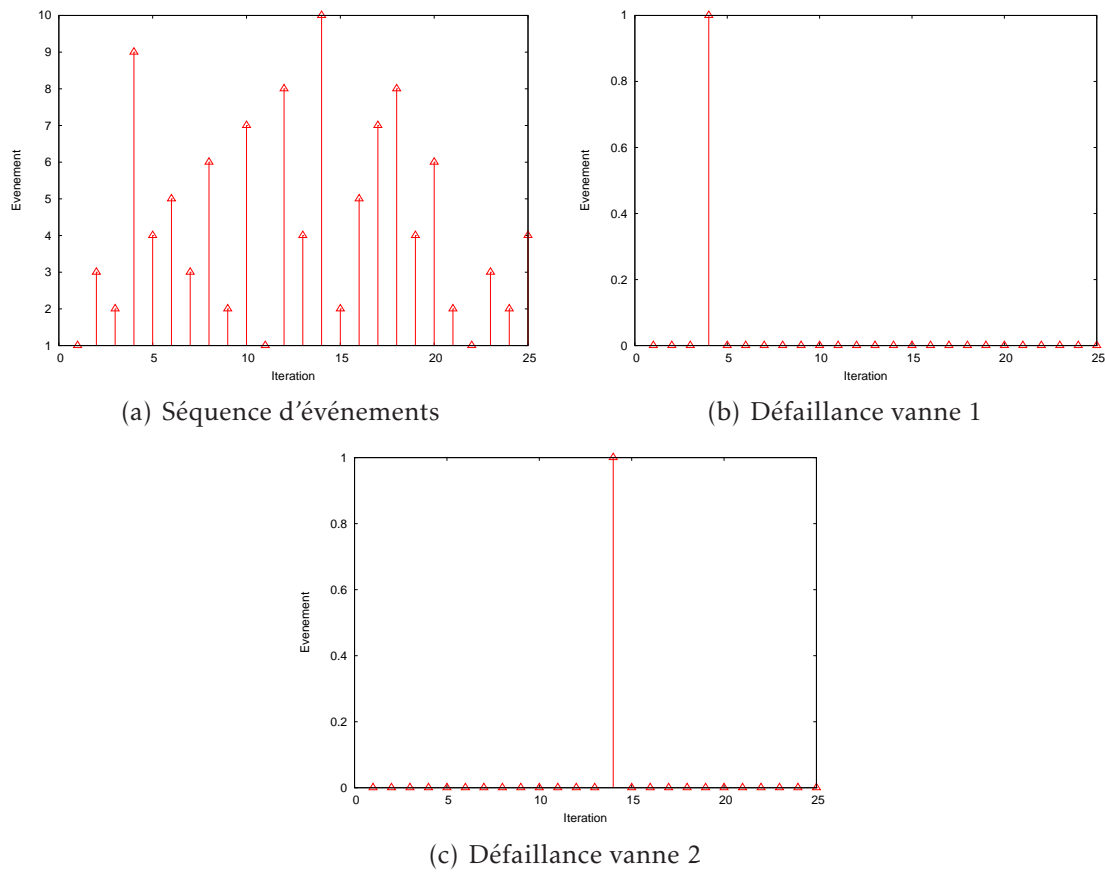


FIG. 5.12 – Séquences d'événements et défaillances

$$\begin{aligned}
\Sigma &: [1, 2, 4, 3, 8, 7, 7, 8, 13, 3, 5, 6, 11, 2, 6, 5, 9, 9, 15, 7, 12, 1, 2, 4, 3, 1] \\
\Sigma_{f_{V_{13}}} &: \left[ \overline{1, 2, 7, 12}, \overline{1, 2, 7, 12}, \overline{3, 4, 8, 13}, \overline{3, 4, 8, 13}, \overline{3, 4, 8, 13}, \overline{1, 2, 7, 12}, \overline{1, 2, 7, 12}, \overline{3, 4, 8, 13}, \dots \right. \\
&\dots \overline{3, 4, 8, 13}, \overline{3, 4, 8, 13}, \overline{5, 6, 9, 14}, \overline{5, 6, 9, 14}, \overline{10, 11, 15, 16}, \overline{1, 2, 7, 12}, \overline{1, 2, 7, 12}, \dots \\
&\dots \overline{1, 2, 7, 12}, \overline{1, 2, 7, 12}, \overline{5, 6, 9, 14}, \overline{10, 11, 15, 16}, \overline{1, 2, 7, 12}, \overline{1, 2, 7, 12}, \overline{1, 2, 7, 12}, \dots \\
&\left. \dots \overline{1, 2, 7, 12}, \overline{3, 4, 8, 13}, \overline{3, 4, 8, 13} \right] \\
\Sigma_{f_{V_{32}}} &: \left[ \overline{1, 3, 5, 10}, \overline{2, 4, 6, 11}, \overline{2, 4, 6, 11}, \overline{1, 3, 5, 10}, \overline{1, 3, 5, 10}, \overline{1, 3, 5, 10}, \overline{7, 8, 9, 15}, \overline{7, 8, 9, 15}, \dots \right. \\
&\dots \overline{12, 13, 14, 16}, \overline{1, 3, 5, 10}, \overline{1, 3, 5, 10}, \overline{2, 4, 6, 11}, \overline{2, 4, 6, 11}, \overline{2, 4, 6, 11}, \overline{2, 4, 6, 11}, \dots \\
&\dots \overline{1, 3, 5, 10}, \overline{7, 8, 9, 15}, \overline{7, 8, 9, 15}, \overline{7, 8, 9, 15}, \overline{7, 8, 9, 15}, \overline{12, 13, 14, 16}, \overline{1, 3, 5, 10}, \dots \\
&\left. \dots \overline{2, 4, 6, 11}, \overline{2, 4, 6, 11}, \overline{1, 3, 5, 10} \right]
\end{aligned}$$

Puisque les deux partitions  $\pi_{f_{V_{13}}}$  et  $\pi_{f_{V_{32}}}$  permettent de recomposer la partition initiale, la combinaison des sorties des deux machines partielles nous permet d'estimer l'état de  $\Sigma$ , en calculant l'intersection des blocs de partitions représentés par les sorties de  $\Sigma_{f_{V_{13}}}$  et de  $\Sigma_{f_{V_{32}}}$

$$\hat{s}_{\Sigma} = s_{\Sigma_{f_{V_{13}}}} \cap s_{\Sigma_{f_{V_{32}}}} : [1, 2, 4, 3, 3, 1, 7, 8, 13, 3, 5, 6, 11, 2, 2, 1, 7, 9, 15, 7, 12, 1, 2, 4, 3]$$

La figure (5.13) illustre les quatre séquences de sorties

Les séquences indicatrices de défauts sont données par la figure (5.14)

### Détection de défaillances sur le modèle temporel

Cette partie est consacrée à la validation de la méthode de détection sur le modèle temporel qui décrit le système à trois cuves. Pour les besoins de la simulation deux nouvelles défaillances sont incorporées dans le modèle de fonctionnement défaillant décrit par l'équation (5.3) :

- $def_{V_{13}}$  est une défaillance qui représente un défaut de la vanne  $V_{13}$ .
- $def_{V_{32}}$  est une défaillance qui représente un défaut de la vanne  $V_{32}$ .

Ces défaillances représentent l'ouverture ou la fermeture intempestive des vannes correspondantes, par exemple, pour la vanne  $V_{13}$ , l'évolution de la défaillance est donnée par :

$$def_{V_{13}} = \begin{cases} -1 & \text{si } v_{13} = 0 \\ v_{13} & \text{si } v_{13} \neq 0 \end{cases} \quad (5.19)$$

Le modèle de fonctionnement défaillant devient

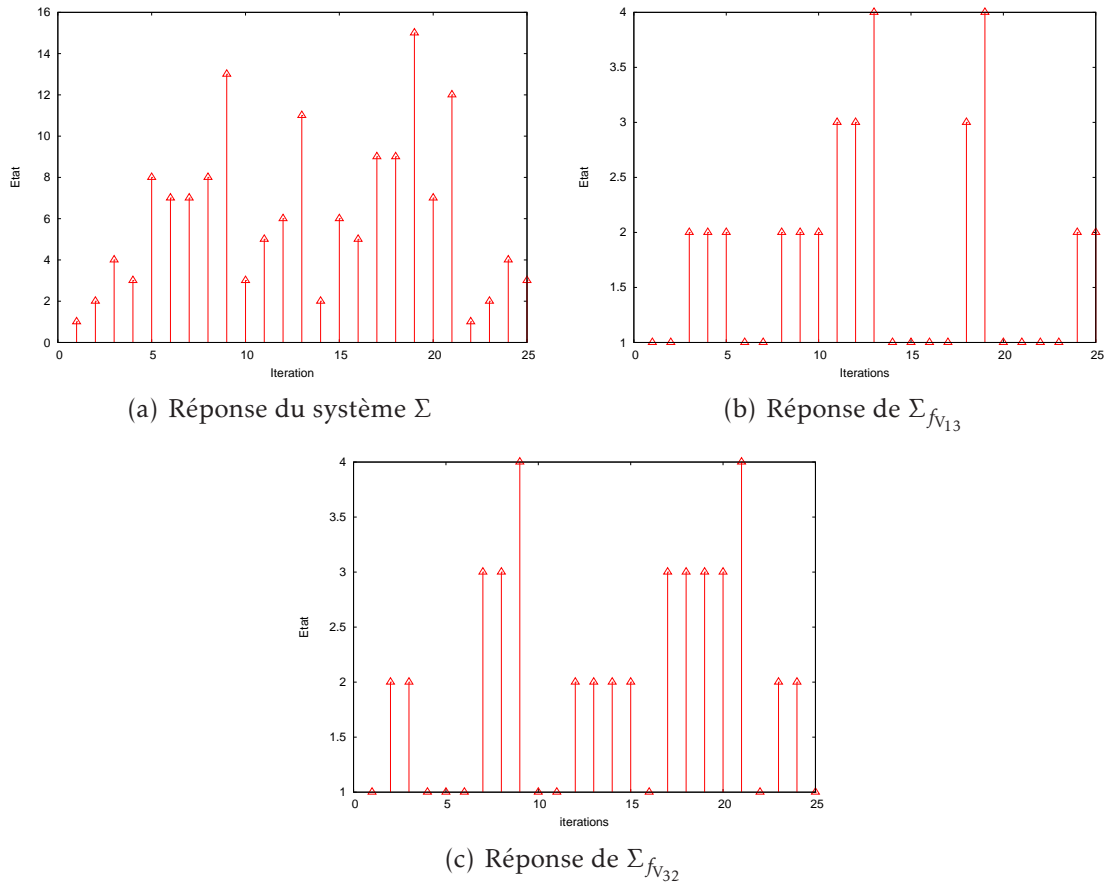


FIG. 5.13 – Graphes représentant les réponses du système complet  $\Sigma$  et des machines partielles  $\Sigma_1$  et  $\Sigma_2$

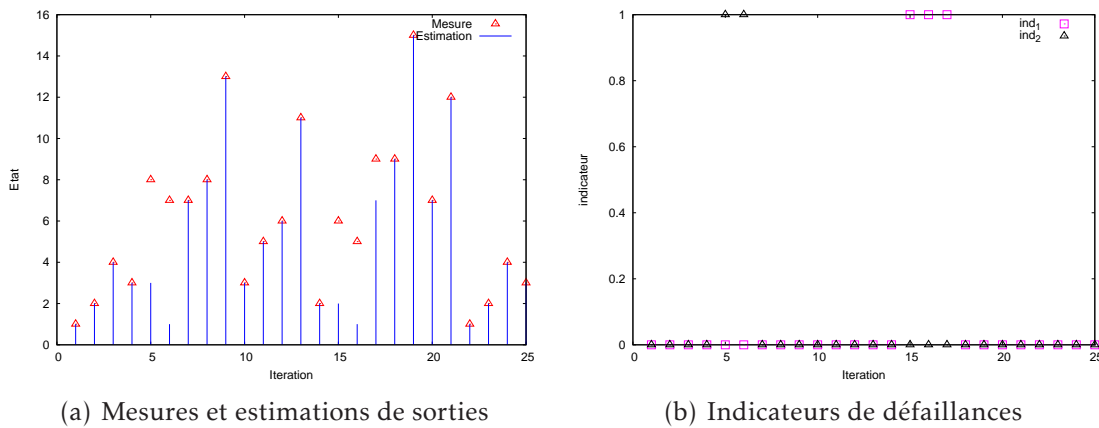


FIG. 5.14 – Graphe représentant l'évolution des indicateurs de défaillances en fonction de l'évolution de  $\Sigma$

$$\left\{ \begin{array}{l} \dot{x}_1 = \frac{1}{S}Kv_1 - K(v_{13} - def_{V_{13}})(1 - b_{13})\frac{S_p}{S} \text{sign}(x_1 - x_3)\sqrt{2g}\sqrt{|x_1 - x_3|} - \frac{f_1}{S}\sqrt{2g}\sqrt{x_1} \\ \dot{x}_2 = \frac{1}{S}Kv_2 + (1 - b_{32})K(v_{32} - def_{V_{32}})\frac{S_p}{S} \text{sign}(x_3 - x_2)\sqrt{2g}\sqrt{|x_3 - x_2|} \cdots \\ \quad \cdots - \frac{f_2}{S}\sqrt{2g}\sqrt{x_2} - Kv_{20}(1 - b_{20})\frac{S_p}{S}\sqrt{2g}\sqrt{x_2} \\ \dot{x}_3 = K(v_{13} - def_{V_{13}})(1 - b_{13})\frac{S_p}{S} \text{sign}(x_1 - x_3)\sqrt{2g}\sqrt{|x_1 - x_3|} \cdots \\ \quad \cdots - (1 - b_{32})K(v_{32} - def_{V_{32}})\frac{S_p}{S} \text{sign}(x_3 - x_2)\sqrt{2g}\sqrt{|x_3 - x_2|} - \frac{f_3}{S}\sqrt{2g}\sqrt{x_3} \end{array} \right.$$

Deux capteurs mesurant les débits  $q_{13}$  et  $q_{32}$  sont ajoutés

$$\left\{ \begin{array}{l} y_1 = q_{13} = K(v_{13} - def_{V_{13}})(1 - b_{13})\frac{S_p}{S} \text{sign}(x_1 - x_3)\sqrt{2g}\sqrt{|x_1 - x_3|} \\ y_2 = q_{32} = K(v_{32} - def_{V_{32}})(1 - b_{32})\frac{S_p}{S} \text{sign}(x_3 - x_2)\sqrt{2g}\sqrt{|x_3 - x_2|} \end{array} \right. \quad (5.20)$$

Les valeurs de ces deux capteurs sont discrétisées, par exemple pour  $y_1$  :

- si  $y_1 < 0$  alors la direction du flux va de la cuve centrale  $C_3$  vers la première cuve  $C_1$ .
- si  $y_1 > 0$  alors la direction du flux va de la première cuve  $C_1$  vers la cuve centrale  $C_3$ .
- si  $y_1 = 0$  alors la vanne  $V_{13}$  est fermée.
- si  $y_1 \neq 0$  alors la vanne  $V_{13}$  est ouverte.

et la même chose pour  $y_2$ . Pendant les premiers instants de l'ouverture d'une vanne, le système est considéré dans un état transitoire non mesuré. L'état transitoire débute au moment où une sortie passe de  $y = 0$  à  $y \neq 0$  et se poursuit tant que la vanne n'est pas complètement ouverte. N'ayant pas accès à une maquette réelle, nous considérons que cette étape transitoire dure une seconde ; ce qui correspond à deux périodes d'échantillonnage durant lesquelles les mesures et les indicateurs de défaillances sont considérés comme non valides.

Suivant la modélisation événementielle proposée dans la section précédente, seize états sont définis suivant la table de correspondance 5.3. Les états de 1 à 9 sont considérés comme valides, ce qui signifie que la génération d'indicateurs de défaillances ne se fait que lorsque le système est dans l'un des neuf premiers états.

La simulation s'effectue en utilisant le logiciel MATLAB/SIMULINK comme pour la simulation du modèle temporel pur, en incorporant des blocs STATEFLOW pour représenter les machines séquentielles partielles découplées et les modules de génération d'indicateurs de défaillances.

Les figures 5.15(a) et 5.15(b) montrent les blocs STATEFLOW qui représentent les machines séquentielles partielles obtenues précédemment (voir la table 5.3.4).

Les simulations sont effectuées sur 300 secondes avec un pas d'échantillonnage de 0.5 seconde (figure 5.16 et 5.17(a) ). Une défaillance de la vanne  $V_{13}$  est introduite à  $t = 100s$  et une défaillance de la vanne  $V_{32}$  est introduite à  $t = 150s$  (figure 5.17(b)).



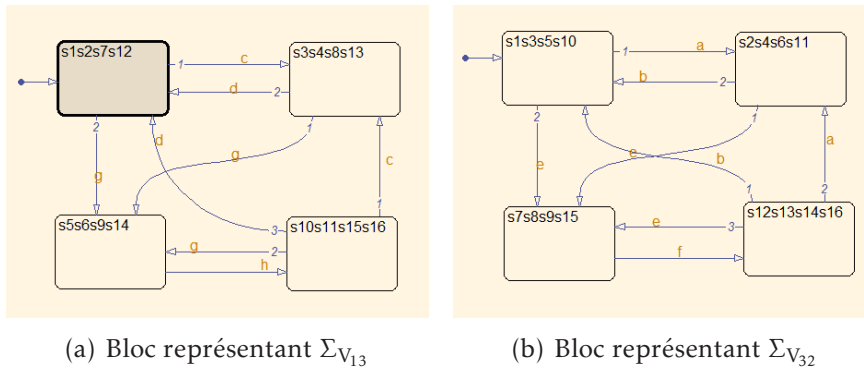


FIG. 5.15 – Blocs représentant les machines séquentielles partielles  $\Sigma_{V_{13}}$  et  $\Sigma_{V_{32}}$

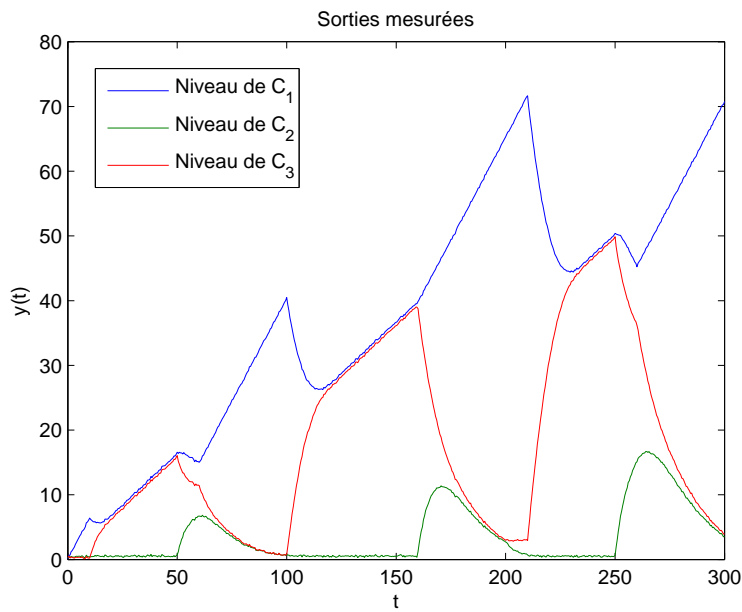


FIG. 5.16 – Évolution des niveaux de liquide dans les trois cuves

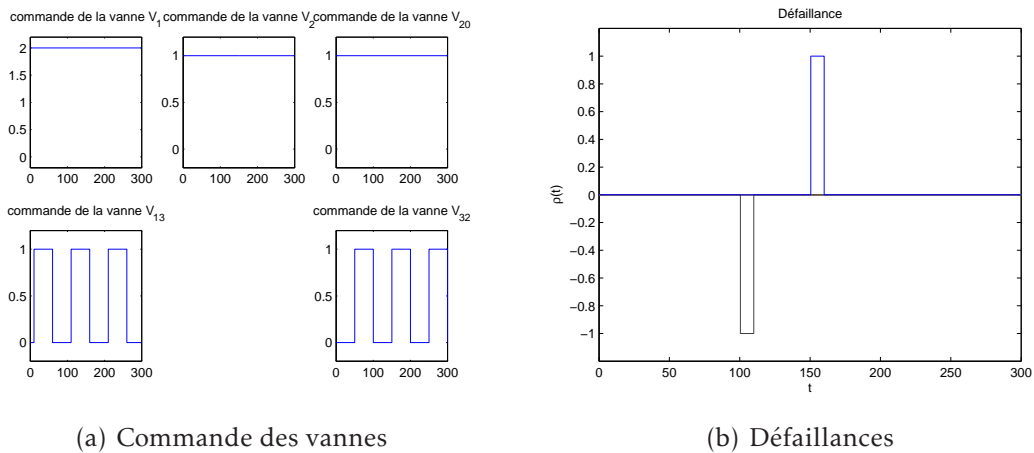


FIG. 5.17 – Entrées et défaillances du modèle continu

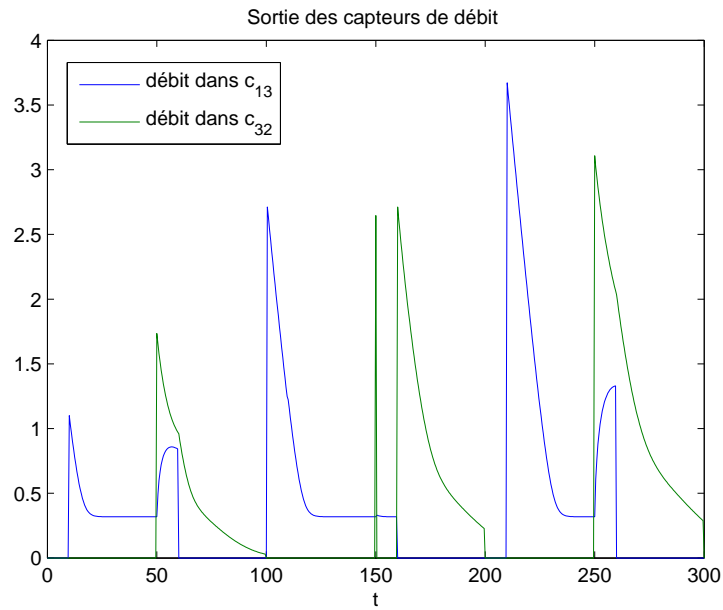


FIG. 5.18 – Évolution des débits de liquide dans les conduites  $c_{13}$  et  $c_{32}$

Les sorties des capteurs de débits sont données figure 5.18.

Les événements obtenus par discrétisation des sorties sont donnés figure 5.19(a) et figure 5.19(b).

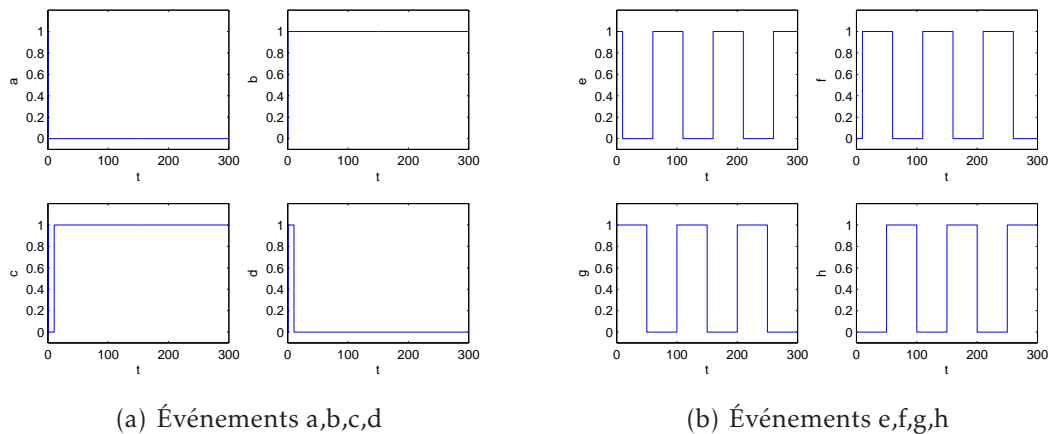


FIG. 5.19 – Événements d'entrée du modèle événementiel

La discrétisation des sorties permet d'obtenir l'évolution *discrète* du système (figure 5.20(a)). L'évolution du modèle événementiel de bon fonctionnement est donné figure 5.20(b). Nous remarquons bien que les deux graphiques ne sont plus similaires aux instants  $t = 100[s]$  et  $t = 150[s]$ , ce qui correspond aux instants d'apparition des défaillances  $f_{V_{13}}$  et  $f_{V_{32}}$ .

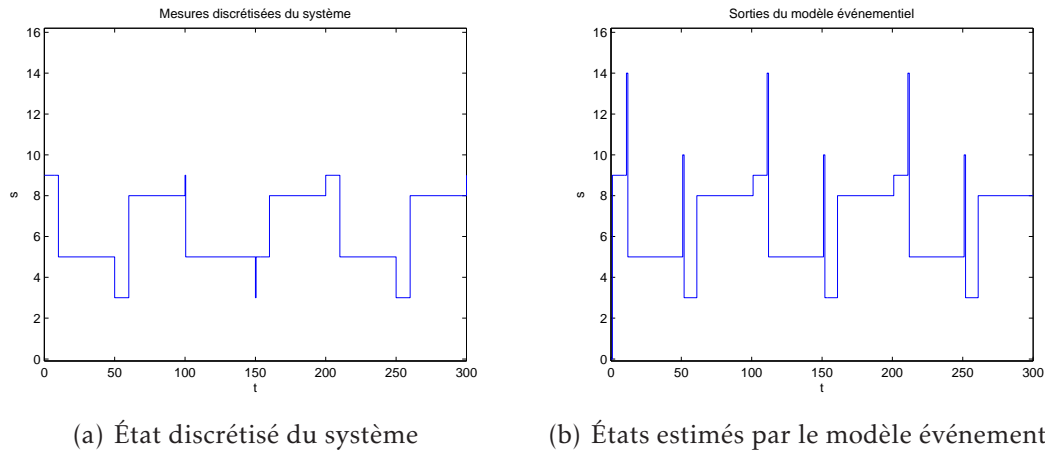


FIG. 5.20 – Évolution discrétisée du système et du modèle

Les sorties des machines séquentielles partielles  $\Sigma_{V_{13}}$  et  $\Sigma_{V_{32}}$  sont montrées figure 5.21(a) et 5.21(b).

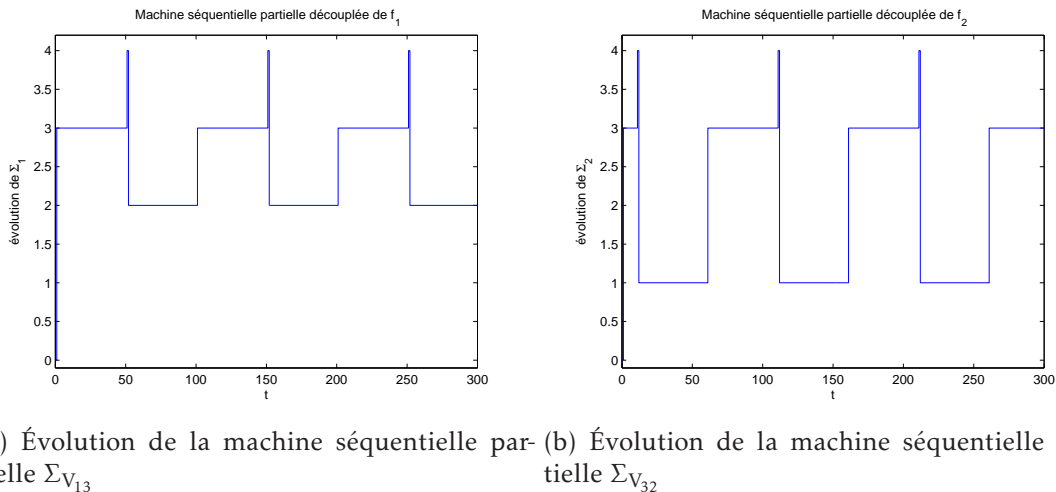
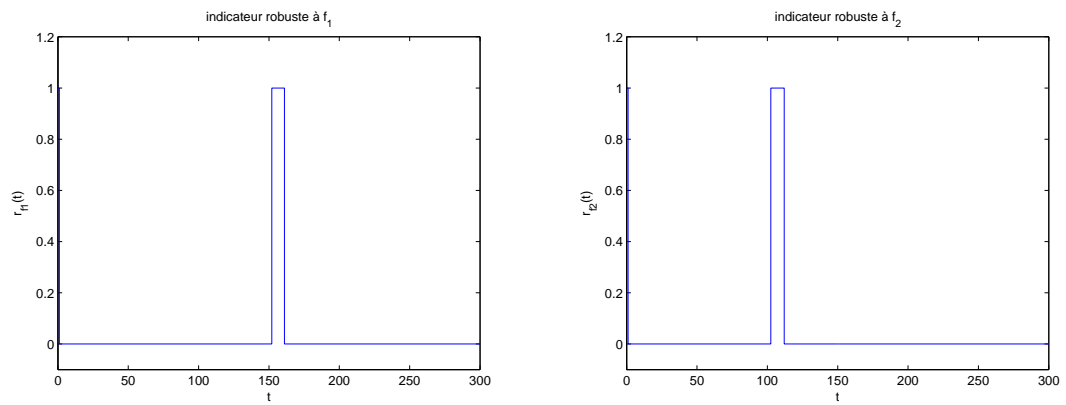


FIG. 5.21 – Évolution des machines séquentielles partielles découplées

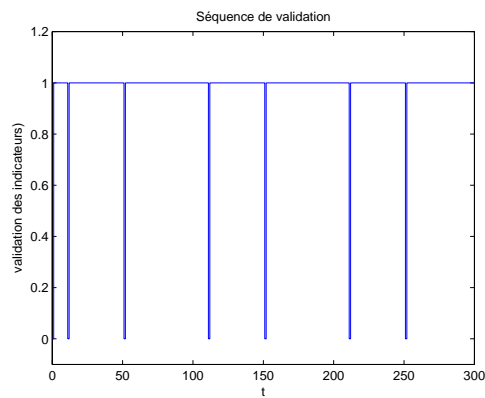
Les indicateurs de défaillances sont obtenus en comparant l'évolution discrétisée du système avec les états des machines séquentielles partielles et en appliquant un filtre pour éliminer les divergences qui durent moins de deux périodes d'échantillonnage (figure 5.22(a) et 5.22(b)). Ces indicateurs sont valides pour les états  $1 \dots 9$  du système, ce qui correspond à la valeur « 1 » de la séquence de validation (figure 5.22(c)).

Nous remarquons que l'indicateur robuste à  $f_{V_{13}}$  réagit seulement à  $f_{V_{32}}$  et que l'indicateur robuste à  $f_{V_{32}}$  réagit seulement à  $f_{V_{13}}$ . La réaction se fait avec un petit retard (1s) dû au traitement. Le module de détection conçu dans la section précédente est donc validé.

**Remarque 5.3.2** *A titre indicatif, les résidus continus sont donnés figure 5.23.*



(a) indicateur de défaillance robuste à  $f_{V13}$       (b) indicateur de défaillance robuste à  $f_{V32}$



(c) indicateur de validité de détection

FIG. 5.22 – Évolution des indicateurs de défaillances

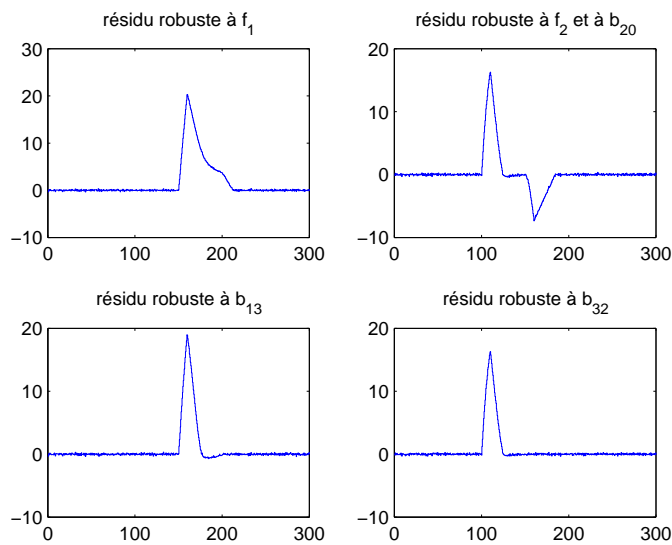


FIG. 5.23 – Résidus continus

*Les deux défaillances actionneurs sont bien détectées. Les résidus ne sont pas conçus pour détecter les deux défaillances  $f_{V_{13}}$  et  $f_{V_{32}}$ , ce qui explique que  $f_{V_{13}}$  et  $f_{V_{32}}$  soient interprétées comme des défaillances dans les conduites  $c_{13}$  et  $c_{32}$ . Remarquons que  $f_{V_{13}}$  est interprétée également comme une fuite dans  $C_1$ , ce qui explique la robustesse du premier résidu par rapport à  $f_{V_{13}}$ .*

## 5.4 Conclusion

Tout au long de ce chapitre, l'exemple classique du processus à trois cuves a été traité :

- Sous la forme temporelle : le modèle est traité avec toutes ses non linéarités non différentiables (fonction « sign »).
- Sous la forme événementielle : L'approche de détection et de localisation d'événements inconnus proposée au chapitre 2 a été illustrée, dans les cas de mesures disponibles en continu ou par intermittence. L'approche est identique au cas temporel, seuls les détails de calcul diffèrent. L'approche proposée a été validée sur un modèle événementiel pur, et sur un modèle temporel discrétisé.

Dans les deux cas, l'utilisation du découplage comme critère de décomposition a permis de réaliser des systèmes de détection et de localisation de défaillances. Cette implémentation a été possible grâce à l'utilisation conjointe d'outils algébriques pour le calcul des indicateurs de défaillances, à savoir l'algèbre des paires et l'algèbre des fonctions.

# Chapitre 6

## Conclusion générale

### 6.1 Contribution

Cette thèse s'inscrit dans la continuité des travaux de l'équipe SFSD<sup>21</sup> du LAGIS<sup>22</sup> sur la détection et la localisation des défauts en utilisant des modèles comportementaux. Le point de départ a été de rechercher une méthode efficace pour la détection et la localisation de défaillances pour les systèmes décrits par un modèle d'évolution non linéaire. La méthode choisie est basée sur la synthèse d'un banc de filtres robustes [DI01, HKE99b, JPS03]. Il est apparu que le traitement séparé des problèmes de découplage et de convergence permet de simplifier le traitement de modèles complexes. La problématique a été formulée pour pouvoir résoudre le problème de décomposition indépendamment, puisque des solutions efficaces, permettant de résoudre le problème de convergence, existent dans la littérature [MH88].

L'algorithme de décomposition a été réalisé en utilisant des outils algébriques pour pouvoir traiter n'importe quel type de non-linéarités présentes dans le modèle, notamment les non-linéarités non différentiables qu'il n'est pas possible de traiter avec les outils de la géométrie différentielle. En étudiant les outils de l'algèbre des fonctions, des algorithmes de calcul d'opérateurs ont été développés. Par ailleurs, il est apparu comme naturel d'étendre l'approche de décomposition pour la localisation de défaillances à d'autres types de modèles comportementaux. Une étude de l'algèbre des paires a été entreprise pour adapter la méthode développée pour les systèmes à dynamique temporelle, au cadre des systèmes à dynamique événementielle représentés par des automates à états finis. Une synthèse des méthodes de surveillance existantes pour les systèmes événementiels a été réalisée, ce qui a permis de souligner des parallèles entre les méthodes de surveillance pour les deux types de systèmes (temporels et événementiels). En utilisant les points communs entre les deux types d'approche, une méthode de localisation de défauts représentés par des événements inconnus a été proposée. Cette méthode est obtenue par extension de la méthode décrite pour les systèmes temporels. L'utilisation des opérateurs de l'algèbre des paires a permis de justifier l'approche de manière théorique, et de la réaliser sous forme algorithmique facile

---

<sup>21</sup>Sûreté de fonctionnement des systèmes dynamiques

<sup>22</sup>Laboratoire d'automatique, génie informatique et signal de Lille

à implémenter. L'utilisation d'un banc de machines séquentielles partielles découplées pour la génération d'indicateurs de défaillances structurés, est à notre connaissance, une nouveauté. L'exemple qui est traité dans le chapitre illustratif a été choisi pour montrer l'efficacité des deux approches, temporelle et événementielle, appliquées au même système représenté de deux manières différentes.

Finalement, la contribution de ce travail est résumée en cinq points principaux :

- Simplifier la formulation du problème de génération de résidus structurés pour la localisation de défauts. Les contraintes sont divisées en deux groupes séparés pour créer deux problèmes indépendants : La robustesse des résidus et leurs convergences.
- Présenter et détailler les notions inhérentes à l'algèbre des paires qui, pour rappel, a été développée pour la synthèse de circuits intégrés. Cette étude permet d'aborder, par la suite, les opérateurs de l'algèbre des fonctions sous un autre angle et de proposer des méthodes de calcul et d'implémentation originales.
- Présenter l'outil « algèbre des fonctions », qui est l'extension de l'algèbre des paires aux ensembles infinis, et l'utiliser pour réaliser la décomposition pour les systèmes à dynamique temporelle. Les propriétés intrinsèques des opérateurs de l'algèbre des fonctions permettent d'obtenir des résultats dans des situations où d'autres techniques mathématiques sont mises en échec. Par ailleurs, des parallèles sont tracés pour rattacher les notions algébriques aux notions géométriques dans le cadre des systèmes continus.
- Proposer des solutions pour améliorer l'algorithme de décomposition utilisé pour la synthèse de sous-systèmes découplés. L'étude de l'approche existante [Zhi97] a permis de cerner les améliorations possibles, et une extension a été réalisée, en utilisant des outils d'élimination de variables pour relaxer les contraintes d'application de la décomposition.
- Étendre la méthodologie de décomposition avec critères de robustesse aux modèles événementiels. Il est possible dès lors de formuler une approche de détection et de localisation d'événements inconnus par décomposition du modèle événementiel pour la génération d'indicateurs de défaillances structurés. Ces notions sont une extension directe de l'approche de surveillance proposée pour les systèmes à dynamique temporelle. Le cadre théorique a été posé pour la réalisation d'un banc de machines séquentielles partielles pour la génération d'indicateurs de défaillances.

## 6.2 Perspectives

De nombreuses perspectives sont apparues tout au long de ce travail, certaines comme des idées abstraites, d'autres comme des notions concrètes, dont le développement a commencé et est actuellement en cours. Les plus significatives sont citées ci-dessous

- Implémenter de manière plus efficace les techniques d'élimination de variables dans l'étape initiale de l'algorithme de décomposition. Actuellement, seule l'injection linéaire de défaillances est considérée.

- Ajouter un degré d'abstraction à la méthode de décomposition pour pouvoir traiter *simultanément* des modèles temporels et événementiels. L'application visée est le découplage de systèmes représentés par des modèles hybrides.
- Proposer une méthode de détection et de localisation de défaillances événementielles en générant un indicateur de défaillances directionnel. L'utilisation de l'algèbre des paires permet de construire un *squelette* de la machine séquentielle qui permet de donner un sens à l'indicateur de différences qui est actuellement considéré comme binaire.
- Utiliser l'algorithme de décomposition en dehors du thème de la surveillance. Par exemple, la décomposition canonique d'un système non linéaire peut être réalisée en utilisant l'algèbre des fonctions, par analogie avec la décomposition canonique de systèmes linéaires proposée dans [Kal62], où l'auteur définit quatre sous-parties en fonction des propriétés de commandabilité et d'observabilité.





# Annexe A

## Rappels sur les propriétés des machines séquentielles

### A.1 Introduction

Cette annexe rappelle des notions liées à la décomposition de machines séquentielles, et à l'utilisation de l'algèbre des paires sur les machines séquentielles. Nous y aborderons les notions d'équivalence de machines séquentielles, de transformations entre deux machines et de la construction d'un réseau de machines. A chaque fois les outils de l'algèbre des paires sont utilisés

### A.2 Rappels sur les machines séquentielles

Nous utilisons dans ce qui suit les définitions (2.9,2.10) pour les machines de Mealy, de Moore et d'état. Les notions d'isomorphisme et d'homomorphisme sont abordées, ainsi que des notions plus générales d'équivalence entre machines.

La notion d'isomorphisme permet d'identifier deux machines séquentielles qui ont la même dynamique, mais qui diffèrent par leurs ensembles d'entrées, d'états et de sorties.

**Définition A.1** Soit deux machines séquentielles du même type définies par

$$\Sigma(S, I, O, \delta, \lambda) \text{ et } \Sigma' = (S', I', O', \delta', \lambda')$$

Les deux machines sont isomorphes si et seulement si il existe trois bijections

$$\begin{cases} f_1: S \rightarrow S' \\ f_2: I \rightarrow I' \\ f_3: O \rightarrow O' \end{cases} \quad (\text{A.1})$$

qui réalisent

$$\begin{aligned} f_1[\delta(s, x)] &= \delta'[f_1(s), f_2(x)] \\ f_1[\lambda(s, x)] &= \lambda'[f_1(s), f_2(x)] \end{aligned}$$

Le triplet  $(f_1, f_2, f_3)$  est appelé isomorphisme entre les machines  $\Sigma$  et  $\Sigma'$ .

Si les applications  $f_1, f_2, f_3$  ne sont pas des bijections mais des surjections, alors  $\Sigma'$  est une image homomorphe de  $\Sigma$ . On se réfère alors au triplet  $(f_1, f_2, f_3)$  en tant qu'homomorphisme de la machine  $M$  vers  $M'$ .

Nous pouvons considérer l'isomorphisme comme une *imitation* de fonctionnement. Dès lors, deux machines séquentielles isomorphes sont interchangeable *via* une substitution de leurs ensembles de définitions. Cette substitution (ou transformation équivalente) n'entraîne pas de perte d'informations (figure A.1).

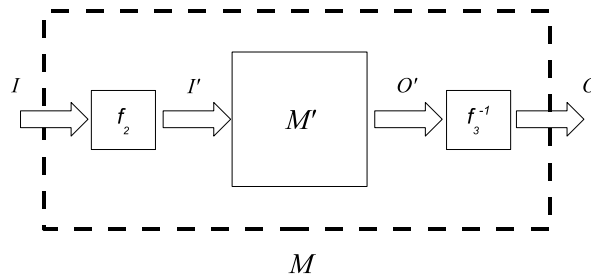


FIG. A.1 – Représentation schématique des machines séquentielles isomorphes  $\Sigma'$  et  $\Sigma$

La notion d'homomorphisme quant à elle est unilatérale. La machine  $\Sigma$  peut simuler le fonctionnement de son image homomorphe  $\Sigma'$ , mais l'inverse n'est pas possible. On dit alors que  $\Sigma$  est plus riche que  $\Sigma'$  et peut en reproduire le fonctionnement. La transformation de  $\Sigma$  à  $\Sigma'$  entraîne une perte d'information (figure A.2).

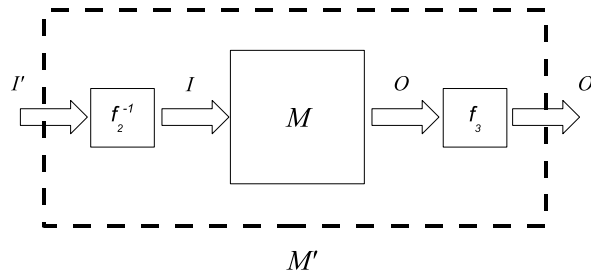


FIG. A.2 – Représentation schématique  $\Sigma$  et de son image homomorphe  $\Sigma'$

Les notions d'isomorphisme et d'homomorphisme nous mènent directement à la notion d'équivalence de machines séquentielles. Deux machines séquentielles sont dites équivalentes si l'une des deux conditions est remplie :

- Toutes les séquences d'entrées produisent des sorties identiques pour les deux machines.
- Pour tout état de la première machine, il existe un état équivalent dans la seconde.

**Définition A.2** Soit deux machines  $\Sigma_1 = (S_1, I, O, \delta_1, \lambda_1)$  et  $\Sigma_2 = (S_2, I, O, \delta_2, \lambda_2)$  avec les mêmes ensembles d'entrées et de sorties. Les états  $s_1 \in S_1$  et  $s_2 \in S_2$  sont dits équivalents si et seulement si pour toute séquence de commandes  $\bar{x}$

$$\bar{\lambda}_1(s_1, \bar{x}) = \bar{\lambda}_2(s_2, \bar{x})$$

**Définition A.3** Deux machines de même type  $\Sigma_1$  et  $\Sigma_2$  sont équivalentes si et seulement si tout état  $s_1 \in S_1$  possède un état équivalent  $s_2 \in S_2$  et réciproquement.

**Illustration** La figure A.3 illustre la relation d'équivalence : Les machines  $\Sigma_1$  (Moore) et  $\Sigma_2$  (Mealy) sont équivalentes au sens de (A.3) et les deux sont équivalentes à  $\Sigma'$  au sens de la définition (A.2)

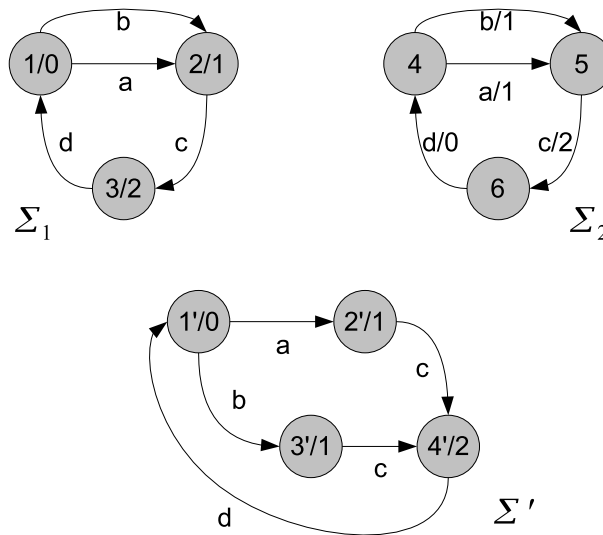


FIG. A.3 – Schéma représentant les machines équivalentes  $\Sigma_1$ ,  $\Sigma_2$  et  $\Sigma'$

Puisqu'il existe un certain nombre d'équivalences pour une machine  $\Sigma$ , définissons la plus petite machine équivalente (en nombre d'états) comme une réduction de  $\Sigma$ .

**Définition A.4** Une machine séquentielle  $\Sigma(S, O, I, \delta, \lambda)$  est dite réduite si et seulement si aucun de ses états ne possède d'état équivalent autre que lui-même :

$$\forall s_1 \in S : s_1 \equiv s_2 \Rightarrow s_1 = s_2$$

Le résultat suivant [HS66] est important dans ce contexte, puisqu'il stipule que pour toute machine  $\Sigma$  il existe un homomorphisme qui permet de déterminer la machine réduite équivalente, et que cette machine est unique :

**THÉORÈME A.1** Considérons la machine séquentielle  $\Sigma(S, O, I, \delta, \lambda)$  et sa machine équivalente réduite  $\Sigma'$ . Pour toute autre machine  $\Sigma_i \equiv \Sigma$  il existe un homomorphisme de  $\Sigma_i$  à  $\Sigma$

**Corollaire A.1.1** Si deux machines réduites  $M$  et  $M'$  sont équivalentes, alors elles sont forcément isomorphes.

**Illustration** Dans l'exemple précédent, la machine  $\Sigma_1$  est la machine réduite équivalente de  $\Sigma'$ .

Ces notions sont suffisantes pour permettre définir le cadre théorique de la méthode de décomposition de machines séquentielles présentée dans le chapitre 4.



# Annexe B

## Rappels sur la géométrie différentielle

### B.1 Introduction

Cette partie a pour objectif de rappeler les notions les plus courantes de la géométrie différentielle : le difféomorphisme, l'espace tangent, les champs de vecteurs. Des outils classiques de l'algèbre de Lie sont rappelés également notamment la dérivée et le crochet de Lie.

### B.2 Variété et difféomorphisme

Rappelons d'abord le théorème des fonctions implicites

**THÉORÈME B.1** *dit des fonctions implicites* : Soit  $\phi$  une application  $k$  fois continûment différentiable, avec  $k \geq 1$ , d'un ouvert  $U \subset \mathbb{R}^n$  vers  $\mathbb{R}^{n-p}$  avec  $0 \leq p < n$ , et supposons qu'il existe au moins un  $x_0 \in U$  tel que  $\phi(x_0) = 0$ . Si en chaque point  $x \in U$  l'application linéaire tangente  $D\phi(x)$  est de rang plein  $n - p$ , il existe un voisinage  $V = V_1 \times V_2 \subset U$  de  $\mathbb{R}^n$  et une application  $k$  fois continûment différentiable  $\varphi$  de  $V_1$  dans  $V_2$  tels que les deux ensembles  $\{x \in V_1 \times V_2 \mid \phi(x) = 0\}$  et  $\{(x_1, x_2) \in V_1 \times V_2 \mid x_2 = \varphi(x_1)\}$  sont égaux.

Autrement dit, la fonction  $\varphi$  vérifie localement  $\phi(x_1, \varphi(x_1)) = 0$  et la variable dépendante  $x_2 = \varphi(x_1)$  est décrite à l'aide des  $p$  variables dépendantes localement  $x_1$ .

**Définition B.1** *Etant donné une application différentiable  $\Phi$  de  $\mathbb{R}^n$  dans  $\mathbb{R}^{n-p}$  avec  $0 \leq p < n$ , on suppose qu'il existe au moins un  $x_0$  solution de l'équation implicite  $\Phi(x) = 0$  et que l'application linéaire tangente  $D\Phi(x)$  est de rang plein  $(n - p)$  dans un voisinage  $V$  de  $x_0$ . On appelle variété différentiable de dimension  $p$  l'ensemble  $X$  défini par l'équation implicite  $\Phi(x) = 0$*

$$X = \{x \in V \mid \Phi(x) = 0\}$$

*Cet ensemble n'est pas vide du fait du théorème B.1. Si  $\Phi$  est de surcroît  $k$  fois différentiable, alors  $X$  est une variété de classe  $C^k$ , avec  $k = 0, \dots, \infty$*

Le difféomorphisme local permet de décrire les variétés de manière intrinsèque, c'est-à-dire indépendamment de la base dans laquelle  $\Phi(X) = 0$  est écrite.

**Définition B.2** Soit une application  $\phi$  d'un ouvert  $U \subset \mathbb{R}^n$  de classe  $C^k$ , avec  $k \geq 1$ . On dit que  $\phi$  est un difféomorphisme local de classe  $C^k$  dans un voisinage  $U(x_0)$  d'un point  $x_0$  de  $U$  si  $\phi$  est inversible de  $U(x_0)$  dans un voisinage  $V(\phi(x_0))$  d'un point  $\phi(x_0)$  de  $V$  et si  $\phi^{-1}$  est aussi de classe  $C^k$ .

**Définition B.3** On dit que  $\phi$  est un homéomorphisme local de classe  $C^k$  si  $\phi$  est de classe  $C^k$ , localement inversible et son inverse est continue.

Les difféomorphismes locaux sont caractérisés par les deux théorèmes suivants :

**THÉORÈME B.2 dit d'inversion locale :**  $\phi$  est un difféomorphisme local de classe  $C^k$  avec ( $k \geq 1$ ) au voisinage de  $x_0$  si et seulement si son application linéaire tangente  $D\phi(x_0)$  est bijective

**THÉORÈME B.3 dit de rang constant :** Soit  $\phi$  une application de classe  $C^k$  avec ( $k \geq 1$ ) d'une variété  $X$  de dimension  $n$  de classe  $C^k$  dans une variété  $Y$  de dimension  $p$  de classe  $C^k$ . Si  $\text{rang}(D\phi(x)) = q \leq \min(n, p)$  pour tout  $x$  dans un voisinage  $U$  du point  $x_0 \in X$ , alors :

1.  $\forall y \in \phi(U) \subset Y$ ,  $\phi^{-1}(\{y\})$  est une sous-variété de  $X$  de dimension  $n - q$  et de classe  $C^k$ .
2.  $\phi(U)$  est une sous-variété de  $Y$  de dimension  $q$  et de classe  $C^k$ . En particulier
  - si  $n \leq p$ ,  $\phi$  est injective de  $U$  sur  $Y$  si et seulement si  $\text{rang}(D\phi(x)) = n$  pour tout  $x \in U$ . En en déduit que  $\phi$  est un homéomorphisme de  $U$  sur  $\phi(U)$ .
  - si  $n \geq p$ ,  $\phi$  est surjective de  $U$  sur  $V$  ouvert sur  $Y$  si et seulement si  $\text{rang}(D\phi(x)) = p$ .

L'interprétation la plus simple du difféomorphisme est géométrique : les coordonnées curvilignes qui permettent d'exprimer une variété  $X$  dans un sous-espace vectoriel de  $\mathbb{R}^p$  sont données par

$$y_1 = \Phi(x), \dots, y_{n-p} = \Phi_{n-p}, y_{n-p+1} = \Psi_1(x), \dots, y_n = \Psi_p(x)$$

où les différents  $\Psi_i$  sont choisies pour que l'application

$$x \mapsto (\Phi_1(x), \dots, \Phi_{n-p}, \Psi_1(x), \dots, \Psi_p(x))$$

forment un difféomorphisme local. On dit alors qu'on a redressé  $X$  puisque

$$X = \{y \mid y_1 = y_2 = \dots = y_{n-p} = 0\}$$

**Définition B.4** L'espace tangent à  $X$  au point  $x \in X$  est l'espace vectoriel

$$T_x X = \ker D\Phi(x)$$

Le fibré tangent  $TX$  est l'ensemble  $TX = \bigcup_{x \in X} T_x X$

Notons que puisque  $D\Phi(x)$  est de rang  $n - p$  dans l'ouvert  $V$  alors

$$\dim T_x X = \dim \ker D\Phi(x) = p, \forall x \in V$$

**Définition B.5** Un champ de vecteurs  $f$  sur  $X$  est une application qui fait correspondre à tout  $x \in X$  le vecteur  $f(x) \in T_x X$

**Définition B.6** Une courbe intégrale du champ de vecteurs  $f$  est une solution locale de l'équation différentielle  $\dot{x} = f(x)$

Les courbes intégrales de  $f$  existent et sont uniques si  $f$  est de classe  $C^k$  (c'est-à-dire localement lipschitzienne). Rappelons qu'une fonction  $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$  est localement lipschitzienne si et seulement si

$$\forall U \subset \mathbb{R}^n, \forall x_1, x_2 \in U, \exists K : \|f(x_1) - f(x_2)\| \leq K \|x_1 - x_2\|$$

L'équation différentielle  $\dot{x} = f(x)$  avec  $f$  localement lipschitzienne admet au voisinage de tout point  $x_0$  une courbe intégrale passant par  $x_0$  pour  $t = 0$ .

## B.3 Algèbre de Lie

Considérons maintenant un système de coordonnées locales  $(x_1, \dots, x_n)$ . Les composantes du champ de vecteurs  $f$  dans ces coordonnées sont notées  $[f_1 | f_2 | \dots | f_n]^T$ . On associe à  $f$  une opération de dérivation appelée *dérivée de Lie dans la direction  $f$* . On note  $t \mapsto X_t(x)$  la courbe intégrale de  $f$  qui passe par  $x$  à  $t = 0$ .

**Définition B.7** Soit  $h : \mathbb{R}^n \rightarrow \mathbb{R}^n$  une fonction de classe  $C^1$ . On appelle *dérivée de Lie de  $h$  dans la direction  $f$* , notée  $L_f h$ , la dérivée de  $h$  le long de la courbe intégrale de  $f$  en  $t = 0$

$$L_f h(x) = \frac{d}{dt} h(X_t(x))|_{t=0} = \sum_{i=1}^n f_i(x) \frac{\partial h}{\partial x_i}(x)$$

Cette formule identifie un champ de vecteurs  $f$  à l'opérateur différentiel linéaire

$$L_f = \sum_{i=1}^n f_i(x) \frac{\partial}{\partial x_i}$$

Considérons à présent un champ de vecteurs  $f$  et une fonction  $h$ . L'opération  $L_f$  peut être itérée : On définit un opérateur différentiel linéaire  $L_f^k h$  pour tout  $k \geq 0$  tel que

$$L_f^0 h = h \text{ et } L_f^k h = L_f(L_f^{k-1} h), \forall k \geq 1$$

En outre, si  $g_1, \dots, g_k$  sont  $k$  champs de vecteurs sur  $X$ , la dérivée de Lie itérée d'ordre  $r_1 + \dots + r_k$  est définie par

$$L_{r_1}^{g_1} \dots L_{r_1}^{g_1} h = L_{r_1}^{g_1} (L_{r_2}^{g_2} \dots L_{r_k}^{g_k})$$

Un champ de vecteurs peut être redressé par un changement de coordonnées. Pour le faire, nous définissons deux notions, l'image d'un difféomorphisme et l'intégrale première d'un champ de vecteurs.



**Définition B.8** L'image par le difféomorphisme  $\varphi$  du champ  $f$ , noté  $\varphi * f$  est le champ de vecteurs donné par

$$\varphi * f = \left[ L_f \varphi_1(\varphi^{-1}(y)), \dots, L_f \varphi_n(\varphi^{-1}(y)) \right]^T$$

On dit que  $f$  est redressée par difféomorphisme :

$$L_f \varphi_1(\varphi^{-1}) = 1 \quad (\text{B.1})$$

$$L_f \varphi_i(\varphi^{-1}) = 0 \quad \forall i = 2, \dots, n \quad (\text{B.2})$$

On en déduit que la dérivée de Lie n'est pas affectée par le changement de coordonnées puisque  $L_f h(x) = L_{\varphi * f}(h \circ \varphi^{-1})(y)$ .

**Définition B.9** Une intégrale première de  $f$  est une fonction  $\gamma$  constante le long des courbes intégrales de  $f$ , c'est-à-dire qui vérifie  $L_f \gamma = 0$ .

Nous pouvons maintenant définir la notion de crochet de Lie entre deux champs de vecteurs distincts.

**Définition B.10** Le crochet de Lie entre deux champs de vecteurs  $f$  et  $g$  est le champ de vecteurs noté  $[f, g]$  déterminé par

$$L_{[f, g]} = L_f L_g - L_g L_f$$

Ce qui donne en coordonnées locales

$$[f, g] = \sum_{i=1}^n \left( \sum_{j=1}^n (f_j \frac{\partial g_i}{\partial x_j} - g_j \frac{\partial f_i}{\partial x_j}) \right) e_i$$

Le crochet de Lie possède les propriétés suivantes :

- $[f, g] = -[g, f]$  (antisymétrie).
- $[\alpha f, \beta g] = \alpha \beta [f, g] + (\alpha L_f \beta) g - (\beta L_g \alpha) f$ , pour toute paire de fonction  $C^\infty$   $(\alpha, \beta)$
- $[f_1, [f_2, f_3]] + [f_2, [f_3, f_1]] + [f_3, [f_1, f_2]] = 0$  (identité de Jacobi)

Le crochet de Lie s'interprète comme suit : considérons le sous-espace vectoriel  $E(x)$  de  $T_x X$  engendré par  $f(x)$  et  $g(x)$  pour tout  $x$  dans un voisinage donné, le crochet  $[f, g]$  indique si les courbes intégrales de  $f$  et  $g$  ont tendance à rester tangentes à  $E$  pour un  $t$  suffisamment petit ( $[f, g] \in E$ ) ou non ( $[f, g] \notin E$ ).

## B.4 Distribution de champs de vecteurs et intégrabilité

**Définition B.11** Une distribution de champs de vecteurs  $\mathcal{D}$  est une application qui fait correspondre le sous-espace vectoriel  $\mathcal{D}(x) \subset T_x X$  à tout point  $x \in X$

Considérons un ouvert  $V$  de  $X$ . La distribution  $\mathcal{D}$  est régulière et de rang  $k$  constant dans  $V$  s'il existe des champs de vecteurs réguliers  $g_1, \dots, g_k$  tels que :

- $\forall x \in V : \text{rang}(g_1(x), \dots, g_k(x)) = k$

–  $\mathcal{D}(x)$  décrit un espace vectoriel  $\{(g_1(x), \dots, g_k(x))\}$  pour tout  $x \in V$ . On écrit aussi

$$\mathcal{D}(x) = \text{e.v.} \{g_1(x), \dots, g_k(x)\}$$

Notons qu'une distribution de champs de vecteurs ne dépend pas des coordonnées choisies. Nous allons définir maintenant une propriété essentielle des distributions :

**Définition B.12** *La distribution  $\mathcal{D}$  est dite involutive si et seulement si pour tout couple de champs de vecteurs  $f$  et  $g$  de  $\mathcal{D}$  on a  $[f, g] \in \mathcal{D}$*

Une distribution involutive est donc caractérisée par  $[\mathcal{D}, \mathcal{D}] \subset \mathcal{D}$ . Par ailleurs, si  $\mathcal{D}$  n'est pas involutive, on définit sa clôture involutive par :

**Définition B.13** *La clôture involutive  $\bar{\mathcal{D}}$  d'une distribution  $\mathcal{D}$  est la plus petite distribution involutive contenant  $\mathcal{D}$ .*

Notons que  $\bar{\mathcal{D}}$  peut être construite à partir des crochets itérés des champs  $g_1, \dots, g_k$ .

Nous avons fini de définir les outils de l'algèbre de Lie. Nous pouvons à présent nous poser la question suivante : Est ce que l'on peut, pour une distribution donnée, trouver un difféomorphisme qui la redresse ? Le théorème suivant, dit **théorème de Frobenius**, donne la réponse :

**THÉORÈME B.4** *Soit une distribution  $\mathcal{D}$  régulière de rang  $k$  constant. La distribution  $\mathcal{D}$  peut être redressée par difféomorphisme si et seulement si  $\mathcal{D}$  est involutive. Dans ce cas, en tout point  $x \in U$  ouvert dense sur  $X$ , il existe une sous-variété, appelée variété intégrale, de dimension  $k$ , tangente en tout point à  $\mathcal{D}$ .*

## B.5 Équations aux dérivées partielles du premier ordre

Formulons maintenant le problème de résolution d'un ensemble d'équation différentielles partielles du premier ordre dans un ouvert  $V$  de la variété  $X$ .

$$\begin{cases} L_{g_1} y = 0 \\ \vdots \\ L_{g_k} y = 0 \end{cases}$$

où  $y$  est la fonction inconnue et où  $\{g_1, \dots, g_k\}$  sont des champs de vecteurs réguliers de rang  $k$  pour tout  $x \in V$ .

Soit  $\mathcal{D}(x) = \text{e.v.} \{(g_1(x), \dots, g_k(x))\}$ ,  $\forall x \in V$ . Si  $y$  est une solution du système alors  $L_g y = 0$  pour tout  $g \in \bar{\mathcal{D}}$ , puisque  $L_{[g_i, g_j]} = 0$ . Selon le théorème de Frobenius, pour peu que  $\mathcal{D}$  soit involutive, elle peut être redressée dans  $V$  par un difféomorphisme  $\xi = \varphi(x)$  et l'image de  $\mathcal{D}$  par  $\varphi$  dans  $\varphi(V)$  est donnée par

$$\varphi * \mathcal{D}(\xi) = \text{e.v.} \{\varphi * g_1(\xi), \dots, \varphi * g_k(\xi)\} = \text{e.v.} \left\{ \frac{\partial}{\partial \xi_1}, \dots, \frac{\partial}{\partial \xi_k} \right\}$$

Maintenant, posons  $z = y \circ \varphi^{-1}$ . Dans ce cas

$$L_g y = 0 \Leftrightarrow \frac{\partial z}{\partial \xi_1} = 0, \dots, \frac{\partial z}{\partial \xi_k} = 0$$

par conséquent  $z$  ne dépend pas des différents  $\xi \in \varphi(V)$ . La solution de l'équation différentielle est donnée par  $y = z \circ \varphi$ , si nous connaissons  $z$ .

Nous avons montré que la résolution d'une équation différentielle du premier ordre peut être formulée comme un redressement du champ de vecteurs correspondant. Par contre, il est aisé de voir que cette solution ne peut être obtenue que sous l'hypothèse de la connaissance préalable de la base  $\left\{ \frac{\partial}{\partial \xi_1}, \dots, \frac{\partial}{\partial \xi_k} \right\}$  qui permet de redresser  $\{g_1, \dots, g_k\}$ .

Par ailleurs si  $\mathcal{D}$  n'est pas involutive, sa clôture involutive  $\bar{\mathcal{D}}$  peut toujours être redressée si elle est de rang constant dans  $V$ .

## B.6 Formes différentielles et intégrabilité

Considérons la distribution  $TX$  qui à tout point de la variété  $X$  fait correspondre l'espace tangent  $T_x X$ .  $TX$  est involutive et possède une base locale  $\left\{ \frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_k} \right\}$ .

L'espace dual de  $T_x X$  est noté  $T_x^* X$  et est appelé *espace cotangent* à la variété  $X$  au point  $x$  et possède une base  $\{dx_1, \dots, dx_k\}$  définie par

$$\left\langle \frac{\partial}{\partial x_i}, dx_j \right\rangle = \delta_{i,j}, \quad \forall i, j = 1, \dots, n$$

où  $\delta$  représente le symbole de Kronecker et  $\langle, \rangle$  est le produit scalaire. Le fibré cotangent noté  $T^*X$  est alors donné par  $T^*X = \bigcup_{x \in X} T_x^* X$ .

Soit une fonction  $\varphi : X \rightarrow \mathbb{R}$  une fonction de classe  $C^\infty$ . La différentielle de  $\varphi$  est définie par l'expression  $d\varphi = \sum_{i=1}^n \frac{\partial \varphi}{\partial x_i} dx_i$  qui définit également l'application linéaire qui lui est tangente.  $d\varphi(x)$  est alors un élément de  $T_x^* X$  et si  $f(x) \in T_x X, \forall x$ , le produit scalaire précédent devient

$$\langle d\varphi, f \rangle = \sum_{i,j=1}^n f_i \frac{\partial \varphi}{\partial x_j} \left\langle \frac{\partial}{\partial x_i}, dx_j \right\rangle = \sum_{i=1}^n f_i \frac{\partial \varphi}{\partial x_i} = L_f \varphi$$

On définit de la sorte la différentielle  $d\varphi$  de  $\varphi$  comme un objet abstrait sans références par à ses valeurs aux points  $x \in X$

**Définition B.14** On appelle *forme différentielle de degré 1, ou 1-forme, une application de*

$$w : X \rightarrow T_x^* X \quad (\text{B.3})$$

$$x \mapsto w(x) = \sum_{i=1}^n w_i dx_i \quad (\text{B.4})$$

$w(x)$  est une combinaison linéaire des covecteurs de la base locale de  $T_x^*X$  à coefficients  $w_i$  qui sont de classe  $C^\infty$  sur  $X$ . L'ensemble des sections  $C^\infty$  de  $T^*x$  est un espace vectoriel noté  $\Lambda^1(X)$ .

Le produit scalaire d'une 1-forme et d'un champ de vecteurs  $f$  est donné par

$$\langle w, f \rangle = \sum_{i=1}^n f_i w_i$$

**Remarque B.6.1** En toute rigueur, une 1-forme n'est pas forcément une différentielle de fonction. On dit qu'une 1-forme est exacte s'il existe une fonction  $\varphi$  telle que  $w = d\varphi$ . Par ailleurs, on dit qu'une 1-forme est fermée si  $dw = 0$ . Le lemme de Poincaré énonce que toute forme fermée sur un ouvert contractile est exacte.

Pour pouvoir définir le critère d'intégrabilité pour les 1-formes, nous allons définir la notion d'image réciproque.

**Définition B.15** Soit une 1-forme  $w$  sur la variété  $X$  de dimension  $n$  et un difféomorphisme local  $\varphi$  d'une variété  $Y$  de dimension  $m$  dans  $X$ . Nous appelons image réciproque de  $w$  par  $\varphi$ , notée  $\varphi^*w$ , la 1-forme sur  $Y$  définie par

$$\forall v \in TY : \langle \varphi^*w, v \rangle = \langle w, \varphi_* v \rangle$$

On peut vérifier que  $d\varphi^*w = \varphi^*dw$ .

Définissons également la variété intégrale.

**Définition B.16** Soit une famille de 1-formes  $\{w_1, \dots, w_r\}$  indépendantes. La variété  $Y$  est appelée variété intégrale du système différentiel

$$\begin{cases} w_1 = 0 \\ \vdots \\ w_r = 0 \end{cases} \quad (\text{B.5})$$

s'il existe une application différentiable  $\varphi$  de  $Y$  dans  $X$  telle que  $\varphi^*w_1, \dots, \varphi^*w_r$  soient nulles.

Le système (B.5) est appelé système de Pfaff s'il n'est composé que de 1-formes. Par ailleurs, le système

$$\begin{cases} w_1 = 0 \\ dw_1 = 0 \\ \vdots \\ w_r = 0 \\ dw_r = 0 \end{cases} \quad (\text{B.6})$$

est appelé fermeture du système (B.5). Si un système de Pfaff est équivalent algébriquement à sa fermeture, alors il est fermé.

La condition nécessaire et suffisante d'existence d'une solution est énoncée par le théorème dual du théorème de Frobenius

**THÉORÈME B.5** Un système de Pfaff de rang  $r$  est complètement intégrable dans un ouvert  $U \subset X$  si et seulement s'il est fermé dans  $U$ .

## **B.7 Conclusion**

Ce bref rappel de la géométrie différentielle et de l'algèbre de Lie permet de formuler et de résoudre le problème de l'existence d'un difféomorphisme qui redresse une distribution de vecteurs donnée, ou dans le cas le plus général, une 1-forme différentielle. Ce formalisme permet de déterminer la solution d'un ensemble d'équations différentielles. Le but de ce rappel est de situer les outils de l'algèbre des fonctions présentés au chapitre 3.

# Annexe C

## Simulations

### Introduction

Dans cette annexe, les résultats des simulations du système à trois cuves sont donnés.

### C.1 Fuite dans la 1<sup>re</sup> cuve $f_1$

Pour une défaillance permanente (figure C.1).

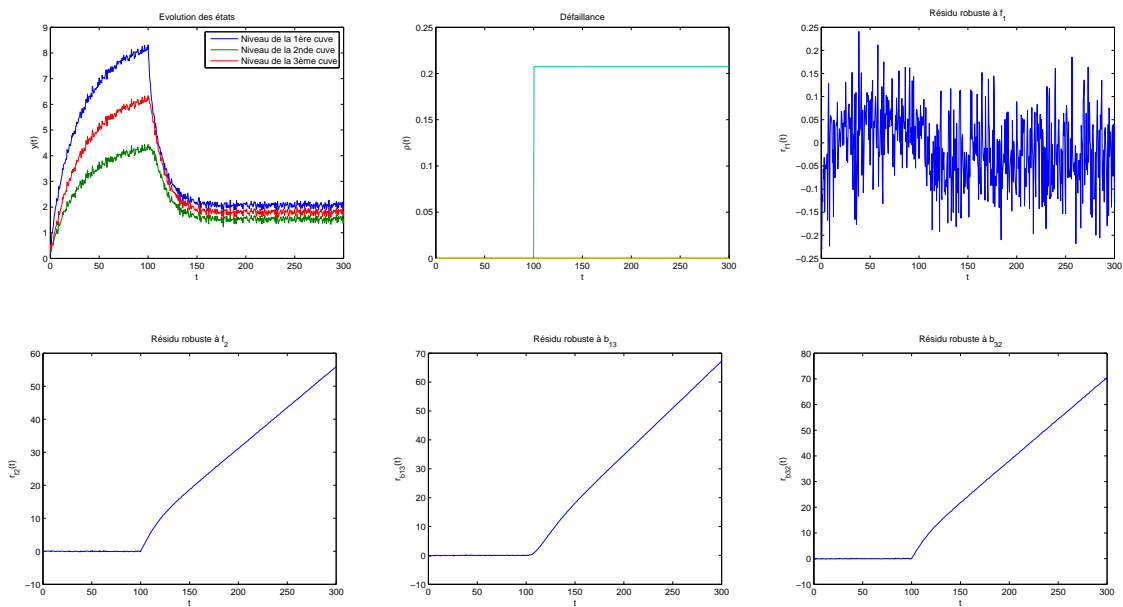


FIG. C.1 – Résidus dans le cas d’une défaillance  $f_1$  permanente

Pour une défaillance intermittente (figure C.2).

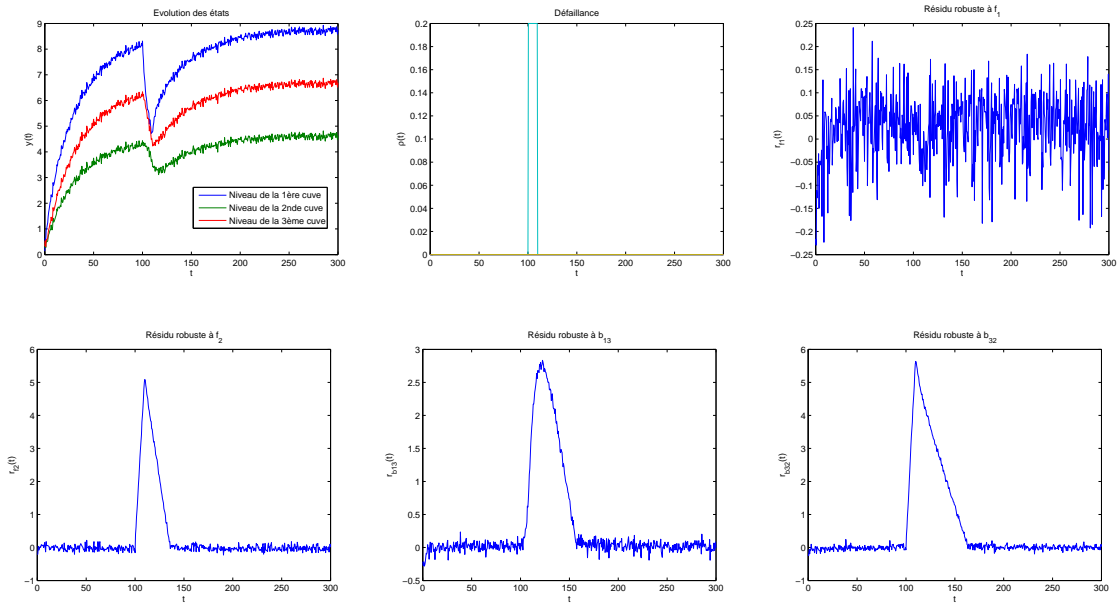


FIG. C.2 – Résidus dans le cas d’une défaillance  $f_2$  temporaire

## C.2 Fuite dans la 2<sup>e</sup> cuve $f_2$

Pour une défaillance permanente (figure C.3).

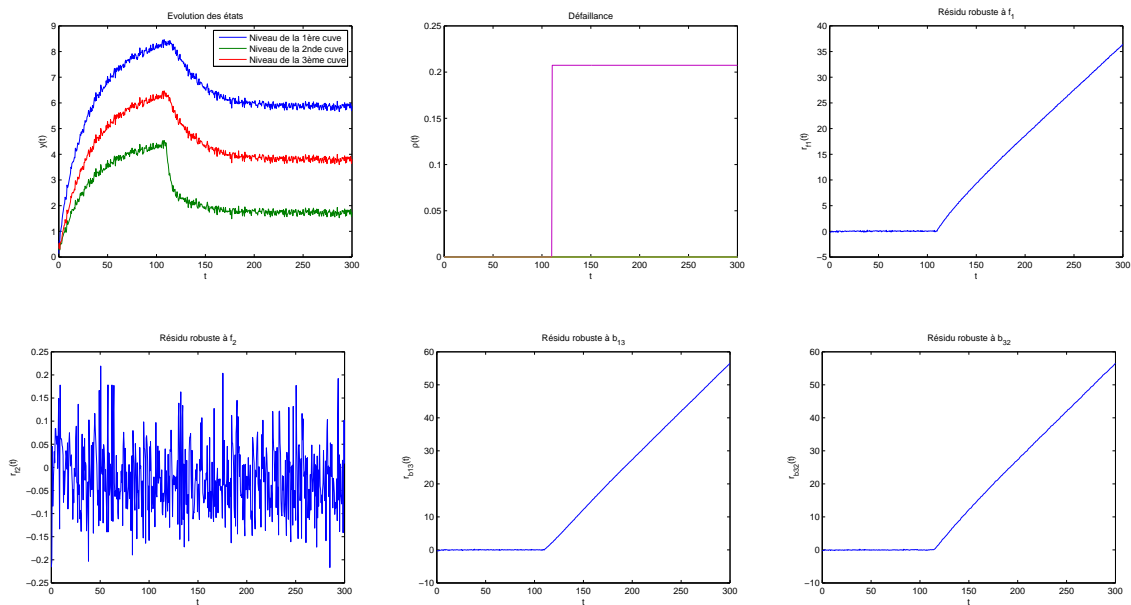


FIG. C.3 – Résidus dans le cas d’une défaillance  $f_2$  permanente

Pour une défaillance intermittente (figure C.4).

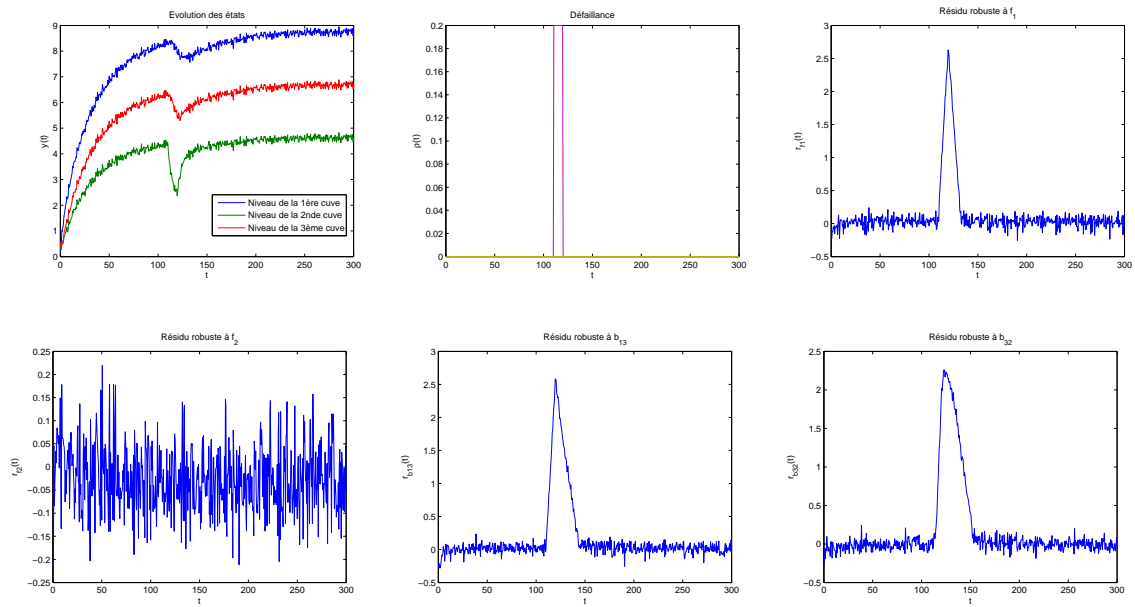


FIG. C.4 – Résidus dans le cas d’une défaillance  $f_2$  temporaire

### C.3 Fuite dans la 3<sup>e</sup> cuve $f_3$

Pour une défaillance permanente (figure C.5).

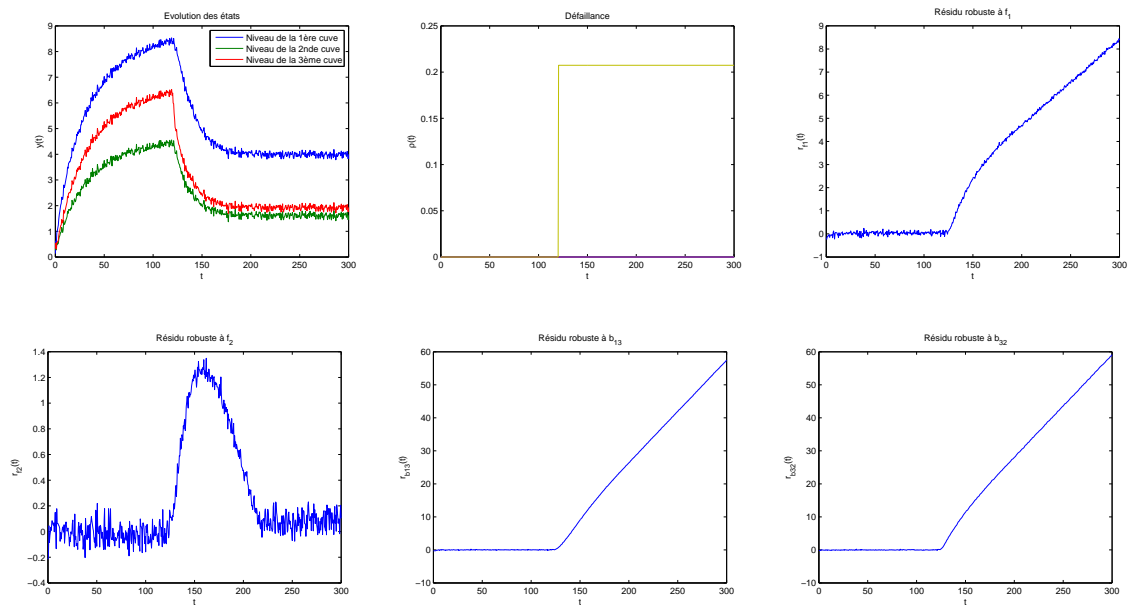


FIG. C.5 – Résidus dans le cas d’une défaillance  $f_3$  permanente

Pour une défaillance intermittente (figure C.6).



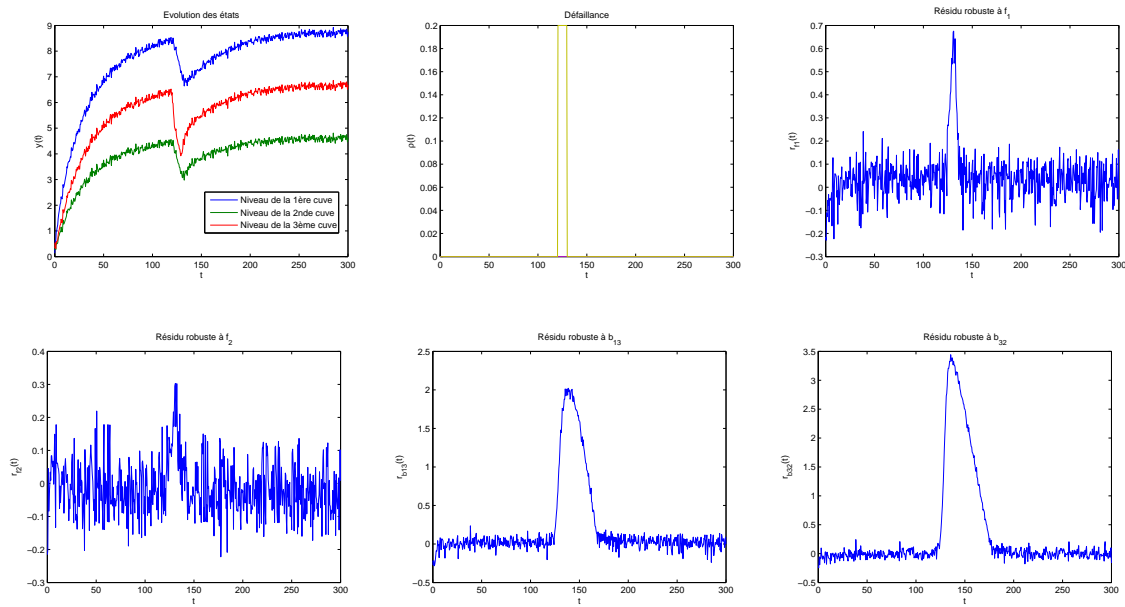


FIG. C.6 – Résidus dans le cas d’une défaillance  $f_3$  temporaire

### C.4 Fuite dans la 2<sup>e</sup> conduite $b_{32}$

Pour une défaillance permanente (figure C.7).

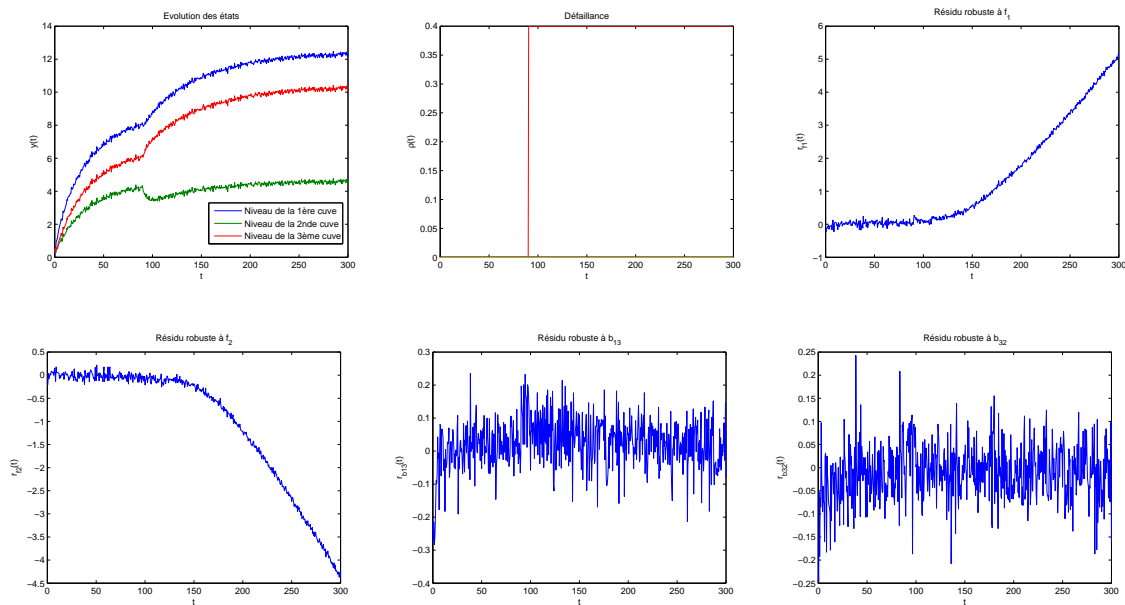


FIG. C.7 – Résidus dans le cas d’une défaillance  $b_{32}$  permanente

Pour une défaillance intermittente (figure C.8).

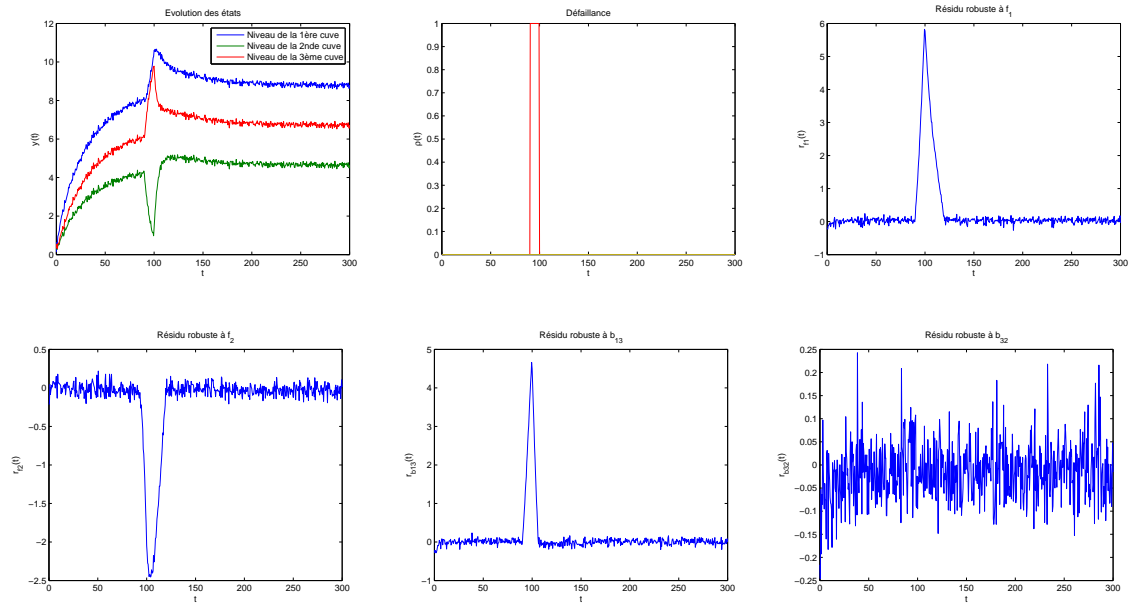


FIG. C.8 – Résidus dans le cas d’une défaillance  $b_{32}$  temporaire

## C.5 Fuite dans la conduite de sortie $b_{20}$

Pour une défaillance permanente (figure C.9).

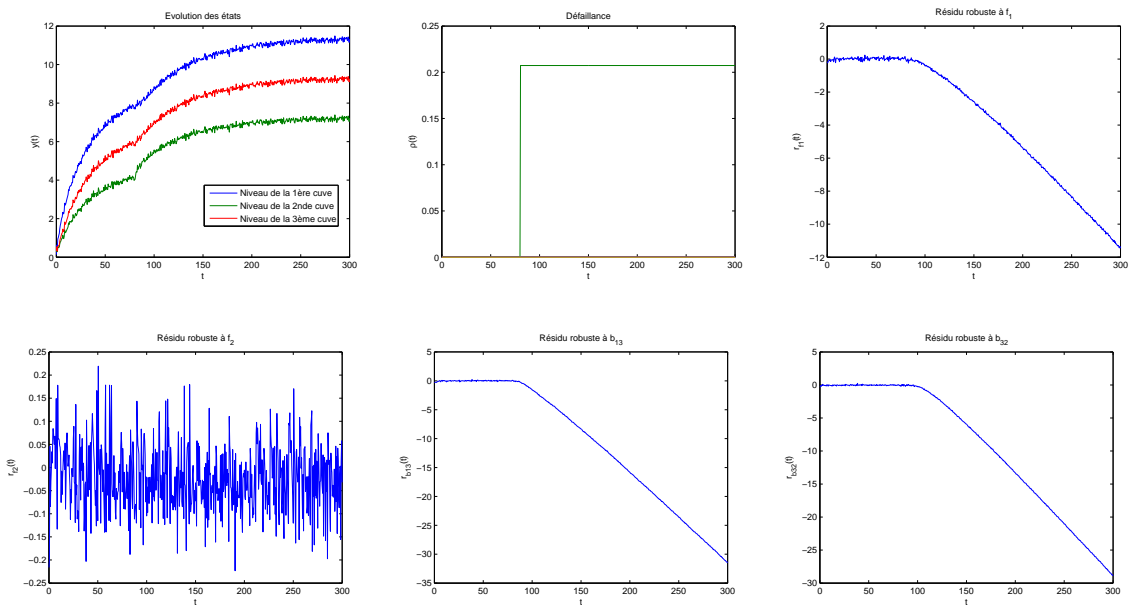
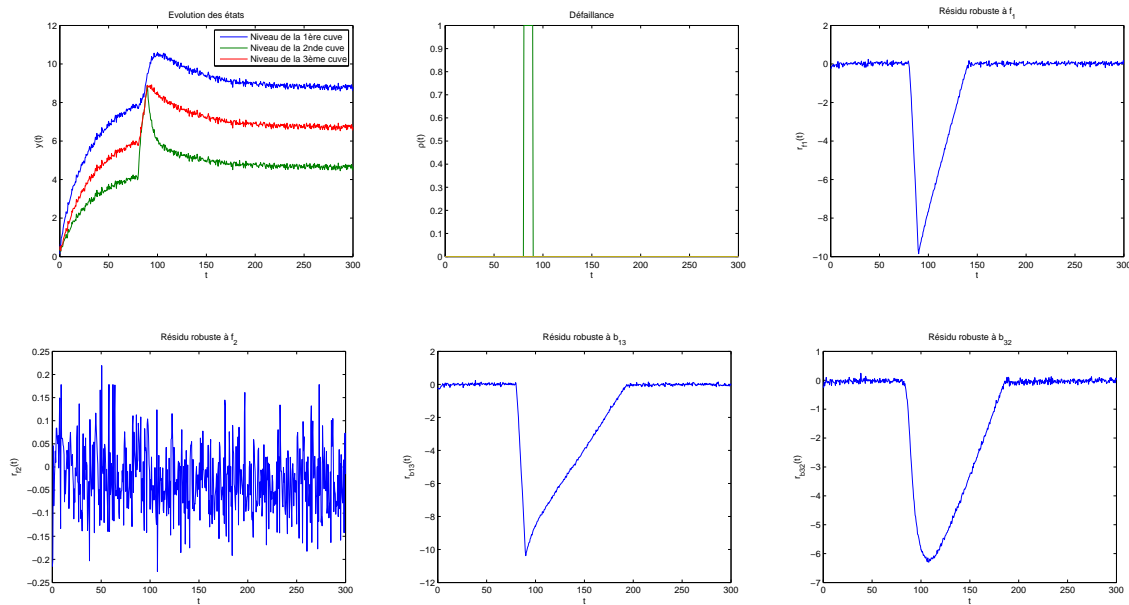


FIG. C.9 – Résidus dans le cas d’une défaillance  $b_{20}$  permanente

Pour une défaillance intermittente (figure C.10).

FIG. C.10 – Résidus dans le cas d’une défaillance  $b_{20}$  temporaire

## Discussion

Les simulations donnent les résultats escomptés. La défaillance  $f_3$  fait réagir les quatre résidus. Les défaillances  $f_2$  et  $b_{20}$  ne sont pas discernables, car elles font réagir les mêmes résidus. Remarquons cependant que la forme des résidus diffère, car une fuite entraîne une baisse du niveau de la deuxième cuve, et un blocage entraîne une montée du niveau. Les défaillances  $f_2$  et  $b_{20}$  qui sont *a priori* non-discernables, peuvent être discernées avec une analyse supplémentaire en fonction de l’état de la vanne de sortie. Une remarque peut être faite au sujet du 3<sup>e</sup> résidu (robuste à  $b_{13}$ ) lors de l’occurrence d’une défaillance  $b_{32}$ , où la réaction du résidu est noyée dans le bruit. La faible réaction du résidu est due à l’augmentation de la dimension du sous-système robuste à  $b_{32}$  en utilisant l’élimination de variables, ce qui a causé une baisse de la sensibilité du générateur de résidu.

# Bibliographie

- [BCC05] D. BERDJAG, C. CHRISTOPHE et V. COCQUEMPOT : Formulation algébrique du problème de génération de résidus pour la détection et la localisation de défaillances. *In Journées Doctorales et Nationales du GDR MACS JDMACS-JNMACS*, Lyon, FRANCE, Septembre 2005. GDR-MACS.
- [BCC06a] D. BERDJAG, C. CHRISTOPHE et V. COCQUEMPOT : An algebraic method for nonlinear system decomposition. *In 6th IFAC Symposium on Fault Detection Supervision and Safety for Technical Processes, SAFEPROCESS'2006*, Beijing, P.R. CHINA, August 2006. IFAC.
- [BCC06b] D. BERDJAG, C. CHRISTOPHE et V. COCQUEMPOT : Nonlinear model decomposition for fault detection and isolation system design. *In 45th IEEE Conference on Decision and Control*, San Diego CA, USA, December 2006. IEEE.
- [BCCJ06] D. BERDJAG, C. CHRISTOPHE, V. COCQUEMPOT et B. JIANG : Nonlinear model decomposition for robust fault detection and isolation using algebraic tools. *International Journal of Innovative Computing, Information and Control*, 2(6):1337–1353, December 2006.
- [BCD05] P. BOUYER, F. CHEVALIER et D. D'SOUZA : Fault diagnosis using timed automata. *In Proc. 8th International Conference on Foundations of Software Science and Computer Structures (FoSSaSC'05)*, pages 219–233, 2005. SED FDI temporisé.
- [Bea71] R.V. BEARD : *Failure accomodation in linear systems through self reorganisation*. Thèse de doctorat, Massachussets Institute of Technology, Cambridge, MA, 1971.
- [BKLS03] M. BLANKE, M. KINNAERT, J. LUNZE et M. STAROSWIECKI : *Diagnosis and Fault-Tolerant Control*. Springer, 2003. Ch. 5.
- [BN93] M. BASSEVILLE et I.V. NIKIFOROV : *Detection of Abrupt Changes - Theory and Application*. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1993.
- [BSC04] J. BIN, M. STAROSWIECKI et V. COCQUEMPOT : Fault diagnosis based on adaptive observer for a class of nonlinear systems with unknown parameters. *International Journal of Control*, 77(4):415–426, March 2004.
- [CCJ04] C. CHRISTOPHE, V. COCQUEMPOT et B. JIANG : Link between high gain observer-based residual and parity space one for fdi. *Transactions of Institute of Measurements and Control*, 26(4):325–337, 2004.

- [Chr01] Cyrille CHRISTOPHE : *Surveillance des systèmes non linéaires : Application aux machines électriques*. Thèse de doctorat, Université des Sciences et Technologies de Lille, Lille, november 2001.
- [Cla78] R.N. CLARK : A simplified instrument detection scheme. *IEEE Transactions on Aerospace and Electronic Systems*, 14:456–465, 1978.
- [Cla89] R.N. CLARK : State estimation schemes for instrument fault detection. *In Fault Diagnosis in dynamic systems*, 1989.
- [CLO91] D. COX, J. LITTLE et D. O'SHEA : *Ideals, Varieties, and Algorithms*. Springer Verlag, 1991.
- [CP99] J. CHEN et R.J. PATTON : *Robust model-based fault diagnosis for dynamic systems*. Kluwer academic publishers, 1999.
- [CW84] E.Y. CHOW et A.S. WILLSKY : Analytical redundancy and the design of robust failure detection systems. *In IEEE Transactions on Automatic Control*, volume 29, pages 603–614, 1984.
- [DGH79] K.C. DALI, E.J. GAI et J.V. HARRISON : Generalized likelihood test for fdi in redundancy sensor configuration. *Journal of guidance and control*, 2:9–17, 1979.
- [DI00] C. DEPERSIS et A. ISIDORI : A differential geometric approach to nonlinear fault detection and isolation. *In Safeprocess'2000 IFAC Symposium*, pages 542–547, Budapest , Hungary, 2000.
- [DI01] C. DEPERSIS et A. ISIDORI : A geometric approach to nonlinear fault detection and isolation. *IEEE Transactions on Automatic Control*, 46(6):853–865, June 2001.
- [Dio91] S. DIOP : Elimination in control theory. *Mathematics of Control, Signals, and Systems*, 4:17–32, 1991.
- [DJDZ99] S.X. DING, T. JEINISH, E.L. DING et D. ZHOU : Application of observer based fdi schemes to the three tank system. *In Proceedings of the 4th European Control Conference, ECC'99*, Karlsruhe, Germany, 1999.
- [FJ03] M. FLIESS et C. JOIN : An algebraic approach to fault diagnosis for linear systems. *In Proc. CESA Conf.*, 2003.
- [FJSR04] M. FLIESS, C. JOIN et H. SIRA-RAMÍREZ : Robust residual generation for linear fault diagnosis : an algebraic setting with examples. *International Journal of Control*, 1977:1223 – 1242, 2004.
- [FKKS02a] E. FOMINA, A. KEEVALLIK, M. KRUS et A. SUDNITSON : A decomposition procedure for register-transfer level power management. *In International Conference on Computer Systems and Technologies CompSysTech'2002*, 2002. AP decomposition FSM.
- [FKKS02b] E. FOMINA, A. KEEVALLIK, M. KRUS et A. SUDNITSON : Web-based tools for finite state machine decomposition with analysis of information flows. *In Proc. 8th International Baltic Electronics Conference BEC'2002*, pages 165–168, 2002.

- [Fra1] P.M. FRANK : *Fault diagnosis in dynamic system via state estimation - a survey*, pages 35–98. D. Reidel Press, 1.
- [Fra90] P.M. FRANK : Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy - a survey and some new results. *Automatica*, 26:459–474, 1990.
- [FVZ99] V. F. FILARETOV, M. K. VUKOBRATOVIC et A. N. ZHIRABOK : Observer-based fault diagnosis in manipulation robots. In Elsevier Science LTD., éditeur : *Mechatronics*, volume 9, pages 929–939, 1999.
- [FVZ03] V. F. FILARETOV, M. K. VUKOBRATOVIC et A. N. ZHIRABOK : Parity relation approach to fault diagnosis in manipulation robots. In Elsevier Science LTD., éditeur : *Mechatronics*, volume 13, pages 141–152, 2003.
- [FW89] P.M. FRANK et J. WUNNENBERG : *Fault diagnosis in dynamic systems : Theory and application*, chapitre Robust fault diagnosis using robust using unknown input unknown schemes, pages 47–98. Prentice-Hall, 1989.
- [Ger98] Janos J. GERTLER : *Fault Detection and Diagnosis in Engineering Systems*. New York - Basel, 1998.
- [Gha05] M. GHAZEL : *Surveillance des systèmes à évènements discrets à l'aide des réseaux de Petri Temporels*. Thèse de doctorat, Ecole Centrale de Lille, 2005. chap. 6.
- [Had03] C.N. HADJICOSTIS : Encoded finite-state machines for non concurrent error detection and identification. In *Proceedings of the 2003 International Symposium on Circuits and Systems, ISCAS '03.*, 2003.
- [HKE99a] H. HAMMOURI, M. KINNAERT et E.H. ELYAAGOUBI : *New Directions in Nonlinear Observer Design*, chapitre Application of Nonlinear Observers to Fault Detection and Isolation, pages 423–443. Springer-Verlag, 1999.
- [HKE99b] H. HAMMOURI, M. KINNAERT et E.H. ELYAAGOUBI : Observer based approach to fault detection and isolation for nonlinear systems. *IEEE Transactions of Automatic Control*, 44:1879–1884, 1999.
- [HKE01] H. HAMMOURI, M. KINNAERT et E.H. ELYAAGOUBI : A geometric approach to fault detection and isolation for bilinear systems. *IEEE Transactions of Automatic Control*, 46:1451–1455, 2001.
- [HS66] J. HARTMANIS et R. STEARNS : *The algebraic structure theory of sequential machines*. Prentice-Hall, New York, 1966.
- [HV99] C.N. HADJICOSTIS et G.C. VERGHESE : Structured redundancy for fault tolerance in lti state-space models and petri nets. *Kybernetika*, 35:39–55, 1999.
- [HZKW03] S. HASHTRUDI ZAD, R.H. KWONG et W.M. WONHAM : Fault diagnosis in discrete-event systems : framework and model reduction. *IEEE Transactions on Automatic Control*, 48(7):1199– 1212, July 2003.
- [IF91] R. ISERMANN et B. FREYERMUTH : Process fault diagnosis based on process model knowledge - part 1 : Principles for fault diagnosis with parameter estimation. *Transactions of the ASME*, 113:620–626, 1991.

- [Ise84] R. ISERMANN : Process fault-detection based on modelling and estimation methods - a survey. *Automatica*, 20:387–404, 1984.
- [Ise05] R. ISERMANN : Model-based fault detection and analysis - status and application. *Annual Reviews in Control*, 29:71–85, 2005.
- [Isi95] A. ISIDORI : *Nonlinear Control Systems 3rd ed.* Springer-Verlag, Berlin, 1995.
- [Jon73] H.L. JONES : *Failure detection in linear systems.* Thèse de doctorat, Massachusetts Institute of Technology, Cambridge, MA, 1973.
- [JPS03] C. JOIN, J.-C. PONSART et D. SAUTER : Nonlinear filters bank for fdi : an alternative to the fundamental problem of residual generation. *In Proc. 5th IFAC symp. on Fault Detection Supervision and Safety for Technical Processes, SAFEPROCESS'2003*, pages pp. 645–650, Washington D.C., 2003.
- [Kal62] R.E. KALMAN : Canonical structure of linear dynamical systems. *Proceedings of the National Academy of Sciences*, 48:596–600, 1962.
- [KSGF99] B KÖPPEN-SELIGER, E.A. GARCIA et P.M. FRANK : Fault detection : different strategies for modelling applied to the three tank benchmark - a case study. *In Proceedings of the 4th European Control Conference, ECC'99*, Karlsruhe, Germany, 1999.
- [Loo01] T. LOOTSMA : *Observer-based fault detection and isolation for nonlinear systems.* Thèse de doctorat, Aalborg university,, Denmark, 2001.
- [LTS<sup>+</sup>01] S. LAFORTUNE, D. TENEKETZIS, R. SENGUPTA, M. SAMPATH et K. SINNAMOHIDEEN : Failure diagnosis of dynamic systems : An approach based on discrete event systems. *In Proceedings of the American Control Conference*, Arlington, VA, USA, June 2001.
- [Lue66] D.G. LUENBERGER : Observing the state of a linear system. *In IEEE Trans. Mil. Electron.*, volume MIL-8, pages 74–84, 1966.
- [LW88] X. LIN et W. WONHAM : On observability of discrete-event systems. *Information Sciences*, 44:179–198, 1988.
- [LWC05] M.L. LEUSCHEN, I.D. WALKER et J.R. CAVALLARO : Fault residual generation via nonlinear analytical redundancy. *IEEE Transactions on Control Systems Technology*, 13(3):452 – 458, May 2005.
- [Mas86] M.A. MASSOUMNIA : A geometric approach to the synthesis of failure detection filters. *In IEEE Transactions on automatic control*, volume AC-31, pages 839–846. IEEE, September 1986.
- [MCC<sup>+</sup>97] D. MAQUIN, V. COCQUEMPOT, J.P. CASSAR, M. STAROSWIECKI et J. RAGOT : Generation of analytical redundancy relations for fdi purposes. *In IEEE Int. Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives, SDEMPED'97*, pages 270–276, Carry-le Rouet, France, September 1997.
- [MH88] E.A. MISAWA et J.K. HEDRICK : Nonlinear observers - a state of the art survey. *In Proceedings of the Symposium, ASME Winter Annual Meeting*, pages 59–69, Chicago, IL ; USA, 27 Nov.-2 Dec. 1988 1988.

- [Mir79] L.A. MIRONOVSKY : Functional diagnosis of linear dynamic systems. *Automation and remote control*, 8:120–128, 1979.
- [MVW89] M.A. MASSOUMNIA, G.C. VERGHESE et A.S. WILLSKY : Failure detection and identification. *IEEE Transactions on Automatic Control*, 31(3):316–321, 1989.
- [Pat94] R.J. PATTON : Robust model-based fault diagnosis : The state of the art. *In Proceedings IFAC Symposium SAFEPROCESS'94*, volume 1, pages 1–24, Espoo, Finland, June 1994. IFAC.
- [PC91] R.J. PATTON et J. CHEN : A re-examination of the relationship between parity space and observer-based approaches in fault diagnosis. *European journal of diagnosis and safety in automation*, 1:183–200, 1991.
- [Pet99] S. PETTERSSON : *Analysis and Design of Hybrid Systems*. Thèse de doctorat, Chalmers University of Technology, Göteborg, Sweden, 1999.
- [PFC00] R. PATTON, P.M. FRANK et R.N. CLARK : *Issues of Fault Diagnosis for Dynamic Systems*. Springer, 2000.
- [Rit50] J.F. RITT : *Differential Algebra*. American Mathematical Society, 1950.
- [SCV01] M. STAROSWIECKI et G. COMTET-VARGA : Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica*, 37:687–699, 2001.
- [Shu91] A.Ye SHUMSKY : Fault isolation in nonlinear dynamic systems by functional diagnosis. *Automation and Remote Control*, 12:148–155, 1991.
- [Shu94] A.Ye SHUMSKY : An approach to failure isolation in nonlinear uncertain systems. *In Proceedings of the American control conference*, Baltimore, Maryland, June 1994.
- [Shu97] A. Y. SHUMSKY : Robust residual generation for diagnosis of nonlinear systems : parity relation approach. *In Preprints of the 3rd IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS 97*, 1997.
- [Shu99] A. SHUMSKY : Parity space approach to fault diagnosis in nonlinear systems. *In European Control Conference*, 1999.
- [Shu00] A.Ye SHUMSKY : Continuous-time parity relation method for fault diagnosis in non linear uncertain systems. *In Preprints of the 4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS'2000*, volume 2, pages 998–113, Budapest, Hungary, June 2000.
- [SSL<sup>+</sup>95] M. SAMPATH, R. SENGUPTA, S. LAFORTUNE, K. SINNAMOHIDEEN et D. TENEKETZIS : Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, 40:1555–1575, 1995.
- [SSL<sup>+</sup>96] M. SAMPATH, R. SENGUPTA, S. LAFORTUNE, K. SINNAMOHIDEEN et D. TENEKETZIS : Failure diagnosis using discrete-event models. *IEEE Transactions on control systems technology*, 4:105–124, 1996.



- [SZ05] A.Ye. SHUMSKY et A.N. ZHIRABOK : Nonlinear diagnostic filter design : Algebraic and geometric points of view. *In IFAC World Cong.*, Prague, July 2005.
- [TU03] S. TAKAI et T. USHIO : Supervisory control of a class of concurrent discrete event systems. *In Proceedings of the 16th Workshop on circuits and systems*, pages 471–476, 2003.
- [UOO98] T. USHIO, I. ONISHI et K. OKUDA : Fault detection based on petri net models with faulty behaviours. *In International Conference on Systems, Man, and Cybernetics*, volume 1, pages 113–118. IEEE, Oct 1998.
- [Ush02] T. USHIO : Detection of human errors based on discrete event system theory. *In Proceedings of the 41st SICE Annual Conference*, numéro 1760–1763, 2002. SED FDI Petri.
- [Utk81] V.I. UTKIN : Principles of identification using sliding regimes. *Sov. Phys. Dokl.*, 26(3):271–272, 1981.
- [UUTY03] Y. UKAWA, T. USHIO, S. TAKAI et S. YAMAMOTO : Fault diagnosis in a class of concurrent discrete event systems. *In Proceedings of the 42st SICE Annual Conference*, pages 2360–2363, 2003. SED FDI concurrent simultan e.
- [Zad99] S. Hashtrudi ZAD : *Fault diagnosis in discrete-event and hybrid systems*. Th ese de doctorat, University of Toronto, 1999.
- [Zhi96] A.N. ZHIRABOK : Fault diagnosis il linear dynamic systems via nonlinear observer. *In Proceedings of the 35th Conference on decidsion and control*, Kobe, JAPAN, 1996. IEEE.
- [Zhi97] A.N. ZHIRABOK : Fault detection and isolation : linear and nonlinear systems. *In Proceedings of the IFAC Symposium Safeprocess'97*, pages 903–908, Hull, 1997.
- [Zhi01] A.N. ZHIRABOK : On canonical decomposition of nonlinear systems. *In M. THOMA,  diteur : Lecture notes in control and Information Sciences*, volume 259, pages 555–564. Springer-Verlag, 2001.
- [Zhi06] A.N. ZHIRABOK : Nonlinear dynamic systems : their canonical decomposition based on invariant functions. *Automation and Remote Control*, 67(4):517–528, 2006.
- [ZKW99] S.H. ZAD, R.H. KWONG et W.M. WONHAM : Fault diagnosis in timed discrete-event systems. *In Proceedings of the IEEE Conference on Decision and Control*, volume 2, pages 1756–1761, 1999.
- [ZM04] A.N. ZHIRABOK et A.M. MICHCHENKO : Global transformation of nonlinear dynamic systems into controllability canonical forms. *In MTNS'2004*, 2004.
- [ZP93a] A.N. ZHIRABOK et O.V. PREOBRAZHENSKAYA : Instrument fault detection in nonlinear dynamic systems. *In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 5, pages 114–119, June 1993.

- [ZP93b] A.N. ZHIRABOK et O.V. PREOBRAZHENSKAYA : Robust fault detection in technological equipment. *In Japan Int'I Electronics Manufacturing Technology Symposium*. IEEE, 1993.
- [ZS87] A.N. ZHIRABOK et A.Ye. SHUMSKY : Functional diagnosis of continuous dynamic systems described by equations whose right-hand side is polynomial. *Automation and Remote Control*, 8:154–164, 1987.
- [ZS93] A.N. ZHIRABOK et A.Ye. SHUMSKY : A new mathematical techniques for nonlinear systems research. *In Proceedings 12th World Congress IFAC*, pages 485–488, Sidney, 1993.

# **Méthodes algébriques pour la décomposition de modèles comportementaux :**

## **Application à la détection et à la localisation de défaillances.**

### **Résumé**

Cette thèse traite de la détection et de la localisation des défaillances à base de modèles comportementaux. L'approche suggérée consiste à extraire du modèle comportemental un sous-modèle découplé par rapport aux perturbations et par rapport à un sous-ensemble de défaillances et couplé par rapport au sous-ensemble complémentaire de défaillances. A l'aide de ce sous-modèle, des indicateurs de défaillances structurés sont générés. Ces indicateurs possèdent intrinsèquement des propriétés de robustesse aux perturbations et à certaines défaillances et sont sensibles au reste des défaillances. Cette approche repose donc sur la décomposition du modèle comportemental du système en sous-modèles.

Une méthodologie de décomposition est présentée pour traiter les modèles continus et les modèles événementiels de la même manière. Cette méthodologie de décomposition est formulée en utilisant des outils algébriques connus sous le nom d'*algèbre des paires* et d'*algèbre des fonctions*. Les deux algèbres sont présentées et leurs opérateurs sont détaillés et illustrés. Il est montré que la procédure de décomposition décrite pour les modèles continus en utilisant l'algèbre des fonctions est similaire à celle décrite pour les modèles événementiels en utilisant l'algèbre des paires. Une procédure de décomposition générale est ensuite proposée pour tous les modèles comportementaux déterministes.

Les résultats théoriques sont illustrés par la décomposition des modèles comportementaux (continu et événementiel) du système à trois cuves, et par la synthèse d'un module de détection et de localisation de défaillances pour chaque cas traité.

**Mots clés :** Génération de résidus robustes, décomposition de modèles, détection et localisation de défaillances, systèmes non-linéaires, systèmes événementiels, méthodes algébriques, système à trois cuves.

## **Behavioral models decomposition using algebraic methods :**

### **Application on fault detection and isolation.**

### **Abstract**

This thesis concerns fault detection and isolation using behavioral models. The suggested approach consists in the determination of a sub-system which is decoupled from the perturbation and from a subset of faults and sensitive to the remaining faults. Using this sub-system, structured fault indicators are generated. These indicators are intrinsically robust to the perturbations and to some faults and sensitive to the remaining faults. Thus, this approach is based on the decomposition of the behavioral model of the process to monitor into sub-systems.

A decomposition methodology is proposed to handle continuous and discrete models in a similar way. Algebraic tools, known as *pair algebra* and *algebra of functions*, are used to formulate the decomposition methodology. Both algebras are presented, and their operators detailed and illustrated. It is shown that the decomposition procedure for the continuous case based on algebra of functions is similar to the discrete case based on pair algebra. Finally, a decomposition methodology for all types of deterministic behavioral models is proposed.

The theoretical results are illustrated by the decomposition of the behavioral models (continuous and discrete) of the three tank process, and the synthesis of fault detection and isolation module for both cases.

**Keywords :** Robust residual generation, model decomposition, fault detection and isolation, nonlinear systems, discrete-event systems, algebraic methods, three-tanks system.