

N° d'ordre : 40560

UNIVERSITE LILLE 1 : SCIENCES ET TECHNOLOGIES

THESE

présentée en vue d'obtenir le grade de

DOCTEUR

en

Spécialité : AUTOMATIQUE

par

Vincent de Flaugergues

Algorithmes d'analyse structurelle
pour le diagnostic de systèmes complexes.
Application à la boucle d'air d'un moteur Diesel

Partie publique

Soutenue le 6 juillet 2011 devant le jury d'examen :

Président

Rapporteur *Frédéric Hamelin, Professeur, CRAN, Université Henri Poincaré*

Rapporteur *Michel Vergé, Professeur, LMSP, Arts et Métiers ParisTech*

Membres *Jean-Philippe Cassar, Professeur, LAGIS, Université Lille1*
Erik Frisk, Associate Professor, Université de Linköping, Suède
Jean-Michel Dion, Directeur de Recherche CNRS, GIPSA-LAB, INP Grenoble
Marco Pengov, Docteur Ingénieur, PSA Peugeot Citroën

Invité *Nicolas Lefebvre, Docteur Ingénieur, PSA Peugeot Citroën*

Directeurs de thèse *Vincent Cocquempot, Professeur, LAGIS, Université Lille1*
Mireille Bayart, Professeur, LAGIS, Université Lille1

Remerciements

Je souhaite remercier en préambule de ce rapport mes directeurs de thèse, Vincent Cocquempot et Mireille Bayart, qui m'ont transmis leur savoir et leurs connaissances, avec une patience dont je leur sais gré. Je les remercie de leur disponibilité, notamment devant les problèmes auxquels j'ai pu être confronté durant cette thèse, ainsi que de l'exigence soutenue dont ils ont fait preuve à l'égard de mon travail, tout au long de ces trois années.

Je suis reconnaissant envers Frédéric Hamelin, Professeur à l'Université Henri Poincaré, et envers Michel Vergé, Professeur aux Arts et Métiers Paritech, tous deux rapporteurs de cette thèse, d'avoir fait la critique de ce travail et d'y avoir apporté des commentaires qui ont aidé à l'enrichir.

Je remercie Jean-Philippe Cassar, Professeur à l'Université de Lille¹, Jean-Michel Dion, Directeur de Recherche CNRS à l'INP Grenoble, et Erik Frisk, Professeur à l'Université de Linköping, de l'intérêt qu'ils portent à ce travail en acceptant de l'examiner, et, lorsque nous avons eu l'occasion de nous rencontrer, de l'intérêt qu'ils lui ont accordé.

Mes tuteurs successifs en entreprise, Nicolas Lefebvre, Sylvain Millemann, et Marco Pengov, ont toute ma gratitude, car la confiance qu'il m'ont témoignée pour mener ce sujet, leurs encouragements, leurs explications, conseils et directives ont été décisifs dans l'accomplissement de ce travail.

Enfin, je remercie Jean-Yves Marteau de m'avoir accueilli dans le service *C3E*, dont il assume la direction au sein de la Direction de la Recherche, et pour les bonnes conditions d'études qu'il nous a procurées.

Table des matières

Nomenclature	9
Introduction	11
I Diagnostic des systèmes complexes : algorithmes d'analyse structurelle pour la recherche des sous-systèmes surveillables	15
1 Analyse Structurelle et Diagnostic	19
1.1 Principes généraux du diagnostic à base de modèles	24
1.1.1 La terminologie du diagnostic	24
1.1.2 Le diagnostic à base de modèle	25
1.1.3 Localisation d'un défaut	27
1.2 Les fondements de l'Analyse Structurelle	30
1.2.1 Matrice d'incidence et graphe biparti	31
1.2.2 Modélisation structurelle des défauts	33
1.2.3 Décomposition canonique	34
1.2.4 Recherche des sous-systèmes surveillables	34
1.2.5 Calcul a posteriori de la table d'isolabilité structurelle.	36
1.3 Méthode de génération des résidus	36
1.3.1 Choix d'une méthode de génération de résidus	37
1.3.2 Aide à la conception des générateurs de résidu	37
1.3.3 Caractérisation des sous-systèmes surveillables calculables	42
1.4 Contenu et apports du mémoire	45
2 Algorithmes optimisés pour la recherche exhaustive des sous-systèmes surveillables	47
2.1 Introduction : objectifs et aperçu des développements algorithmiques	51
2.1.1 Critère de validité des sous-systèmes surveillables	51
2.1.2 Présentation des algorithmes actuels	53

2.1.3	Prise en compte des non-inversibilités et d'un cahier des charges dans les algorithmes actuels	58
2.2	Détermination des inversibilités sur les contraintes différentielles	61
2.3	Décomposition canonique modifiée en intégrant des inversibilités	64
2.4	Etudes a priori de l'isolabilité des fautes	73
2.4.1	Introduction	73
2.4.2	Analyse de l'isolabilité en tenant compte des contraintes d'inversibilité	73
2.5	Intégration des non-inversibilités dans la recherche exhaustive des sous-systèmes surveillables	74
2.5.1	Algorithme 1 : simplification du système et recherche des couplages complets orientables	76
2.5.2	Algorithme 2 : Parcours de l'arbre des <i>PSO</i> en utilisant la décomposition canonique modifiée	80
2.5.3	Algorithme 3 : Parcours de l'arbre des <i>PSO</i> avec évitement des boucles algébriques (causalité intégrale)	83
2.6	Prise en compte du cahier des charges	85
2.7	Récapitulatif	86
3	Présentation de la plate-forme logicielle développée et application sur un exemple	89
3.1	Présentation des fonctionnalités de l'outil logiciel	93
3.1.1	Ecriture du modèle structurel	93
3.1.2	Analyse du modèle structurel	95
3.2	Application au benchmark <i>DAMADICS</i>	97
3.2.1	Modèle structurel du benchmark <i>DAMADICS</i>	99
3.2.2	Sous-systèmes surveillables du benchmark <i>DAMADICS</i>	100
3.2.3	Isolabilité des fautes du benchmark <i>DAMADICS</i>	102
4	Conclusion et perspectives	107
4.1	Simulation des résidus et robustesse	107
4.2	Choix d'un sous-système surveillable	108
4.3	Comment prendre en compte les modes de fonctionnement ?	109
4.4	Analyse structurelle par niveaux	109

II Etude pour la conception du diagnostic fonctionnel de la boucle d'air

d'un moteur diesel (confidentiel)	111
Conclusion	113
Bibliographie	115

Table des matières

Nomenclature

Acronymes

<i>DAE</i>	<i>Differential Algebraic Equation</i> (équation différentielle algébrique)
<i>DAMADICS</i>	<i>Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems</i>
<i>FDI</i>	<i>Fault Detection and Isolation</i>
<i>IA</i>	Intelligence Artificielle
<i>MSO</i>	<i>Minimal Structurally Overdetermined</i>
<i>ODE</i>	<i>Ordinary Differential Equation</i> (équation différentielle ordinaire)
<i>PC</i>	<i>Possible Conflicts</i>
<i>PSO</i>	<i>Proper Structurally Overdetermined</i>
<i>RRA</i>	Relation de Redondance Analytique
<i>SCC</i>	<i>Strongly Connected Components</i> (composantes fortement connexes)
<i>SST</i>	Sous-Systèmes Testables

Notations

S	Matrice d'incidence structurelle
S^+	Partie sur-contrainte de S
S^0	Partie juste-contrainte de S
S^-	Partie sous-contrainte de S
S^*	Matrice structurelle ne contenant pas les arcs non-inversibles
$S^{+,*}$	S^+ ne contenant pas les arcs non-inversibles
$S_{+}^{+,*}$	Partie sur-contrainte de $S^{+,*}$
$S_0^{+,*}$	Partie juste-contrainte de $S^{+,*}$
$S_{-}^{+,*}$	Partie sous-contrainte de $S^{+,*}$
S_{mod}^+	Partie sur-contrainte de S^+
S_{mod}^0	Partie juste-contrainte de S^0
S_{mod}^-	Partie sous-contrainte de S^-
S_{NA}^+	Partie non atteignable de S^+
S_A^+	Complément de S_{NA}^+ dans S^+

Nomenclature

Introduction

Ce rapport présente mon travail de doctorat, réalisé sous la direction scientifique des Professeurs Vincent Cocquempot et Mireille Bayart, encadré par les Docteurs Marco Pengov et Nicolas Lefebvre, et effectué en collaboration, dans le cadre d'un contrat CIFRE, entre le Laboratoire d'Automatique, Génie Informatique et Signal (LAGIS) de l'Université Lille 1, et l'entreprise PSA Peugeot Citroën.

Contexte. Les moteurs Diesel actuels sont des systèmes complexes, commandés par un grand nombre d'actionneurs (pneumatiques, électriques, ...), et dont le contrôle-commande doit traiter de nombreuses informations (pressions, températures, positions de vanne, vitesse, ...). Cette complexité tend à s'accroître, afin de respecter les normes environnementales, et de satisfaire les exigences des clients en termes d'agrément de conduite, de consommation, de confort, et enfin, de sécurité. La multiplication des fonctions (agrément, réduction de la consommation, ...), des organes (calculateurs notamment), et donc des causes de panne, rend nécessaire la conception de systèmes de diagnostic performants.

Les systèmes de diagnostic peuvent être débarqués – ce sont les outils utilisés en garage – ou embarqués – ils informent en temps réel le conducteur de la présence de défauts. Dans le secteur automobile, la conception des systèmes embarqués ou débarqués de diagnostic repose essentiellement sur le retour d'expérience des systèmes précédemment traités et sur les études de sûreté de fonctionnement. En embarqué, le diagnostic utilise des tests électriques (capteurs), des tests de seuil (vérification qu'une grandeur est dans une plage de fonctionnement), et des tests de cohérence des signaux (cohérence estimation-mesure).

Cette thèse s'inscrit dans le cadre de la recherche des tests de cohérence des signaux, et a été réalisée au service C3E (Composants Energie Electricité Electronique) de la Direction de la Recherche (DRD/DRIA/DSTF/C3E) du groupe PSA Peugeot Citroën, au sein de l'équipe Signal et Diagnostic SID.

La problématique principale du sujet est la suivante : dès que l'on veut traiter des « systèmes » complexes, composés de nombreux sous-systèmes, comportant un grand nombre d'actionneurs et de capteurs, la recherche des tests de cohérence n'est pas évidente. Ce sujet est donc né de l'intérêt du « métier » diagnostic, qui réalise le diagnostic des fonctions et des systèmes véhicule, pour une méthodologie, adoptant une vision « système », et permettant :

- de connaître le potentiel de diagnostic d'un système ;
- de connaître l'ensemble des tests de cohérence pouvant être réalisés. Cette connaissance pourra permettre de sélectionner des tests de diagnostic, embarqués ou débarqués, plus

Introduction

- efficaces ;
- de justifier, d'appuyer, ou éventuellement d'orienter les choix d'instrumentation lors de la conception, pour faciliter le diagnostic.

La méthode d'Analyse Structurale s'appuie sur une description du système sous la forme d'un graphe. Ce graphe représente les interconnexions entre composants et les variables physiques du système ; sa construction ne nécessite aucune connaissance précise sur la nature des composants ou des relations entre les variables du modèle. Seule la connaissance d'interconnexions importe. L'Analyse Structurale peut ainsi être mise en œuvre très tôt dans le cycle de développement d'un système et permet donc au concepteur de rapidement disposer d'informations qui peuvent l'assister dans la conception des systèmes de diagnostic, dans ses choix d'instrumentation et d'architecture sur des critères objectifs de surveillabilité.

Objectifs. *Du point de vue industriel*, l'objectif de la thèse est de développer la méthodologie d'Analyse Structurale pour l'appliquer au diagnostic de systèmes pilotés dans l'automobile. Il s'agit de fournir une méthode outillée permettant d'assister les concepteurs dans la mise au point des systèmes de diagnostic : connaissance du potentiel de diagnostic, définition exhaustive des possibilités de tests, et étude de l'instrumentation. Au final, ces informations doivent permettre d'augmenter la couverture des diagnostics embarqués et débarqués. La méthode sera automatisée pour être appliquée sur les modèles de conception *Simulink* et sera plus particulièrement illustrée sur un modèle *Simulink* de boucle d'air du moteur *DV6C TED4*.

Du point de vue académique, le fil conducteur est l'application de la méthode à un système industriel très complexe. Les objectifs académiques portent ainsi en premier lieu sur la maîtrise et la gestion de l'explosion combinatoire des algorithmes de recherche exhaustive des tests de diagnostic. En effet, si l'on utilise les algorithmes actuels sur le modèle *Simulink* de boucle d'air du moteur *DV6C TED4*, 15 jours de calcul ne suffisent pas pour que ceux-ci se terminent. Dans ce mémoire, nous cherchons à atteindre cet objectif de maîtrise et de gestion de l'explosion combinatoire par l'intégration des non-inversibilités dans la méthode d'analyse structurale d'une part, et d'autre part, la prise en compte d'un cahier des charges de diagnostic. En second lieu, les objectifs académiques portent sur la génération des tests de diagnostic assistée par une analyse structurale.

Organisation du mémoire. Ce mémoire est structuré en deux grandes parties : la première, théorique, présente l'Analyse Structurale, et les algorithmes développés afin de réduire la complexité de la recherche exhaustive de tests fonctionnels ; la seconde, confidentielle, présente le travail réalisé sur la boucle d'air, et montre comment l'Analyse Structurale peut être utilisée pour assister la conception du système de diagnostic fonctionnel d'un système industriel complexe.

1. Première partie :

- (a) Au premier chapitre de ce mémoire, nous rappelons les fondements et les principes de l'Analyse Structurale, puis une brève étude bibliographique est présentée. La génération des tests de diagnostic assistée par l'Analyse Structurale est discutée ; nous faisons et justifions ici le choix d'une méthode particulière de génération des résidus. Celle-ci consiste à propager des variables connues dans un schéma de simulation. La

troisième partie, en faisant la synthèse des deux premières, exprime les problématiques soulevées par l'analyse structurelle de systèmes complexes et positionne notre travail ;

- (b) Le deuxième chapitre présente les algorithmes développés, en réponse aux problématiques scientifiques identifiées. Ces algorithmes visent à fournir, grâce à une analyse structurelle, les tests d'une part calculables (pour notre méthode de génération particulière), et d'autre part utiles (pour la discrimination des défauts). Ils permettent de maîtriser la complexité des algorithmes d'analyse structurelle existants, quand ils sont appliqués sur un modèle de grande taille ;
- (c) Le troisième chapitre présente l'outil logiciel développé, ses fonctionnalités, et son application à un exemple relativement simple : une vanne instrumentée prise comme benchmark du projet européen *DAMADICS* ;
- (d) Une conclusion ainsi que quelques perspectives sur les développements théoriques et applicatifs à poursuivre figurent au quatrième chapitre.

2. Seconde partie :

- (a) Au chapitre 5, le premier de la seconde partie, le système boucle d'air du moteur *DV6C TED4* est présenté ;
- (b) L'avant-dernier chapitre présente nos résultats sur le diagnostic fonctionnel de la boucle d'air. Ainsi, on utilisera les algorithmes développés au chapitre 2 pour déterminer l'ensemble des tests fonctionnels permettant d'effectuer le diagnostic de la boucle d'air ;
- (c) Le chapitre 7 conclut la seconde partie du rapport.

Introduction

Première partie

Diagnostic des systèmes complexes : algorithmes d'analyse structurelle pour la recherche des sous-systèmes surveillables

Organisation et contenu de la première partie. Cette première partie du mémoire est consacrée à la méthode utilisée : l'Analyse Structurelle, et aux contributions théoriques de cette thèse. Les algorithmes seront appliqués sur un exemple de vanne, choisi comme benchmark dans le cadre d'un projet européen (5^e FP EU RTN), sur le développement et l'application de méthodes pour le diagnostic d'actionneurs dans des systèmes de contrôle industriels (« Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems – *DAMADICS* »). Cette partie est organisée en trois chapitres :

1. Le premier chapitre présente l'Analyse Structurelle et les principaux travaux du domaine ; il précise de plus l'orientation prise par notre travail et les questions scientifiques traitées dans ce mémoire, qui sont issues des difficultés rencontrées lors de l'application de la méthode sur un modèle industriel complexe, de grande taille. Ces questions scientifiques portent principalement sur les aspects algorithmiques de l'Analyse Structurelle, c'est-à-dire sur la complexité des algorithmes ;
2. Le deuxième chapitre est dédié aux aspects algorithmiques de l'Analyse Structurelle ; y sont présentés des algorithmes développés afin de maîtriser la complexité. Deux idées sont explorées dans cette partie : l'intégration des non-inversibilités et la prise en compte d'un cahier des charges dans les algorithmes existants ;
3. Les algorithmes mentionnés au chapitre 2 ont été intégrés dans un outil logiciel, qui réalise une analyse structurelle des modèles décrits sous forme de schéma *Simulink*. Nous présenterons rapidement dans ce chapitre les fonctionnalités de cet outil, puis montrerons un cas d'application. L'exemple retenu pour illustrer le fonctionnement de l'outil est le modèle *Simulink* du benchmark *DAMADICS*.

Chapitre 1

Analyse Structurelle et Diagnostic

Résumé

Dans ce premier chapitre, les principes fondamentaux du diagnostic sont tout d'abord rappelés, puis l'analyse structurelle et son utilisation dans le diagnostic à base de modèle sont décrits. Dans la première partie, nous définissons donc le vocabulaire employé dans le domaine du diagnostic : les notions de résidu, de signature, d'isolabilité, etc.

Dans la deuxième partie, nous définissons dans un premier temps les concepts de l'analyse structurelle : graphe biparti, matrice d'incidence, couplage, boucle du graphe, etc. Dans un second temps, nous présentons les outils de la théorie des graphes utilisés : décomposition de Dulmage-Mendelsohn, algorithmes de recherche des générateurs potentiels de résidu.

Ensuite, nous précisons, parmi les différentes méthodes de génération des résidus, celle que nous allons employer : il s'agit d'une méthode de propagation des variables connues du système, en utilisant un simulateur.

Enfin, dans la dernière partie, les apports de la thèse sont résumés.

Abstract

In this first chapter, fundamental principles of model-based diagnosis are first recalled, then structural analysis and its place in model-based diagnosis are described.

In the first part, the vocabulary used in the field of diagnosis is defined, that is to say, notions of residual, of signature, of isolability, etc.

In the second part, concepts of structural analysis are defined : bipartite graph, incidence matrix, matching, loop, etc. On the other hand, the used graph theory tools are presented : the Dulmage-Mendelsohn decomposition and the algorithms for searching potential residual generators.

Then, among the different methods for residual generation, the one we will use is stated : it is a propagation method of known variables, using a simulator.

In the last part, the thesis' contributions are summarized.

Généralités et contenu du chapitre. L'analyse structurelle est une méthode d'analyse des modèles, permettant de caractériser certaines propriétés d'un système, comme l'observabilité, la commandabilité, ou la surveillabilité. Dans ce mémoire, l'analyse structurelle est utilisée pour la conception de systèmes de surveillance et s'inscrit dans le cadre du diagnostic à base de modèle, dont nous dirons pour l'instant qu'il consiste, dans son principe, à tester la cohérence des commandes et des mesures d'un système, en utilisant un modèle de celui-ci. Ces tests de cohérence indiquent si une faute s'est produite, informent sur la localisation de la faute, et, en cela, sont des indicateurs de défaut, aussi appelés résidus. Nous reviendrons sur ces notions à la section 1.1, qui présente les principes élémentaires du diagnostic à base de modèle.

Dans ce cadre de diagnostic à base de modèle, l'analyse structurelle est une méthode permettant d'*analyser la surveillabilité* des systèmes et de *guider la recherche d'indicateurs de défauts* : nous verrons en effet que, d'une part, elle met en évidence les fautes qu'il est impossible de diagnostiquer quelle que soit l'approche utilisée, et que, d'autre part, elle fournit différentes façons de construire des indicateurs de défauts. Présenter l'Analyse Structurelle, son intérêt et son utilisation, sera l'objet de la section 1.2.

Plusieurs formalismes d'analyse structurelle sont étudiés dans la communauté scientifique. En ce qui nous concerne, la méthode d'analyse structurelle sur laquelle nous travaillons s'appuie sur une description du modèle sous la forme d'un graphe biparti ((Declerck and Staroswiecki, 1991), (Blanke *et al.*, 2006)). Ce graphe représente la structure du modèle, c'est-à-dire la façon dont sont reliées les différentes variables du modèle, ou, dit autrement, la façon dont les variables du modèle sont contraintes entre elles par les relations du modèle de comportement. Ce graphe est dit biparti, car ses arcs lient deux types de sommets, le premier type représentant les variables du modèle, et le second représentant les relations, aussi appelées contraintes, du modèle.

L'analyse structurelle s'inscrit plus particulièrement dans la famille des méthodes qui utilisent des modèles structurels :

- la structure d'un système peut être représentée par un digraphe. L'observabilité et la commandabilité structurelles des systèmes linéaires sont étudiées dans (Dion *et al.*, 2003) en utilisant cette représentation. Ce modèle est aussi utilisé pour le placement de capteurs dans (Commault and Dion, 2007). Dans (Boukhobza and Hamelin, 2007), la représentation est étendue aux systèmes bilinéaires, et dans (Boukhobza *et al.*, 2006), aux systèmes descripteurs.
- la structure d'un système, ainsi que la façon dont se propagent les phénomènes, peut être représentée par un graphe causal ((Montmain and Gentil, 2000), (Gentil and Montmain, 2004)) ;
- la structure d'un système, enfin, peut être modélisée par un bond-graph, où sont également indiquées des informations sur la causalité ((Samantaray *et al.*, 2006), en ce qui concerne le diagnostic).

L'Analyse Structurelle est une méthode utilisée par la communauté Automatique, mais qui partage des similitudes avec l'approche du diagnostic pour les systèmes logiques, développée initialement dans (De Kleer and Williams, 1987). Des liens ont été mis en évidence entre ces deux approches grâce à un travail collectif du groupe IMALAIA (Intégration de modèles alliant Automatique et IA, (Cordier *et al.*, 2000), (Cordier *et al.*, 2004)). Le lecteur pourra se référer également à (Pulido and Gonzalez, 2004).

L'analyse du graphe biparti permet de déterminer des propriétés (structurelles) d'observabilité et de surveillabilité du système, de détectabilité et d'isolabilité des fautes. Cette analyse est indépendante de la nature des relations du modèle ; celles-ci peuvent être non-linéaires, qualitatives, ou sous forme de table. En effet, la construction du graphe biparti ne nécessite aucune connaissance précise sur la nature des relations ou la valeur des paramètres du modèle, aussi l'analyse structurelle peut être menée au moment de la conception du système, et fournir une aide au concepteur du diagnostic, à travers notamment :

- l'étude de la détectabilité et de l'isolabilité des fautes ;
- l'identification de possibles générateurs de résidu ;
- la détermination d'implantations de capteurs favorisant la surveillance.

Pour des systèmes complexes (pour des raisons de taille – nombre de contraintes linéaires, non linéaires, nombre de capteurs – et des raisons purement structurelles – interconnexions des contraintes), comme le cas d'application de cette thèse, une analyse structurelle complète fournit un très grand nombre de générateurs potentiels (structurellement admissibles) de résidu. Tous ne sont cependant pas réalisables en pratique, en raison, en particulier, de contraintes de calculabilité, d'inversion de fonctions, de résolution formelle ou numérique.

Un des objectifs de cette thèse est de tenir compte de ces contraintes d'implantation pour effectuer l'analyse structurelle, de manière à déterminer un nombre plus limité de générateurs de résidus, mais réellement implantables. Par ailleurs, dans bien des cas, il n'est pas nécessaire de rechercher tous les sous-systèmes surveillables d'un système complexe. Le cahier des charges (ou les spécifications) du système de diagnostic précise les éléments à surveiller en priorité (en fonction de critères de sûreté de fonctionnement). Le deuxième objectif de cette thèse est de tenir compte de ce cahier des charges lors de l'analyse structurelle. Ceci permet de limiter la recherche des générateurs de résidus aux seuls générateurs réellement nécessaires.

1.1 Principes généraux du diagnostic à base de modèles

Dans cette section, nous définirons quelques termes ayant trait au diagnostic des systèmes (1.1.1), puis présenterons les principes du diagnostic à base de modèle (1.1.2), en introduisant les notions de résidu structuré et de signature, de localisation d'une faute et d'isolabilité (1.1.3).

1.1.1 La terminologie du diagnostic

Nous rappelons dans cette section quelques définitions de termes utilisés dans le domaine du diagnostic des systèmes. Le comité technique de l'IFAC SAFEPROCESS a proposé des définitions (Isermann and Ballé, 1997), dans le but de créer une terminologie unifiée, que nous reprenons ici. En outre, une synthèse critique regroupant plusieurs sources peut se trouver sur le site internet (Ploix and Giuliani, 2010), ou figure dans le premier chapitre de (Touaf, 2005).

- **défaut ou faute (*fault*)** : déviation non permise d'au moins une propriété ou un paramètre caractéristique du système des conditions acceptables ou(et) standards (*an unpermitted deviation of at least one characteristic property or parameter of the system from acceptable / usual / standard condition*).

1.1. Principes généraux du diagnostic à base de modèles

- **défaillance (*failure*)** : interruption permanente de la capacité d'un système à assurer une fonction requise dans des conditions opérationnelles spécifiées (*a permanent interruption of a system's ability to perform a required function under specified operating conditions*).
- **panne (*break-down*)** : inaptitude d'un dispositif à accomplir sa fonction requise (Zwengelstein, 1995).
- **détection de défaut (*fault detection*)** : détermination des défauts présents dans un système et de leur instant d'apparition (*determination of the faults present in a system and the time of detection*).
- **localisation de défaut (*fault isolation*)** : détermination de la nature, de l'endroit et de l'instant d'apparition d'un défaut. Suit la détection (*determination of the kind, location and time of detection of a fault. Follows fault detection*).
- **identification de défaut (*fault identification*)** : détermination de la taille et de l'évolution d'un défaut. Suit la localisation (*determination of the size and time-variant behaviour of a fault. Follows fault isolation*).
- **diagnostic (*diagnosis*)** : détermination du type, de la taille, de l'endroit et de l'instant d'apparition d'un défaut. Suit la détection de défaut. Inclut la localisation et l'identification d'un défaut (*determination of the kind, size, location and time of a fault. Follows fault detection. Includes fault isolation and identification*).
- **symptôme (*symptom*)** : caractère distinctif d'un état fonctionnel anormal (Ploix, 1998).
- **résidu (*residual*)** : indicateur de défauts basé sur la différence entre des mesures et des résultats de calcul à partir d'équations de modèle (*a fault indicator, based on a deviation between measurements and model-equation-based computations*).
- **perturbation (*disturbance*)** : une entrée inconnue agissant sur un système (*an unknown (and uncontrolled) input acting on a system*).

Le lecteur trouvera de plus, dans (Brunet *et al.*, 1990), des classifications des défauts, suivant leur caractère (aléatoire ou progressif) ou selon leur origine (capteurs, organes de commande, ...).

La figure 1.1, d'après (Touaf, 2005), classe les anomalies (défaut, défaillance, panne) selon leur criticité.

1.1.2 Le diagnostic à base de modèle

On distingue en général les méthodes de diagnostic à base de modèle – qui utilisent un modèle mathématique du procédé à surveiller – des méthodes dites sans modèle – fondées sur l'analyse statistique des signaux ou des observations. Dans ce deuxième cas, que nous ne développerons pas, un modèle est utilisé également, mais il ne s'agit pas d'un modèle physique, analytique. Un système de diagnostic comporte trois fonctions :

- une fonction de détection du défaut ;
- une fonction de localisation de l'élément défaillant ;

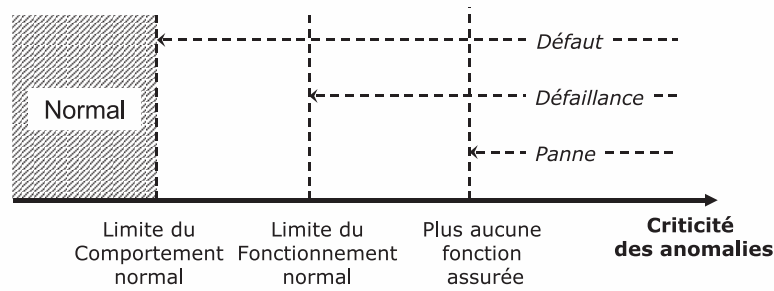


FIGURE 1.1 – Anomalies classées par criticité croissante.

- une fonction d'identification du défaut.

Pour les systèmes de diagnostic fondés sur un modèle analytique du processus à surveiller, les fonctions de détection et de localisation reposent sur la recherche d'incohérences entre l'évolution théorique des variables contraintes par le modèle, et l'évolution réelle du système. Des tests sont réalisés en générant et en évaluant des quantités appelées résidus. Un résidu est une grandeur fonction de variables connues (mesures et commandes) qui quantifie la cohérence entre les observations compte-tenu du modèle du système surveillé. Un résidu peut être par exemple la différence entre une valeur reconstruite et une valeur mesurée. Si le résidu est nul, il y a cohérence entre les observations et le modèle du système ayant servi à construire l'estimation. Si le résidu est non nul, on suppose que cette incohérence est le résultat d'une faute d'un composant ou d'une fonction du système : le résidu sert d'indicateur de défauts, il informe sur l'apparition d'une faute et sur sa localisation (voir par exemple (Chen and Patton, 1999), (Patton *et al.*, 2000), (Düstegör, 2005), (Ploix *et al.*, 2003)). Localiser un défaut nécessite plusieurs résidus ; en recoupant les informations données par plusieurs résidus, on localise plus précisément le défaut.

Génération des résidus. Les trois méthodes classiques de génération de résidus sont :

1. les méthodes par estimation utilisant des observateurs ou des estimateurs ;
2. les méthodes utilisant des relations de redondance analytique (RRA). Dans le cas des modèles linéaires, les RRA sont obtenues par projection dans l'espace de parité ;
3. les méthodes d'identification paramétrique.

Toutes ces méthodes sont présentées dans (Brunet *et al.*, 1990). Pour plus d'informations concernant les méthodes utilisant des observateurs ou de projection dans l'espace de parité, on pourra également se reporter à (Staroswiecki, 2000) ou (Maquin *et al.*, 1997). Dans (Izadi-Zamanabadi, 1999), (Djeziri *et al.*, 2009), (Kallesoe *et al.*, 2006) sont présentées des applications de méthodes utilisant des RRA (sur, respectivement, un système de propulsion de bateau, un générateur de vapeur, et une pompe centrifuge). Une relation de redondance analytique peut être continue ou discrète. Dans le cas continu, la définition est donnée ci-dessous.

Définition 1.1.1 (Relation de redondance analytique continue). Une relation de redondance analytique est une relation ne faisant intervenir que des paramètres et des variables connues z (commandes et mesures), ainsi que leurs dérivées successives : $f(z, \dot{z}, \dots) = 0$. Elle ne sera vérifiée que si le comportement du système coïncide avec le modèle utilisé pour obtenir cette relation.

Une relation de redondance analytique $f(z, \dot{z}, \dots) = 0$ (z représentant des commandes et des mesures), obtenue à partir du modèle, fournit donc un résidu $r = f(z_r, \dot{z}_r, \dots)$ (z_r étant les signaux de commande et les mesures provenant du système réel). Si le comportement du système coïncide avec le modèle, r doit être nul, autrement dit la relation de redondance doit être vérifiée. Dans le cas contraire, lors de l'apparition d'un défaut ou d'une perturbation, le résidu r sera différent de 0.

Décision, alarme, et évaluation statistique des résidus. En l'absence d'erreurs de modélisation ou de bruits de mesure, un résidu sera nul lorsque le système est en fonctionnement normal, et non nul en présence d'une panne. Il suffit d'un seul résidu non nul pour détecter un défaut et déclencher une alarme.

Néanmoins, à cause d'erreurs de modélisation, d'identification imparfaite des paramètres, ou de bruits de mesure, un résidu n'est jamais strictement nul ; une procédure de décision doit être implantée afin de décider si la valeur du résidu s'explique par des erreurs de modélisation ou des bruits de mesure, ou par un défaut d'un composant du système. Estimer la nullité ou la non-nullité d'un résidu est un problème statistique : on parlera d'évaluation statistique des résidus, ou encore de décision statistique. Cette étape de décision statistique, qui détermine si une alarme doit être déclenchée ou non, ne sera pas détaillée dans la suite de ce mémoire. Le lecteur intéressé pourra se référer aux ouvrages (Basseville and Nikiforov, 1993), (Brunet *et al.*, 1990).

Diagnostic. Le principe du diagnostic à base de modèle est donné figure 1.2 : les signaux provenant du procédé sont filtrés et sont utilisés pour générer plusieurs résidus, dont le mode de calcul diffère, ainsi qu'éventuellement, leur sensibilité aux fautes. Un résidu évalué statistiquement non nul informe sur la présence d'un défaut et sur sa localisation. Le module de diagnostic – qui comprend la génération des résidus, leur évaluation statistique, et la procédure de localisation – fournit au final un diagnostic du système surveillé.

Dans la section suivante (1.1.3), nous allons détailler le fonctionnement du module de localisation, car cela sera nécessaire pour la compréhension du chapitre 2 de ce mémoire. Nous avons brièvement discuté des méthodes de génération des résidus. Nous reviendrons sur ces méthodes dans la section 1.3.1 pour évoquer les problèmes posés lorsque le modèle traité est complexe (c'est-à-dire, comporte de nombreuses fonctions interconnectées) et représenté sous forme d'un schéma de simulation *Simulink*, comme c'est le cas pour nos applications industrielles. Nous y détaillerons une méthode de génération de résidus spécifique, qui utilise directement les schémas de simulation, en propageant les signaux d'entrée et de sortie.

1.1.3 Localisation d'un défaut

Il existe plusieurs façons de concevoir le module de localisation, cependant toutes exploitent la sensibilité théorique des résidus aux fautes. En effet, un résidu synthétisé (par exemple, à partir des méthodes discutées à la section précédente : observateurs, espace de parité...) est théoriquement sensible à un sous-ensemble de fautes : on parle habituellement de résidus structurés.

Définition 1.1.2 (Structure d'un résidu). On appelle structure d'un résidu l'ensemble des fautes

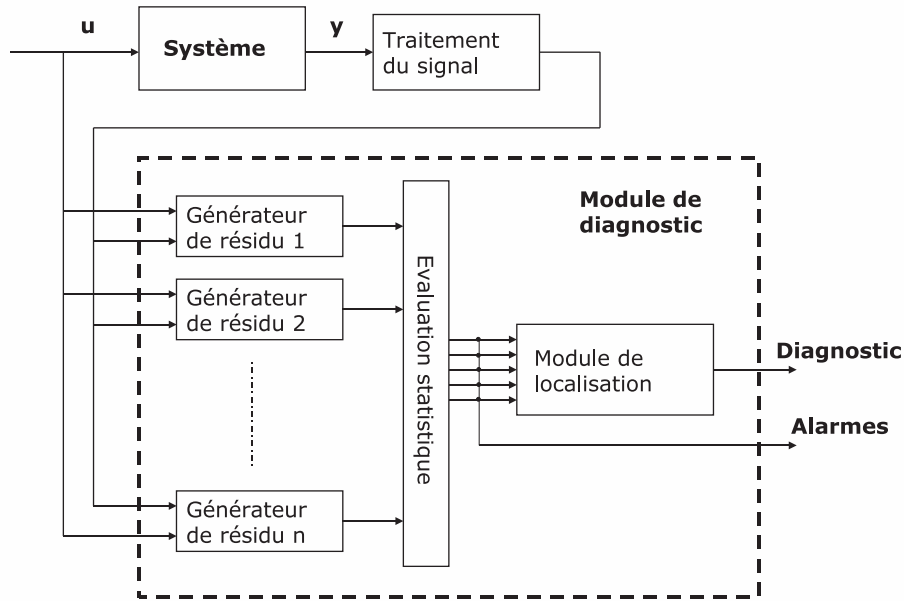


FIGURE 1.2 – Principe du diagnostic à base de modèle

auxquelles ce résidu est potentiellement sensible, c'est à dire l'ensemble des fautes dont l'apparition peut le rendre non nul.

Cette sensibilité est potentielle, ou théorique, ou encore structurelle. Elle est déterminée lors de la conception des générateurs de résidus, avant que les résidus aient été générés. Dans certains cas, la sensibilité réelle peut être étudiée : par exemple, dans le cas d'une méthode par RRA, par l'analyse de la forme d'évaluation de la RRA (Staroswiecki and Comtet-Varga, 2001).

La connaissance de la sensibilité structurelle des générateurs de résidus permet d'obtenir des signatures de fautes, une table de signature des fautes, puis une table d'isolabilité structurelle. Ces tables sont à valeurs booléennes.

Définition 1.1.3 (Table de signature). La table de signature des fautes est la table binaire qui regroupe les structures de tous les résidus. Un « 1 » à la position (i, j) indique que le résidu « r_i » est affecté par la faute « f_j ».

Exemple 1.1.1. Table de signature : ici, le résidu r_1 est potentiellement sensible aux fautes f_1 et f_2 ; le résidu r_2 est sensible aux fautes f_1 , f_3 et f_4 .

TABLE 1.1 – Exemple de table de signature

	f_1	f_2	f_3	f_4
r_1	1	1		
r_2	1		1	1

Définition 1.1.4 (Signature). La signature théorique de la faute f_i est le i^{ieme} vecteur colonne

de la table de signature du système. Elle est notée $\sigma(f_i)$ et traduit l'influence théorique de la faute f_i sur les résidus.

Définition 1.1.5 (Signature réelle). La signature réelle σ_r d'un système est le vecteur colonne traduisant la vérification ou la non-vérification de la relation $r_i = 0$. Un « 1 » à la position i indique que le résidu « r_i » est évalué non nul.

Exemple 1.1.2. Signatures théoriques et réelle : notons $\sigma(f_i)$ la signature de la faute f_i . En reprenant l'exemple 1.1.1, pour chacune des quatre fautes, nous avons les signatures suivantes : $\sigma(f_1) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $\sigma(f_2) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\sigma(f_3) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $\sigma(f_4) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Supposons que r_1 soit évalué non nul et r_2 soit évalué nul, la signature réelle est alors le vecteur $v = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

Nous avons dit, dans la première phrase de cette section, qu'il y avait plusieurs façons de concevoir le module de localisation ; en d'autres termes, plusieurs logiques de localisation permettent, après évaluation des résidus, de déterminer, sur l'exemple 1.1.1 si, en présence d'un défaut, la défaillance est due à f_1 , f_2 , ou f_3 . On citera dans ce mémoire une approche « logique » issue de la communauté IA et une approche par comparaison des signatures de fautes, utilisée dans la communauté FDI, en utilisant une distance entre signatures. Les différences et les recouvrements entre ces méthodes sont établis dans (Cordier *et al.*, 2000) et reposent sur la notion d'exonération, implicite pour la communauté FDI, et sur la prise en compte des fautes multiples ou non. Un système de diagnostic construit sous l'hypothèse d'exonération est un système qui exonère les composants qui interviennent dans une RRA vérifiée, en ce sens qu'il suppose leur bon fonctionnement.

Nous nous appuyons dans ce mémoire sur l'approche par comparaison des signatures de fautes. En ligne, le module de localisation reçoit les alarmes déclenchées par certains résidus : un résidu (évalué statistiquement) non nul permet de détecter un défaut (incohérence entre le modèle et les observations), et donc de déclencher une alarme. Un tel résidu informe sur la localisation de la faute car il n'est sensible qu'à certaines fautes et pas à d'autres. Une façon de localiser la faute est de calculer des distances entre la signature réelle fournie par le système, et les signatures théoriques de faute, calculées hors ligne (Cassar *et al.*, 1994).

Définition 1.1.6 (Distance de Hamming). La distance de Hamming entre deux vecteurs x et y est la valeur $d(x, y) = \sum_{i=1}^n x_i \oplus y_i$. ($\oplus$ étant l'opérateur *XOR*).

La procédure de localisation consiste à calculer successivement les distances entre la signature réelle et chaque signature théorique, et à expliquer le comportement du système par la faute f_k , où :

$$k = \operatorname{argmin}(d(\sigma_r, \sigma(f_i)))$$

Il s'agit, en d'autres termes, de décider quelle est la faute dont l'effet théorique sur les résidus structurés se rapproche le plus de l'effet observé.

Exemple 1.1.3. Localisation : Reprenons l'exemple 1.1.2. Supposons toujours que r_1 soit évalué non nul et r_2 soit évalué nul. Nous avons $d(v, \sigma(f_1)) = 1$, $d(v, \sigma(f_2)) = 0$, $d(v, \sigma(f_3)) = 2$, $d(v, \sigma(f_4)) = 2$. Par conséquent, la faute dont la signature se rapproche le plus de la signature

réelle est f_2 .

Supposons maintenant que r_1 soit évalué nul et r_2 soit évalué non nul. La signature réelle est alors le vecteur $v = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

Nous avons $d(v, \sigma(f_1)) = 1$, $d(v, \sigma(f_2)) = 2$, $d(v, \sigma(f_3)) = 0$, $d(v, \sigma(f_4)) = 0$. Par conséquent, il existe deux fautes dont la signature se rapproche le plus de la signature réelle : f_3 et f_4 . Dans ce cas, il n'est pas possible de localiser précisément le défaut ; on dit que f_3 et f_4 ne sont pas isolables.

Définition 1.1.7 (Isolabilité de deux fautes). Deux fautes ne sont pas isolables si leurs signatures sont identiques. Deux fautes sont isolables si leurs signatures diffèrent.

La table d'isolabilité des défauts synthétise les informations sur l'isolabilité des défauts :

Définition 1.1.8 (Table d'isolabilité). La table d'isolabilité T des défauts $f_1, \dots, f_n$ est la matrice binaire définie par :

- $T(i, j) = 1$ si f_i, f_j ne sont pas isolables ;
- $T(i, j) = 0$ si f_i, f_j sont isolables.

On peut quantifier l'isolabilité de deux fautes par la distance de Hamming entre leur signature (Cassar *et al.*, 1994) : plus la distance de Hamming entre les signatures est grande, plus on est préservé contre les erreurs d'évaluation des résidus, qui causent des fausses alarmes (une relation de redondance est évaluée non vérifiée sans raison) et, à l'inverse, des non-détections.

Exemple 1.1.4. Isolabilité : sur l'exemple 1.1.1, à la page 20, f_3 et f_4 ont la même signature, et ne sont donc pas isolables. En revanche f_1 et f_2 sont isolables. De même, f_1 et f_3 , f_1 et f_4 , f_2 et f_3 , f_2 et f_4 sont isolables. La table d'isolabilité de ces défauts est par conséquent :

TABLE 1.2 – Table d'isolabilité de l'exemple 1.1.1

	f_1	f_2	f_3	f_4
f_1	1			
f_2		1		
f_3			1	1
f_4			1	1

Nous avons par ailleurs les distances de Hamming suivantes entre fautes : $d(f_1, f_2) = 1$, $d(f_1, f_3) = d(f_1, f_4) = 1$, $d(f_2, f_3) = d(f_2, f_4) = 2$. En conséquence, l'isolabilité entre f_2 et f_3 est plus forte que l'isolabilité entre f_1 et f_3 , par exemple. En effet, si le système est soumis à la seule faute f_3 , il suffit d'une erreur d'évaluation des résidus pour diagnostiquer, à tort, une faute f_1 , tandis qu'il en faut deux pour diagnostiquer, toujours à tort, une faute f_2 .

1.2 Les fondements de l'Analyse Structurale

L'Analyse Structurale repose sur un modèle structurel, sous la forme d'un graphe biparti, obtenu à partir d'un modèle de comportement du système, dont les paramètres ne sont pas nécessairement identifiés. L'analyse structurale permet de :

1. analyser les propriétés de surveillabilité du système, c'est-à-dire identifier, sur le système global, la partie qu'il est possible de surveiller, et celle qui à l'inverse n'est pas surveillable. Pour ce faire, on utilise la décomposition de Dulmage-Mendelsohn (Dulmage and Mendelsohn, 1958) du graphe qui détermine le sous-ensemble surveillable, sur lequel il est possible de générer des résidus ;
2. aider à la recherche des générateurs de résidus. Les générateurs de résidus correspondent à un sous-graphe du modèle structurel, nommé sous-système surveillable.

Le diagnostic à base de modèle repose sur la génération de résidus, dont les méthodes les plus employées sont les méthodes à base d'observateur ou de l'espace de parité. L'Analyse Structurelle ne permet pas, quant à elle, de générer à proprement parler des résidus. Elle précède la conception des générateurs des résidus et n'a qu'une vocation d'*analyse* des systèmes : elle permet d'identifier les sous-parties du modèle sur lesquelles des résidus peuvent potentiellement être générés et peut guider la recherche des RRA. Générer effectivement les tests nécessite en général d'autres techniques comme l'élimination formelle de variables dans un système d'équations (bases de Gröbner, (Guernez *et al.*, 1997), (Frisk, 2000), (Maquin *et al.*, 1997)), ou la synthèse d'observateurs (Svärd and Wassén, 2006). En ce sens, l'Analyse Structurelle peut être couplée aux méthodes classiques de diagnostic à base de modèle, dans une phase amont, et leur est complémentaire.

1.2.1 Matrice d'incidence et graphe biparti

L'Analyse Structurelle repose sur la connaissance des liens entre les variables du système, et non sur la valeur des paramètres. Elle permet de traiter par conséquent des modèles « cartographiés » ou « boîte noire ». . . Cette connaissance est représentée sous forme d'un graphe biparti ou de manière équivalente sous forme d'une matrice d'incidence binaire.

Définition 1.2.1 (Graphe biparti). Un graphe biparti est un triplet (V_1, V_2, E) où V_1, V_2 sont les ensembles des sommets du graphe et $E \subset V_1 \times V_2$ est l'ensemble des arêtes. (V_1, V_2) forme une partition en deux ensembles (ou bipartition) de l'ensemble des sommets.

Soit un système décrit par un ensemble C de contraintes (ou relations) liant des variables entre elles, et V l'ensemble de ces variables. Soit $V = X \cup Z$ la partition de l'ensemble des variables V en deux ensembles : celui des variables connues (commandes et mesures) Z et celui des variables inconnues X .

Définition 1.2.2 (Graphe biparti du modèle). La structure du système est modélisée par un graphe biparti (C, V, Γ) où Γ est défini comme suit : il existe un arc entre une contrainte c_i et une variable v_j si et seulement si v_j apparaît dans c_i .

La figure 1.3 présente un exemple de graphe biparti : les éléments de C (sommets-contraintes) sont représentés par des traits, tandis que les éléments de V (sommets-variables) sont représentés par des points. La contrainte c_1 , par exemple, lie la variable connue z_1 et la variable inconnue x_1 , ce qui est représenté grâce aux arcs (c_1, z_1) et (c_1, x_1) .

Définition 1.2.3 (Matrice d'Incidence). La matrice d'incidence du graphe biparti est la matrice S définie de la façon suivante :

$$\begin{cases} S(i, j) = 1 & \text{si } v_j \in V \text{ apparaît dans } c_i, \\ S(i, j) = 0 & \text{si } v_j \in V \text{ n'apparaît pas dans } c_i. \end{cases}$$

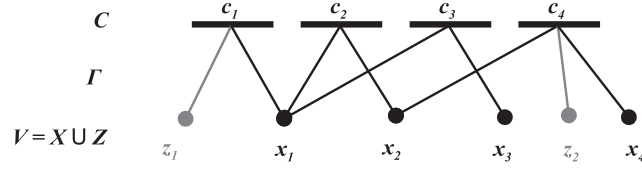


FIGURE 1.3 – Exemple de graphe biparti

Représentation des dérivées. Dans les modèles dynamiques, des variables dérivées interviennent. Il y a deux façons de représenter les dérivées par rapport au temps dans un modèle structurel :

1. on considère que x et $\dot{x}$ sont structurellement la même variable, et l'on traite par conséquent les équations dynamiques comme des équations statiques ;
2. on considère que x et $\dot{x}$ sont deux variables structurelles différentes, et l'on indique le lien entre ces variables en définissant un ensemble de contraintes supplémentaires, appelées contraintes différentielles. Il s'agit de la représentation que nous adopterons dans ce mémoire, car elle est plus précise et plus facile à interpréter, la notion de causalité y apparaissant mieux.

Définition 1.2.4 (Contrainte différentielle). Une contrainte différentielle est une contrainte de la forme $y = \frac{dx}{dt}$, y et x étant considérées comme deux variables distinctes.

Définition 1.2.5 (Variable intégrale, variable dérivée). Pour une contrainte différentielle $y = \frac{dx}{dt}$, x est appelée variable intégrale y , variable dérivée.

Exemple 1.2.1. La matrice d'incidence du circuit RC de la figure 1.4, décrit par les équations du système 1.1 ci-après :

$$\left\{ \begin{array}{l} c_1 : u_R = Ri \\ c_2 : q = Cu_C \\ c_3 : E = u_R + u_C \\ c_4 : i = \frac{dq}{dt} \\ c_5 : i_m = i \\ c_6 : u_{R,m} = u_R \end{array} \right. \quad (1.1)$$

est donnée par la matrice de la table 1.3 ci-après, dont les variables ont été scindées en deux ensembles : l'ensemble des variables inconnues (q , i , u_R , u_C) et celui des variables connues (E , $u_{R,m}$ et i_m). Nous avons ajouté la contrainte différentielle c_4 modélisant le lien entre la charge q et l'intensité i .

Le graphe biparti du modèle structurel du circuit RC est représenté à la figure 1.5. Les contraintes y sont représentées par des barres, et les variables par des points, soit noirs, s'il s'agit de variables inconnues, soit gris, s'il s'agit de variables connues.

Souvent, dans la suite de ce rapport, on restreindra la représentation aux variables inconnues : nous travaillerons sur le graphe (C, X, Γ') où Γ' est défini comme suit : il existe un arc entre une contrainte c_i et une variable inconnue x_j si et seulement si x_j apparaît dans c_i .

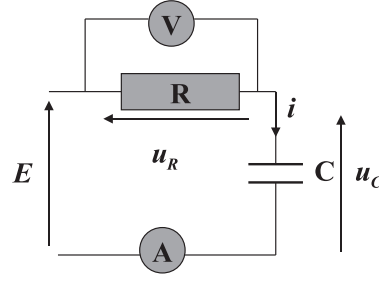


FIGURE 1.4 – Exemple : circuit RC

TABLE 1.3 – Matrice d'incidence du modèle 1.1

	q	i	u_R	u_C	E	$u_{R,m}$	i_m
c_1		1	1				
c_2	1			1			
c_3			1	1	1		
c_4	1	1					
c_5		1					1
c_6			1			1	

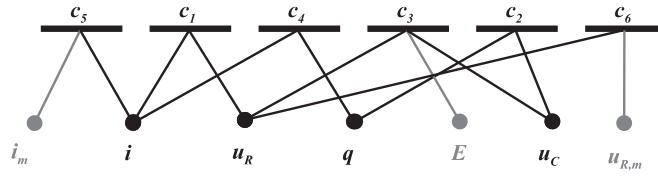


FIGURE 1.5 – Exemple : graphe biparti du modèle structurel du circuit RC

1.2.2 Modélisation structurelle des défauts

Dans (Düstegör *et al.*, 2004d), trois niveaux de connaissance des défaillances sont considérés.

- le niveau de connaissance le plus élémentaire consiste à indiquer les équations du modèle (contraintes) qui ne sont plus valides en cas de défaillances. Une défaillance correspond au viol d'une contrainte (loi physique) ;
- le deuxième niveau de connaissance consiste à décrire, grâce à des variables supplémentaires (dites de défaillance) la façon dont sont modifiées les équations de fonctionnement normal lorsqu'une défaillance survient. Les défaillances peuvent être additives ou multiplicatives suivant la manière dont les variables de défaillance influent sur les équations du modèle ;
- le troisième niveau de connaissance consiste à modéliser l'évolution dynamique de la défaillance. Des équations supplémentaires liant les variables de défaillance sont ajoutées au modèle de bon fonctionnement, qui sont obtenues soit par la connaissance des phénomènes physiques mis en jeu, soit à partir de données expérimentales du processus défectueux.

Sur l'exemple du circuit RC qui précède, une défaillance de l'appareil de mesure du courant peut être décrite comme le viol de la contrainte $c_5 : i_m = i$ – c'est le premier niveau de connais-

sance de la défaillance – ou bien grâce à une variable de faute f venant modifier la relation c_5 : $c_5 : i_m = i + f$, ce qui constitue le deuxième niveau de connaissance. Le troisième niveau de connaissance consiste par exemple à exprimer la dynamique de la défaillance par l'ajout d'une relation. Par exemple, si l'expérience nous montre que le défaut du capteur se traduit par un biais de mesure, l'ajout de la relation $\dot{f} = 0$ permet de le modéliser.

1.2.3 Décomposition canonique

La décomposition de Dulmage-Mendelsohn (Dulmage and Mendelsohn, 1958) permet de décomposer le sous-graphe liant les variables inconnues en trois parties : la partie sur-contrainte S^+ , la partie juste-contrainte S^0 , et la partie sous-contrainte S^- (voir Figure 1.6), dont l'interprétation a été donnée dans (Declerck and Staroswiecki, 1991) :

- $S^+ = (C^+, X^+, E^+)$ est la partie « surveillable » du système ;
- $S^0 = (C^0, X^0, E^0)$ est la partie « observable » du système ;
- $S^- = (C^-, X^-, E^-)$ est la partie non surveillable, non observable.

Cette décomposition est obtenue, par rapport à la matrice initiale, en faisant une permutation des lignes et une autre des colonnes ; elle est unique et induit des partitions des contraintes et des variables inconnues :

1. (C^+, C^0, C^-) est une partition de C , où :
 - C^+ est l'ensemble des contraintes du sous-graphe S^+ ;
 - C^0 est l'ensemble des contraintes du sous-graphe S^0 ;
 - C^- est l'ensemble des contraintes du sous-graphe S^- .
2. (X^+, X^0, X^-) est une partition de X , avec :
 - X^+ est l'ensemble des variables inconnues du sous-graphe S^+ ;
 - X^0 est l'ensemble des variables inconnues du sous-graphe S^0 ;
 - X^- est l'ensemble des variables inconnues du sous-graphe S^- .

Par ailleurs, on notera que la matrice ainsi décomposée est triangulaire inférieure par blocs. C'est dans la partie surveillable, ou sur-contrainte, que l'on trouve les redondances nécessaires pour la génération de résidus.

A des fins de diagnostic, on ne s'intéressera donc qu'à la partie sur-contrainte, S^+ , du système.

1.2.4 Recherche des sous-systèmes surveillables

Une analyse structurelle permet de trouver les ensembles de relations à partir desquels on peut construire un test de diagnostic. On appelle ces ensembles les sous-systèmes surveillables, mais d'autres noms sont donnés dans la littérature : ensembles « Minimal Structurally Overdetermined » (MSO), « Possible Conflicts » (PC), « Sous-Systèmes Testables » (SST), « Sous-systèmes surveillables ».

Définition 1.2.6 (Sous-système surveillable, MSO). Un ensemble de relations M est un sous-système surveillable si et seulement si il vérifie $M^+ = M$ et $|M| - |\text{var}_X(M)| = 1$. Cet ensemble de relations est également appelé « Minimal Structurally Overdetermined ».

Dans cette définition, on note :

- $\text{var}_X(M)$ l'ensemble des variables inconnues apparaissant dans les relations de M ;

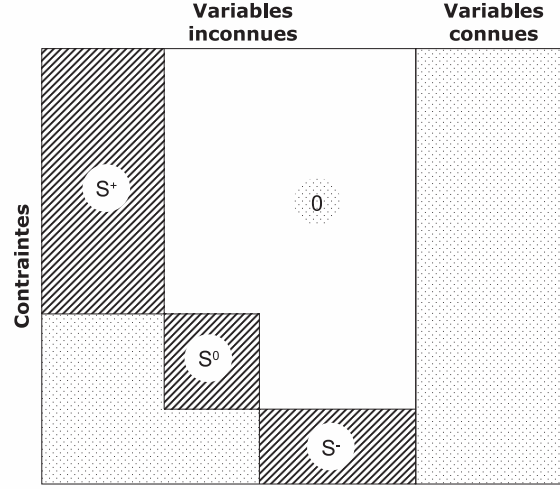


FIGURE 1.6 – Décomposition Canonique de Dulmage-Mendelsohn

– $|\cdot|$ représente le cardinal d'un ensemble.

Exemple 1.2.2. Reprenons l'exemple 1.2.1 Le système est entièrement surcontraint. Par conséquent, $S^+ = S$, et $S^0 = S^- = \phi$. Il existe quatre sous-systèmes surveillables sur le circuit RC :

1. $M_1 = \{c_1, c_5, c_6\}$;
2. $M_2 = \{c_1, c_2, c_3, c_4, c_5\}$;
3. $M_3 = \{c_1, c_2, c_3, c_4, c_6\}$;
4. $M_4 = \{c_2, c_3, c_4, c_5, c_6\}$.

Ces ensembles sont sur-contraints, donc surveillables, ce qui signifie qu'un test de diagnostic peut être construit sur chaque ensemble pour vérifier que les contraintes de l'ensemble sont ou non défaillantes. Les algorithmes qui permettent de rechercher l'exhaustivité de ces sous-systèmes sont récapitulés et présentés, dans leur principe, dans (Armengol *et al.*, 2009). Citons quelques un de ces algorithmes :

- l'algorithme « MSO » (Krysander *et al.*, 2008), dont la complexité, pour un nombre de contraintes $n = |C^+|$, un nombre de variables inconnues $m = |V^+|$, et un degré de redondance structurel $d (= n - m)$, est en $\mathcal{O}(n^{d+1,5})$;
- l'algorithme « Combination of Basic MSO » (CBMSO, (Gelso *et al.*, 2008)) ;
- l'algorithme « Structural Analytical Redundancy Relations » (SARR, (Travé-Massuyès *et al.*, 2006)) ;
- l'algorithme « Possible Conflicts Computation » (PCC, (Pulido and Gonzalez, 2004)) ;
- l'algorithme de recherche exhaustive des sous-systèmes surveillables (Düstegör, 2005).

Les différences entre ces algorithmes tiennent principalement dans leur complexité, récapitulée dans (Armengol *et al.*, 2009). Citons également le « Ranking Algorithm » de (Blanke *et al.*, 2006), mais celui-ci ne fournit pas l'exhaustivité des sous-systèmes surveillables. L'inconvénient de cet algorithme est par conséquent que les relations trouvées ne permettent pas de couvrir certaines parties du système que l'on souhaiterait pourtant surveiller.

Afin de qualifier le potentiel discriminant des sous-systèmes surveillables, nous introduisons la définition suivante :

Définition 1.2.7 (Sous-système surveillable non discriminant). Soit un ensemble de fautes $f_1, \dots, f_n$. Le sous-système surveillable M est non discriminant pour les fautes dans deux cas :

- M est structurellement sensible à toutes les fautes ;
- M n'est structurellement sensible à aucune faute.

1.2.5 Calcul a posteriori de la table d'isolabilité structurelle.

Exemple 1.2.3. Reprenons l'exemple 1.2.2. Les quatre sous-systèmes surveillables trouvés permettent de construire une table d'isolabilité structurelle des fautes. Comme précisé à la section 1.2.2, un premier niveau de modélisation des défauts est de considérer que chaque contrainte peut être violée. Pour cet exemple, nous adoptons ce premier niveau de modélisation : chaque contrainte $c_1, c_2, c_3, c_4, c_5, c_6$ peut être violée et constitue une faute potentielle. La sensibilité structurelle des quatre sous-systèmes aux fautes est donc donnée par la table suivante (1.4) :

TABLE 1.4 – Table de signatures du circuit RC

	c_1	c_2	c_3	c_4	c_5	c_6
M_1	1				1	1
M_2	1	1	1	1	1	
M_3	1	1	1	1		1
M_4		1	1	1	1	1

Si l'on lit cette table de signatures dans le sens vertical, nous avons les signatures des « fautes » $c_1, c_2, c_3, c_4, c_5, c_6$, et l'isolabilité s'en déduit d'après les définitions 1.1.7 et 1.1.8 (table 1.5) :

TABLE 1.5 – Table d'isolabilité du circuit RC

	c_1	c_2	c_3	c_4	c_5	c_6
c_1	1					
c_2		1	1	1		
c_3		1	1	1		
c_4		1	1	1		
c_5					1	
c_6						1

Les fautes c_2, c_3, c_4 ont la même signature théorique, donc sont structurellement non isolables.

1.3 Méthode de génération des résidus

Dans cette section, nous revenons en premier lieu sur le contexte, les objectifs et les problématiques abordées dans cette thèse ; ces dernières nous conduisent à faire le choix d'une méthode de

génération des résidus, exposée en 1.3.1. Ensuite, la notion de couplage, dont nous nous servons pour construire nos générateurs de résidus, est introduite (1.3.2).

1.3.1 Choix d'une méthode de génération de résidus

Notre objectif, d'un point de vue industriel, est de proposer et d'implanter une méthode de construction de tests de diagnostic, sur un système modélisé sous forme d'un schéma *Matlab/Simulink*, complexe de par :

- le nombre de relations (blocs « *Simulink* ») qui le composent (2887 relations) ;
- le nombre de capteurs implantés (11) ;
- la précision de la modélisation, qui conduit à un modèle avec des interactions très fortes entre sous-systèmes, interactions qui n'existeraient pas sur un modèle plus simple ;
- la nature des relations : les blocs constituant le modèle sont des fonctions non-linéaires, non continues, des saturations, des tables, des cartographies, des fonctions logiques . . .

Le modèle sera présenté plus en détail à la section ?? . Mener une analyse structurelle est complexe, et sera traitée par le développement d'algorithmes d'analyse structurelle prenant en compte des critères de calculabilité des résidus, ainsi qu'un cahier des charges (chapitre 2).

Etant donné les spécificités de notre modèle (à savoir, qu'il est codé dans un logiciel de simulation, qu'il est non-linéaire, et qu'il comporte des fonctions non-analytiques), nous faisons le choix de calculer des résidus au moyen d'un logiciel de simulation : les informations provenant des capteurs (les mesures) et celles provenant des organes de contrôle (les commandes) sont propagées à travers un schéma de simulation. Cette méthode de calcul répond aux spécificités de notre modèle, car les logiciels de simulation permettent de traiter les fonctions non-analytiques, non-linéaires . . .

Nos résidus sont ainsi obtenus en testant la cohérence entre des grandeurs calculées en simulation et d'autres mesurées sur le système en fonctionnement. Cette méthode de génération des résidus est généralement adoptée dans la communauté IA, voir par exemple (Pulido and Gonzalez, 2004), ou (Svärd and Wassén, 2006). Les questions posées par la simulation concernent, outre la robustesse aux perturbations, la connaissance des conditions initiales.

- Dans (Pulido *et al.*, 2008), les auteurs abordent la question des conditions initiales en plaçant un estimateur de conditions initiales en amont des générateurs de résidus (simulés). Ainsi, un observateur fournit aux schémas de simulation de la forme d'état les conditions initiales dont ils ont besoin ;
- Dans (Calderón-Espinoza *et al.*, 2007), les auteurs considèrent des conditions initiales bornées et utilisent des techniques d'estimation par intervalles. L'incertitude sur les conditions initiales est ainsi propagée à travers le schéma de simulation, jusqu'au résidu.

1.3.2 Aide à la conception des générateurs de résidu

Les sous-systèmes surveillables sont des sous-systèmes sur-contraints, et pour cette raison, il est possible de les utiliser pour générer des résidus. La construction de générateurs de résidus repose sur le concept de couplage d'un graphe biparti, concept qui est introduit dans cette section. Il est en effet possible d'utiliser la connaissance des liens entre variables, donnée par

le graphe biparti, pour identifier un réordonnancement des équations favorisant le calcul des variables inconnues. La notion-clef est la notion de couplage. Un couplage est, en termes simples, l'association de variables du système avec des contraintes à partir desquelles elles peuvent être calculées. Les variables ne pouvant être couplées ne peuvent pas être calculées ; les variables qui peuvent être couplées à plusieurs contraintes peuvent être calculées de plusieurs façons.

Définition 1.3.1 (Couplage). Un couplage est un ensemble d'arcs non adjacents, i.e. sans sommet commun. Autrement dit, un couplage sur le graphe biparti (C, V, E) est un sous-ensemble E' de E tel que : pour tout $e_{i,j} = (c_i, v_j)$ et $e_{k,l} = (c_k, v_l) \in E'$,

$$\begin{cases} c_i \neq c_k, \\ v_j \neq v_l. \end{cases}$$

Définition 1.3.2 (Couplage maximal). Un couplage maximal est un couplage de taille maximale.

Définition 1.3.3 (Couplage complet). Un couplage complet sur un ensemble de sommets V est un couplage qui couvre tous les sommets de V . On dit aussi que les sommets de V sont couplés.

On citera l'algorithme de (Hopcroft and Karp, 1973) pour la recherche d'un couplage complet.

Un couplage complet sur les variables inconnues d'un sous-système surveillable réordonne les équations de ce sous-système. En terme graphique, le choix d'un couplage complet oriente le graphe biparti associé au sous-système surveillable de manière à former un chemin d'estimation, comme suit :

- un arc appartenant au couplage est orienté de la contrainte vers la variable ;
- un arc n'appartenant pas au couplage est orienté de la variable vers la contrainte.

Il y a toujours une contrainte non couplée, car un sous-système surveillable contient une contrainte de plus que de variables : cette contrainte est dite *redondante*.

Exemple 1.3.1. Reprenons l'exemple 1.2.2, et considérons le sous-système surveillable $\{c_2, c_3, c_4, c_5, c_6\}$. On remarquera qu'il y a bien une équation de plus que de variables inconnues. Faire un couplage revient à choisir un « 1 » par colonne et par ligne. Un exemple de couplage est indiqué sur la table 1.6 ci-dessous par les « ① ». Les « ① » représentent les arcs du couplage.

TABLE 1.6 – Couplage sur un sous-système surveillable

	q	i	u_R	u_C	E	$u_{R,m}$	i_m
c_2	①			1			
c_3			1	①	1		
c_4	1	①					
c_5		1					1
c_6			①			1	

Le graphe biparti orienté associé à ce couplage, donné figure 1.7, indique l'ordre des opérations de calcul : calcul de u_R grâce à sa mesure, puis de u_C grâce à la relation de Kirchhoff c_3 , puis de la charge q grâce à la relation de la capacité c_2 , puis dérivation de la charge pour obtenir le courant i . Un couplage complet a effectivement permis de calculer successivement les variables inconnues, dans l'ordre indiqué par le couplage :

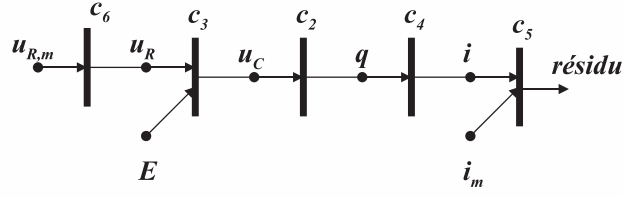


FIGURE 1.7 – Orientation des équations du sous-système surveillable

1. $u_R = u_{R,m}$,
2. $u_C = E - u_{R,m}$,
3. $q = C(E - u_{R,m})$,
4. $i = C(\frac{dE}{dt} - \frac{du_{R,m}}{dt})$.

L'équation non couplée (correspondant à l'ampèremètre) sert à calculer un résidu. Dans cet exemple, l'orientation fournit un chemin de calcul du résidu, que l'on peut implanter en simulation.

Remarquons que cette méthode est aussi utilisée dans (Blanke *et al.*, 2006), non pour obtenir un schéma de simulation, mais afin de construire, par éliminations successives, la relation de redondance analytique :

$$i_m - C(\frac{dE}{dt} - \frac{du_{R,m}}{dt}) = 0 \quad (1.2)$$

L'écriture de cette relation de redondance analytique est possible car le chemin donné par le couplage est sans boucle et car les fonctions sont inversibles formellement.

Dans l'exemple 1.3.1 qui précède, le graphe orienté par ce couplage est sans boucle. Or, dans le cas général, les orientations induites par les couplages peuvent contenir des boucles. Les boucles sont des systèmes de relations qui doivent être résolues ensemble ; elles sont appelées, en théorie des graphes, composantes fortement connexes, et la décomposition du graphe orienté en composantes fortement connexes permet par conséquent d'identifier ces boucles (Düstegör, 2005).

Définition 1.3.4 (Graphe orienté). Un graphe orienté G est un couple (X, A) , où :

- X est l'ensemble des sommets du graphe ;
- $A \subset X \times X$ est l'ensemble des arcs orientés du graphe.

Définition 1.3.5 (Chemin). Un chemin d'un graphe orienté G est une suite $(x_0, x_1, \dots, x_{n-1}, x_n)$ de sommets de G tel que deux sommets consécutifs quelconques x_i et x_{i+1} sont reliés par un arc de G , i.e. $(x_i, x_{i+1}) \in A$.

Définition 1.3.6 (Composante fortement connexe (*Strongly Connected Component* – *SCC*)). Soit G un graphe orienté. Soit la relation $\equiv_G$ sur les sommets de G :

$$x \equiv_G y \Leftrightarrow \exists \text{ un chemin de } x \text{ à } y \text{ et de } y \text{ à } x.$$

Les classes de cette relation d'équivalence sont les composantes fortement connexes de G .

Remarquons que dans la définition ci-dessus, nous considérons que le graphe a été orienté par un couplage. Par conséquent, le chemin de x à y est différent du chemin de y à x .

Nous appelons *boucle* le sous-graphe défini par une composante fortement connexe dont le cardinal est supérieur à 1. Les couplages qui induisent des graphes ayant la même décomposition en composantes fortement connexes sont dits équivalents. On distingue généralement les boucles algébro-différentielles, qui font intervenir une contrainte différentielle, et les boucles algébriques.

Les boucles induites par un couplage sont également identifiées par la décomposition en blocs de Hall du graphe orienté par le couplage (Düstegör *et al.*, 2004b).

Définition 1.3.7 (Décomposition en blocs de Hall d'un graphe, pour un couplage complet $\mathcal{C}$ donné). Soient :

1. $S = (C, X, E)$ un graphe possédant un couplage complet sur X ;
2. $\mathcal{C}$ un de ces couplages complets sur X ;
3. C_c l'ensemble des contraintes couplées ;
4. E_c l'ensemble des arcs liant une contrainte de C_c à X .

L'algorithme de Dulmage-Mendelsohn du graphe (C_c, X, E_c) décompose le graphe en blocs $B_k = (C_k, X_k, E_k)$, $k = 1 \dots p$, tels que :

- $(C_1, \dots, C_p)$ est une partition de C_c ;
- $(X_1, \dots, X_p)$ est une partition de X .

Les blocs B_k sont les blocs de Hall de S , pour le couplage $\mathcal{C}$.

Définition 1.3.8 (Taille d'un bloc de Hall). La taille n d'un bloc de Hall $B = (C, X, E)$ est $n = |C| = |X|$.

Exemple 1.3.2 (Suite de l'exemple 1.3.1.). Considérons maintenant le sous-système surveillable $M_2 = \{c_1, c_2, c_3, c_4, c_5\}$. Ci-après, figure 1.9, se trouve le graphe orienté de ce sous-système pour le couplage complet $\mathcal{C} : \mathcal{C} = \{(c_3, u_R), (c_1, i), (c_4, q), (c_2, u_C)\}$. La décomposition en blocs de Hall du graphe orienté induit par ce couplage fait apparaître un seul bloc de Hall $B = (C, X, E)$ avec :

- $C = \{c_1, c_2, c_3, c_4\}$;
- $X = \{u_R, i, q, u_C\}$;
- $E = \{(c_3, u_R), (c_1, i), (c_4, q), (c_2, u_C)\}$.

La boucle différentielle $c_3 \rightarrow u_R \rightarrow c_1 \rightarrow i \rightarrow c_4 \rightarrow q \rightarrow c_2 \rightarrow u_C \rightarrow c_3$ qui apparaît correspond à l'équation de la charge du circuit RC, qui est une équation différentielle-algébrique.

Regardons de plus près cette boucle algébro-différentielle : c_4 est la contrainte différentielle, orientée dans le sens de l'intégrale ; c_2, c_3, c_1 sont les contraintes algébriques. En réécrivant les contraintes *algébriques* dans l'ordre donné par le graphe de la boucle, on obtient :

1. $u_C = \frac{q}{C}$,
2. $u_R = E - \frac{q}{C}$,
3. $i = \frac{E}{R} - \frac{q}{R.C}$, i.e. : $\dot{q} = -\frac{q}{R.C} + \frac{E}{R}$.

Le système d'équations associé à ce couplage est donc la forme d'état 1.3, dont l'état est la variable q :

$$\begin{cases} \dot{q} &= -q/(R.C) + E/R \\ i_m &= -q/(R.C) + E/R \end{cases} \quad (1.3)$$

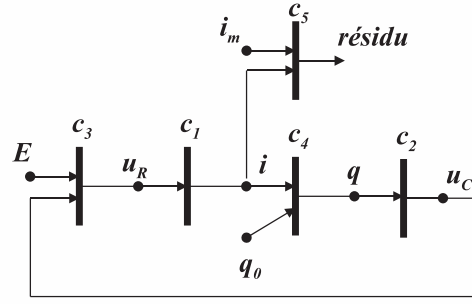


FIGURE 1.8 – Orientation conduisant à une boucle différentielle

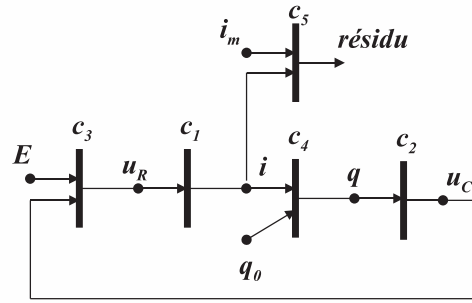


FIGURE 1.9 – Orientation conduisant à une boucle différentielle

Remarquons que, dans l'exemple 1.3.1, l'orientation fournissait à la fois un chemin de calcul du résidu, et un moyen de construire une RRA. Au contraire, dans cet exemple, l'orientation ne peut pas conduire à l'écriture d'une RRA. Néanmoins, un résidu peut toujours être calculé en simulant la forme d'état obtenue en réécrivant les équations du modèle dans le sens indiqué par le couplage. Le graphe de simulation du résidu est obtenu en faisant une projection du graphe structurel orienté de la figure 1.9 sur le schéma de simulation du système.

Le couplage utilisé dans cet exemple correspond à une causalité entrée-sortie, ou intégrale, à l'inverse du couplage de l'exemple 1.3.1, qui correspondait à une causalité différentielle, ou dérivée. Pour plus d'informations sur la notion de causalité, le lecteur pourra se référer à (Gentil *et al.*, 2004) : on retiendra qu'il y a plusieurs définitions de la causalité : l'une d'entre elle est fondée sur une étude physique du processus et traduit la façon dont les variables s'influencent les unes les autres (il s'agit de la causalité entrée-sortie, intégrale) ; une autre définition indique que la causalité peut être définie simplement comme un ordre de calcul. On retrouve ces deux définitions dans nos exemples : l'exemple 1.3.1 utilise une causalité qui est simplement un ordre de calcul ; la causalité de l'exemple 1.3.2 est à la fois un ordre de calcul et la traduction de l'influence des variables les unes sur les autres.

Les boucles algébro-différentielles sont la représentation graphique d'équations algébro-différentielles, les boucles algébriques représentent des systèmes d'équations algébriques avec autant d'inconnues que d'équations. Sans entrer pour l'instant plus avant dans les explications, calculer les variables inconnues de ces systèmes n'est pas immédiat. Nous évoquerons ces points à la

Pour résumer ce développement, un couplage va nous permettre d’orienter le graphe structuré de façon à former une séquence de calcul du résidu, exploitable par un logiciel de simulation. Cette séquence de calcul pourra contenir des boucles ou ne pas en contenir, comme l’ont montré les deux exemples de cette section. Elle devra également vérifier des conditions de calculabilité, discutées section 1.3.3. La présence de boucle est notamment source de problèmes de calculabilité en pratique.

1.3.3 Caractérisation des sous-systèmes surveillables calculables

Les couplages peuvent être utilisés afin d’assister la génération des résidus, et guider la procédure de calcul des variables inconnues, en indiquant un chemin de calcul des variables internes inconnues, à partir des variables d’entrées et sorties accessibles sur le système. Nous nous plaçons dans ce cadre où l’on suit les chemins donnés par un couplage, et où l’on cherche à calculer les variables inconnues par une méthode de simulation. Toujours dans ce cadre, nous cherchons à caractériser les sous-systèmes surveillables calculables ; c’est l’objet de cette section.

Tous les couplages ne conduisent pas à réécrire les équations du modèle sous une forme exploitable. Les trois sous-sections suivantes rendent compte des principales références sur la calculabilité des chemins donnés par un couplage, dans une hypothèse de simulation des résidus.

a) Calculabilité d’un bloc de Hall de taille égale à 1

Une bloc de Hall de taille égale à 1 représente :

- d’un point de vue graphique, une contrainte liée à une variable ;
- d’un point de vue analytique, une contrainte $c(x_1, \dots, x_j, \dots, x_m, z_1, \dots, z_p) = 0$ dont toutes les variables $(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_m, z_1, \dots, z_p)$ sauf une (x_j) ont été préalablement déterminées ou éliminées.

La calculabilité d’un bloc de Hall $B = (c, x, e)$ de taille 1 dépend de l’inversibilité de la contrainte c par rapport à la variable x .

Définition 1.3.9 (Inversibilité). Soit une contrainte $c_i(\dots, x_j, \dots) = 0$, si x_j peut être calculée à partir des autres variables de c_i supposées connues – alors c_i est inversible par rapport à x_j . Sinon, c_i n’est pas inversible par rapport à x_j , information qui est indiquée dans le graphe structuré par $S(i, j) = -1$: l’arc (c_i, x_j) est dit non-inversible.

Par conséquent, cette notion permettra d’écarter les générateurs de résidus qui imposeront d’inverser – la liste n’est pas exhaustive – les tables, les cartographies, les fonctions non-linéaires, les hystérésis, certaines fonctions logiques, les conditions d’activation (comparaisons), et les capteurs logiques, comme les détecteurs : dans (Conrard *et al.*, 2009), les auteurs intègrent cette contrainte dans l’analyse.

Relation entre non-inversibilité et causalité. La notion de causalité, que nous avons introduite à travers l'exemple 1.3.2, est en relation avec celle de non-inversibilité. Les trois définitions ci-dessous, qui portent sur des causalités dont nous nous servirons dans la suite de ce rapport, établissent ce lien.

Définition 1.3.10 (Causalité intégrale). Un graphe biparti G est en causalité intégrale si pour toute contrainte différentielle c de G , $c : x_d = \frac{dx_i}{dt}$, c est choisie :

- non-inversible par rapport à la variable dérivée x_d ;
- inversible par rapport à la variable intégrale x_i .

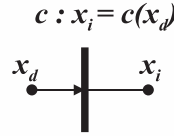


FIGURE 1.10 – Causalité intégrale

Définition 1.3.11 (Causalité dérivée). Un graphe biparti G est en causalité dérivée si pour toute contrainte différentielle c de G , $c : x_d = \frac{dx_i}{dt}$, c est choisie :

- non-inversible par rapport à la variable intégrale x_i ;
- inversible par rapport à la variable dérivée x_d .

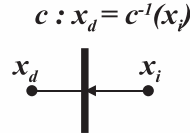


FIGURE 1.11 – Causalité dérivée

Définition 1.3.12 (Causalité mixte). Un graphe biparti G est en causalité mixte si certaines contraintes différentielles c_k de G sont choisies non-inversibles par rapport à leur variable intégrale x_i^k , et si d'autres contraintes différentielles $c_{k'}$ sont choisies non-inversibles par rapport à leur variable dérivée $x_d^{k'}$.

b) Calculabilité d'un bloc de Hall de taille $n > 1$, représentant une équation différentielle algébrique

Considérons un bloc de Hall de taille $n > 1$ représentant une équation différentielle algébrique :

$$F(\dot{x}_1, x_1, x_2, z) = 0 \quad (1.4)$$

Dans l'équation 1.4, la dimension de F est égale à n , x_1 représente l'ensemble des variables dérivées, x_2 représente l'ensemble des variables algébriques, z représente les variables connues.

La forme ci-dessus (1.4) est la forme implicite de l'équation différentielle-algébrique (DAE en abrégé). La forme semi-explicite est donnée ci-dessous (1.5) :

$$\begin{cases} \dot{x}_1 &= f(x_1, x_2, z) \\ 0 &= h(x_1, x_2, z) \end{cases} \quad (1.5)$$

Dans ce travail, nous nous sommes limités aux formes semi-explicites, qui peuvent être résolues en simulation.

Résolution numérique des DAE. Pour les formes semi-explicites, il existe, d'après (Shampine *et al.*, 1999), deux familles de méthodes de résolution numérique : les méthodes dites « directes » et les méthodes dites « ODE ». La méthode « ODE » est utilisée en simulation et consiste à passer par le biais d'une équation différentielle ordinaire en transformant l'équation 1.4 en :

$$\begin{cases} \dot{x}_1 &= f(x_1, x_2, z) \\ x_2 &= h_1(x_1, z) \end{cases} \quad (1.6)$$

Dans l'équation 1.6, la fonction h_1 est obtenue en inversant h par rapport à x_2 . Cette équation se ramène ainsi à l'équation différentielle ordinaire suivante :

$$\dot{x}_1 = f(x_1, h_1(x_1, z), z) \quad (1.7)$$

Dans la communauté IA, des travaux traitant de la calculabilité des boucles algébro-différentielles, en se plaçant dans un contexte de simulation, ont fait l'objet de publications :

Dans (Pulido *et al.*, 2008), il est remarqué que, dans un contexte de simulation, la causalité intégrale est obligatoire dans les boucles.

Dans (Svärd and Nyberg, 2008), les auteurs décrivent :

- les formes explicites et semi-explicites des équations différentielles algébriques ;
- la notion d'index différentiel, qui permet d'estimer la difficulté de résolution ;
- les conditions de calculabilité structurelle par la méthode « ODE » pour les équations différentielles algébriques, à savoir :
 - la possibilité d'écrire une forme semi-explicite ;
 - la possibilité de résoudre la partie algébrique de la forme semi-explicite, et se ramener à une équation différentielle ordinaire ;
 - la connaissance et la cohérence des conditions initiales.

c) Calculabilité d'un bloc de Hall de taille $n > 1$, représentant une boucle algébrique

Considérons un bloc de Hall de taille $n > 1$ représentant un système avec autant de relations algébriques que d'inconnues, qui doivent être résolues ensemble. Dans une approche de simulation, des solveurs de boucle algébrique peuvent être utilisés (une méthode de Newton, par exemple). Ainsi, même dans le cas où une solution analytique ne peut pas être calculée, il se peut

qu'une solution numérique puisse l'être.

Cependant, résoudre numériquement un système algébrique pose une question. Celle-ci concerne le nombre de solutions du système d'équations : il se peut qu'un tel système possède plusieurs solutions, auquel cas le solveur de boucle algébrique convergera vers l'une d'entre elles. Or, cette solution obtenue peut ne pas correspondre à la valeur réelle, interprétable physiquement.

L'étude et la résolution de ces boucles doit par conséquent être menée au cas par cas. Il n'existe pas de critère général, ou structurel, permettant de savoir si une boucle est solvable, avec une solution unique. On citera néanmoins le travail de (Rosich *et al.*, 2009), qui considère que les seules boucles algébriques solvables sont les systèmes linéaires, et qui introduit pour cette raison un symbole 'l' dans la matrice d'incidence, indiquant les dépendances linéaires.

Dans le même sens, on citera également une décomposition de graphe intéressante, qui est la « Combinatorial Canonical Form Decomposition » (CCF Decomposition) exposée dans (Murata, 2000). Cette décomposition, à mi-chemin entre un pivot de Gauss et la décomposition de Dulmage-Mendelsohn, repose sur la connaissance des liens linéaires du modèle, ainsi que de la valeur des coefficients linéaires.

1.4 Contenu et apports du mémoire

Pour notre modèle d'application, dont les dimensions ont été précisées au paragraphe 1.3.1, rechercher l'exhaustivité des systèmes surveillables n'est pas envisageable : les algorithmes que nous avons cités à la section 1.2.4 ne sont pas adaptés à notre modèle. Le temps de calcul de la recherche exhaustive dépasse plusieurs jours. L'objet principal de ce mémoire est donc la maîtrise de la complexité de la recherche des sous-systèmes surveillables.

Par conséquent, plutôt que de chercher toutes les solutions, sans tenir compte de contraintes de calculabilité ou d'exploitation, puis de rajouter ces contraintes pour sélectionner uniquement les solutions réellement implantables, nous voulons intégrer dès que possible ces contraintes dans l'Analyse Structurelle pour aboutir directement à l'ensemble des solutions qui nous intéressent. Il y a deux sortes de contraintes que nous cherchons à intégrer dans l'Analyse Structurelle : les contraintes de calculabilité des résidus et les contraintes dues à un cahier des charges.

Intégrer des critères de calculabilité des résidus dans l'Analyse Structurelle doit permettre de restreindre le nombre de sous-systèmes trouvés en premier lieu. La calculabilité d'un résidu dépend des propriétés d'inversibilité et de non-inversibilité des fonctions et composantes fortement connexes qui constituent la séquence de calcul (ou le schéma de simulation).

Par ailleurs, la prise en compte d'un cahier des charges de diagnostic dans la recherche des sous-systèmes surveillables, avec l'intention de ne pas trouver ceux qui ne sont pas discriminants, doit également contribuer à la maîtrise de la complexité.

Dans le chapitre suivant, nous proposons différents algorithmes permettant de mieux maîtriser la complexité :

1. un algorithme de définition des inversibilités a priori (section 2.2), pour les contraintes différentielles. Il s'agit d'identifier quelles contraintes différentielles apparaissent toujours

dans une boucle et d'interdire la causalité différentielle pour ces contraintes (la causalité intégrale étant obligatoire dans les boucles) ;

2. un algorithme de décomposition canonique d'un graphe biparti, prenant en compte les non-inversibilités (section 2.3) ;
3. un critère d'isolabilité a priori, prenant en compte les non-inversibilités (section 2.4) ;
4. des algorithmes de recherche des sous-systèmes surveillables intégrant les non-inversibilités (section 2.5) ;
5. un algorithme de recherche des sous-systèmes surveillables, tenant compte d'un cahier des charges, et retournant les sous-systèmes permettant de discriminer structurellement les défauts du cahier des charges (section 2.6).

Chapitre 2

Algorithmes optimisés pour la recherche exhaustive des sous-systèmes surveillables

Résumé

Ce deuxième chapitre est consacré aux développements d'algorithmes que nous avons menés. Ces développements visent à simplifier le processus d'analyse en limitant la recherche aux sous-systèmes surveillables possédant de bonnes propriétés de calculabilité, d'une part, et offrant un bon potentiel de discrimination des défauts d'autre part.

La première partie de ce chapitre a pour objectif d'introduire les algorithmes développés. Ceux-ci se divisent en deux catégories : les algorithmes permettant de trouver des sous-systèmes aidant à discriminer les fautes ; et les algorithmes visant à intégrer des contraintes de calculabilité à l'analyse structurelle. Pour cette seconde catégorie, plusieurs algorithmes reposant sur différentes hypothèses de calculabilité (causalité intégrale, causalité mixte, causalité dérivée) sont proposés. Le lecteur trouvera ensuite, pour ces différentes hypothèses, des algorithmes d'analyse de la détectabilité des défauts, d'analyse de l'isolabilité, et de recherche des sous-systèmes calculables.

Abstract

This second chapter is devoted to the algorithms which have been developed. These algorithms aim at simplifying the structural analysis process, by targetting the monitorable subsystems possessing, on one hand, good calculability properties, and on the other hand, good fault discrimination properties.

The purpose of the first part of the chapter is to introduce the developed algorithms. They are divided into two categories : algorithms allowing to find subsystems which help discriminate the faults ; and algorithms aiming at integrating calculability constraints to a structural analysis. For this latter category, several algorithms based on different hypotheses (integral causality, differential causality, mixed causality) are proposed.

The reader will then find, for these different hypotheses, detectability analysis algorithms, isolability analysis algorithms, and computable monitorable subsystems' search algorithms.

2.1 Introduction : objectifs et aperçu des développements algorithmiques

Une analyse structurelle d'un système se déroule usuellement en plusieurs étapes, qui ont été présentées au chapitre précédent :

1. écrire un modèle structurel du système à surveiller. Ce modèle structurel peut dériver d'un modèle de comportement – dont les paramètres n'ont pas besoin d'être connus ou identifiés – ou d'une étude qualitative du procédé. En effet, les relations n'ont pas besoin d'être explicitées : il suffit de savoir qu'il existe un lien entre deux grandeurs ; la nature du lien n'a pas d'utilité, de pertinence.
2. réaliser la décomposition de Dulmage-Mendelsohn de la matrice d'incidence, afin de déterminer les composantes (sous-systèmes) non surveillables.
3. rechercher tous les sous-systèmes surveillables, c'est à dire ceux sur lesquels un résidu sensible aux défauts peut être potentiellement généré.
4. faire le tri entre les sous-systèmes, pour ne garder que ceux qui permettent de générer effectivement un résidu. Faire le tri entre les sous-systèmes suppose de connaître la nature et les propriétés des relations du modèle.

Le critère de tri (étape 4 ci-dessus) que nous souhaitons utiliser et que nous souhaitons intégrer est défini dans la section suivante 2.1.1. Ce critère est « semi-structurel » : il nous faut connaître, pour l'utiliser, deux informations sur les relations liant les grandeurs du modèle : nous avons besoin, d'une part, de savoir si une relation est inversible ou non-inversible, et d'autre part, si elle est dynamique ou algébrique.

Une introduction aux algorithmes d'analyse structurelle et aux modifications effectuées dans le cadre de ce travail, afin de prendre en compte un cahier des charges de diagnostic et des critères de calculabilité, sera ensuite présentée : les algorithmes actuels sur lesquels nous nous appuyons seront d'abord rappelés (2.1.2), puis nous expliquerons le but des algorithmes modifiés (2.1.3).

2.1.1 Critère de validité des sous-systèmes surveillables

Nous proposons un critère de calculabilité des résidus qui s'appuie sur les deux points suivants, mentionnés à la section 1.3.3 :

1. la causalité intégrale est obligatoire dans les boucles algébro-différentielles ;
2. résoudre numériquement une boucle algébro-différentielle est possible si cette boucle peut être réécrite sous la forme d'une ODE.

Les sous-systèmes vérifiant notre critère de validité seront appelés *sous-systèmes validés par les inversibilités* (définition 2.1.3).

Définition 2.1.1 (Graphe algébrique d'une boucle algébro-différentielle). Le graphe algébrique d'une boucle algébro-différentielle est construit en supprimant les contraintes différentielles, de la façon suivante :

- identification des contraintes différentielles $c_d : x_d = \dot{x}_i$;

- suppression de l'arc liant x_d à c_d .

L'analyse du graphe algébrique d'une boucle algébro-différentielle permet de savoir s'il est possible de calculer $x_d = \dot{x}_i$ en fonction de x_i , et ainsi d'aboutir à une ODE. En effet, le graphe algébrique d'une boucle algébro-différentielle permet de visualiser le chemin allant de x_i à x_d . Ce chemin peut contenir des boucles, des fonctions non-inversibles, etc, comme nous l'avons vu à travers les exemples du premier chapitre. Afin de prendre en considération ces aspects, nous suggérons et introduisons les définitions suivantes.

Définition 2.1.2 (Couplage validé par les inversibilités). Un couplage $\mathcal{C}$ est validé par les inversibilités si :

- tous ses arcs sont inversibles ;
- la graphe orienté qu'il détermine ne comporte pas de boucle algébrique ;
- le graphe algébrique des boucles algébro-différentielles induites par $\mathcal{C}$ est sans boucle.

Définition 2.1.3 (Sous-système validé par les inversibilités). Soit M l'ensemble des relations constituant le sous-système surveillable. Soit $var(M)$ l'ensemble des variables inconnues apparaissant dans les relations M . Le sous-système surveillable est validé par les inversibilités s'il existe un couplage complet sur $var(M)$, validé par les inversibilités.

Remarquons que pour un sous-système surveillable M , il y a $card(M)$ couplages complets différents. Pour qu'un sous-système surveillable soit validé par les inversibilités, il suffit donc qu'un de ces couplages vérifie la définition 2.1.2.

La notion de couplage validé par les inversibilités traduit l'idée que, pour qu'un graphe orienté par un couplage fournisse une séquence de calcul implémentable, il doit être sans boucle une fois les contraintes dynamiques retirées. De plus, les fonctions algébriques parcourues une à une doivent être inversibles.

Ce critère est plus restrictif que ceux présentés à la section 1.3.3, car les auteurs de la littérature considèrent que les boucles algébriques peuvent être résolues. Etudions ce que ce critère englobe ou n'englobe pas : ceci fait l'objet des paragraphes suivants, qui proposent une discussion sur la notion de sous-système validé par les inversibilités.

Le lemme qui suit établit que la définition 2.1.3 conduit à écrire une équation différentielle algébrique sous la forme d'une équation différentielle ordinaire :

Lemme 1 (Ecriture d'une équation différentielle ordinaire). *Une équation différentielle algébrique induite par un couplage validé par les inversibilités peut s'écrire sous la forme d'une équation différentielle ordinaire.*

Preuve 1. Dans la définition 2.1.1, supprimer le lien entre x_d et c_d pour chaque contrainte différentielle c_d revient à dire que x_i est calculable sans connaître x_d . x_i est donc considéré comme connu. De plus, le fait qu'il existe un chemin sans boucle inversible signifie que toutes les variables de la boucle peuvent se déduire, précisément en suivant ce chemin, à partir des entrées z de la boucle et des variables intégrales. Par conséquent, une équation différentielle algébrique induite par un couplage validé par les inversibilités correspond à un système d'équations :

$$\begin{cases} \dot{x}_1 &= f(x_1, x_2, z) \\ \dot{x}_2 &= h_1(x_1, z) \end{cases} \quad (2.1)$$

Cette équation se réécrit en l'équation différentielle ordinaire : $\dot{x}_1 = f(x_1, h_1(x_1, z), z)$ $\square$

A l'inverse, pour un sous-système M invalidé, deux cas peuvent se présenter :

1. les couplages de M induisent une boucle lorsque les contraintes dynamiques sont retirées. Cette boucle est donc algébrique ; dans ce cas, il faut connaître les équations pour savoir si la boucle est solvable, et la notion d'inversibilité n'est de fait plus utile. Notons que cette boucle peut se situer dans la partie algébrique d'une boucle différentielle algébrique (calcul de $\dot{x}$ à partir de x), comme en dehors d'une boucle différentielle.
2. les couplages de M nécessitent d'inverser une relation non-inversible. Cette impossible inversion peut se situer de même dans la partie algébrique d'une boucle algébro-différentielle (auquel cas il est impossible de calculer $\dot{x}$ à partir de x), ou en dehors d'une boucle différentielle.

La figure 2.1 illustre un exemple du premier cas. Sur cette figure, le sous-système surveillable orienté contient une boucle différentielle (constituée des contraintes c_3, c_4, c_5, c_{m2} , et de la contrainte différentielle c_d). Le couplage indiqué par les flèches n'est pas validé car il induit une boucle algébrique (constituée des contraintes c_3, c_4, c_5) dans la partie algébrique de la boucle algébro-différentielle.

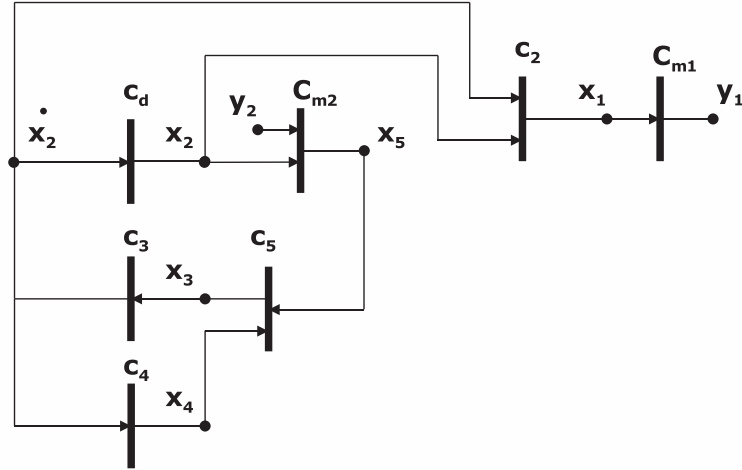


FIGURE 2.1 – Couplage induisant une boucle dans la partie algébrique d'une équation différentielle algébrique

L'exemple de la figure 2.1 montre bien qu'un sous-système non validé n'est pas nécessairement inutilisable mais générer un résidu sera plus complexe. Ici, la boucle algébrique est peut-être solvable. On remarquera donc que la possibilité de générer un résidu ne se déduit pas du seul examen des non-inversibilités.

2.1.2 Présentation des algorithmes actuels

Deux algorithmes pour la recherche des sous-systèmes surveillables sont présentés ici. Ces algorithmes renvoient les mêmes résultats, mais n'ont pas la même complexité. Pour l'algorithme de la décomposition de Dulmage-Mendelsohn, on se référera à (Murota, 2000).

a) Formation des couplages complets pour la recherche des sous-systèmes surveillables.

Le principe de l'algorithme de (Düstegör, 2005) est de rechercher des couplages complets sur l'ensemble des variables inconnues X^+ du système sur-contraint S^+ .

Un couplage complet donne une séquence de calcul des variables inconnues de X^+ . Chaque variable est couplée à exactement une contrainte, par conséquent il y a des contraintes non couplées. Ces contraintes non couplées permettent de construire les sous-systèmes surveillables. Différents couplages sont considérés pour construire l'ensemble des sous-systèmes surveillables. La figure 2.2 donne le principe de cet algorithme, qui est rappelé dans l'encart « Algorithme 1 ».

Sur la figure 2.2, les « 1 » entourés représentent un couplage complet, les blocs sur la diagonale représentent les blocs de Hall, et R est l'ensemble des équations non couplées. Un sous-système surveillable est obtenu en associant une séquence de calcul des variables inconnues avec une équation redondante.

2.1. Introduction : objectifs et aperçu des développements algorithmiques

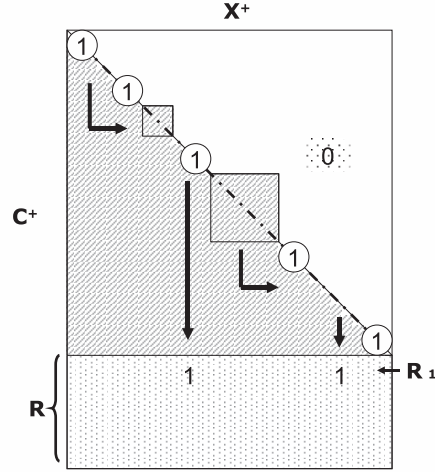


FIGURE 2.2 – Couplage complet pour la recherche des sous-systèmes surveillables

Algorithm 1 $MS = Recherche_exhaustive(S^+)$

Require: Calcul préalable de $S^+ = (C^+, X^+, E^+)$, de taille $n * m$

- 1: $MS = \phi$
 - 2: **for** chaque bloc $B = (C_B \subset C^+, X^+, E_B)$ carré de taille m de S **do**
 - 3: **if** B est juste-contraint **then**
 - 4: Déterminer les blocs de Hall de B
 - 5: Identifier la séquence de calcul de chaque variable inconnue
 - 6: Dériver $(n - m)$ sous-systèmes surveillables, notés MS_B , en substituant aux variables des $(n - m)$ contraintes non couplées leur séquence de calcul
 - 7: $MS = MS \cup MS_B$
 - 8: **end if**
 - 9: **end for**
 - 10: Retourner MS
-

Exemple 2.1.1. Appliquons l'algorithme à la matrice d'incidence du circuit RC de l'exemple 1.2.1, rappelée ici :

TABLE 2.1 – Matrice d'incidence du modèle 1.1

	q	i	u_R	u_C
c_1		1	1	
c_2	1			1
c_3			1	1
c_4	1	1		
c_5		1		
c_6			1	

Il y a C_6^4 blocs carrés (ou couplages complets) à considérer. Par exemple, deux sous-systèmes surveillables sont ainsi déterminés à partir du couplage complet de la table 2.2 : $M_1 = \{c_1, c_5, c_6\}$,

et $M_2 = \{c_1, c_2, c_3, c_4, c_5\}$. Considérer d'autres blocs carrés permet de trouver les deux autres sous-systèmes surveillables, M_3 et M_4 .

TABLE 2.2 – Couplage complet permettant de déterminer deux sous-systèmes surveillables

	i	u_R	u_C	q
c_5	①			
c_1	1	①		
c_3		1	①	
c_2			1	①
c_4	1			1
c_6		1		

b) Algorithme MSO pour la recherche des sous-systèmes surveillables.

L'algorithme présenté dans (Krysander *et al.*, 2008) peut être vu comme un algorithme dichotomique : on choisit une équation e_1 , puis l'espace des MSO (sous-systèmes surveillables) est scindé en deux : l'ensemble des MSO contenant e_1 et l'ensemble des MSO ne contenant pas e_1 .

Cet algorithme est récursif, c'est-à-dire que le principe de dichotomie est reconduit dans chacun des deux sous-ensembles formés. Pour une seconde équation e_2 :

- parmi les MSO ne contenant pas e_1 , il y a ceux qui contiennent e_2 et ceux qui ne contiennent pas e_2 ;
- parmi les MSO contenant e_1 , il y a ceux qui contiennent e_2 et ceux qui ne contiennent pas e_2 .

On est ainsi amené à explorer les ensembles $(S^+ \setminus \{e_1\})^+$, $(S^+ \setminus \{e_2\})^+$... $(S^+ \setminus \{e_1\})^+$ est un sous-système surdéterminé qui contient l'ensemble des MSO ne contenant pas e_1 ; $(S^+ \setminus \{e_2\})^+$ est un sous-système surdéterminé qui contient l'ensemble des MSO ne contenant pas e_2 .

Cet algorithme, comme tout algorithme dichotomique, peut être visualisé comme un arbre dont la racine est le système S^+ et dont les feuilles sont les MSO . Les différents sous-systèmes traversés, qui sont les nœuds internes de l'arbre, sont appelés les ensembles PSO . On passe d'un nœud – ou d'un PSO – P à ses fils en calculant les ensembles $(P \setminus \{e\})^+$, $e \in P$. Cela correspond à rechercher les MSO de P qui ne font pas intervenir e .

Définition 2.1.4 (PSO (Proper Structurally Overdetermined)). Un ensemble P est un PSO si $P^+ = P$.

La figure 2.3 schématise le fonctionnement de cet algorithme : en partant de S^+ , on calcule les PSO $(S^+ \setminus \{e_1\})^+$, $(S^+ \setminus \{e_2\})^+$... Ces PSO sont les nœuds internes de l'arbre. Les PSO fils de S^+ ont un degré de redondance égal au degré de redondance de S^+ , moins 1. D'une manière générale, la différence entre le degré de redondance entre un PSO et l'un de ses PSO fils est égal à 1. Ainsi, les PSO situés à la même profondeur (nombre d'arêtes de la racine au nœud) ont le même degré de redondance. Les ensembles MSO , qui ont un degré de redondance égal à 1, constituent par conséquent les feuilles de l'arbre. Au total, la hauteur de

2.1. Introduction : objectifs et aperçu des développements algorithmiques

l'arbre est égale à d .

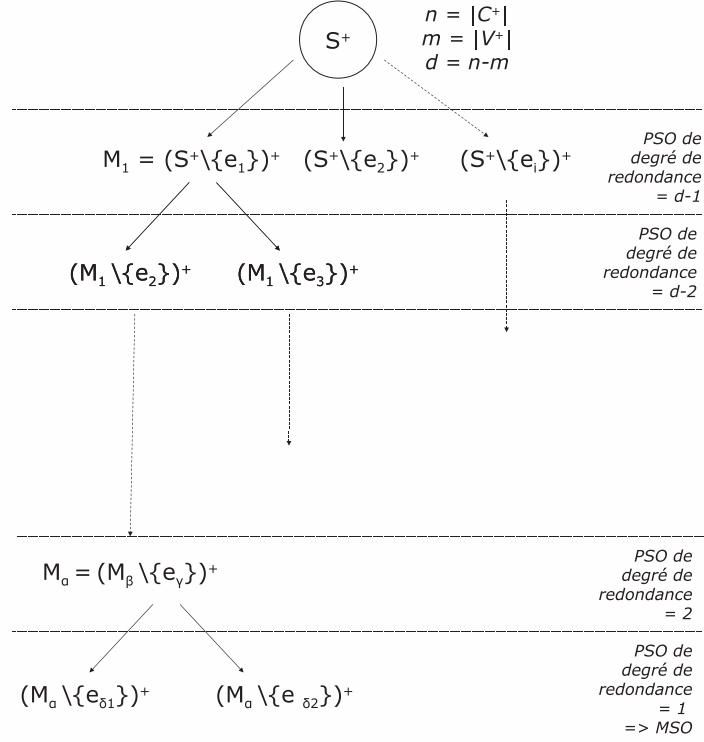


FIGURE 2.3 – Arbre des PSO

Dans (Krysander *et al.*, 2008) sont décrits les mécanismes permettant de ne trouver un *MSO* qu'une seule fois :

1. le retrait de certaines contraintes d'un *PSO* P aboutit au même *PSO* fils : ce sont les contraintes de $P \setminus (P \setminus \{e\})^+$. De façon simplifiée, si e apparaît dans une chaîne de contraintes, alors le retrait de e entraîne le retrait de toute la chaîne de contrainte. Formellement :
 $\forall e_1, \forall e_2, e_2 \in P \setminus (P \setminus \{e_1\})^+ \Rightarrow (P \setminus \{e_1\})^+ = (P \setminus \{e_2\})^+.$
2. si le sous-arbre issu du *PSO* P en retirant la contrainte e , qui a pour racine le *PSO* $(P \setminus \{e\})^+$ a déjà été exploré, alors tous les *MSO* ne contenant pas e et contenus dans P ont été trouvés. L'arbre est alors tronqué.

Exemple 2.1.2. De même, illustrons l'algorithme sur le modèle du circuit RC de l'exemple 1.2.1 (table 2.1). Le retrait successif des équations de cette matrice permet de construire l'arbre des *PSO* donné à la figure 2.4. Cet arbre a une hauteur égale à 1, car le degré de redondance du circuit RC étudié est égal à 2. Les sous-systèmes surveillables, ou *MSO*, sont les feuilles de l'arbre. On notera que le retrait de c_2 , c_3 , ou c_4 aboutit au même *MSO*.

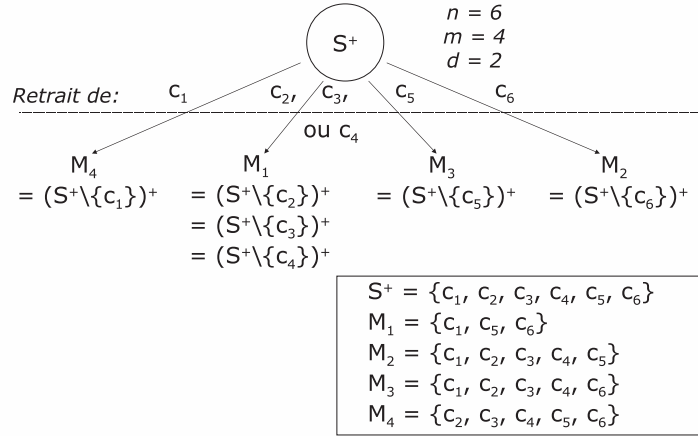


FIGURE 2.4 – Arbre des PSO du modèle RC

2.1.3 Prise en compte des non-inversibilités et d'un cahier des charges dans les algorithmes actuels

a) Prise en compte des non-inversibilités

Nous voulons intégrer un critère de calculabilité (définition 2.1.3). Pour intégrer le critère de calculabilité aux différentes étapes de l'analyse structurale, il nous faut :

1. définir les non-inversibilités ;
2. avoir une décomposition canonique intégrant les non-inversibilités, qui permettrait d'identifier la partie sur laquelle on peut trouver des sous-systèmes validés par les inversibilités ;
3. avoir des algorithmes de recherche des sous-systèmes surveillables validés par les inversibilités.

En réalité, il n'est pas toujours évident d'intégrer nos critères dans la recherche des sous-systèmes surveillables, sans dégrader les performances des algorithmes par rapport aux algorithmes actuels. En effet, il faut être capable de détecter des boucles algébriques, des boucles différentielles qui contiennent des boucles algébriques... le tout a priori, ou à tout le moins, au cours de la recherche. Nous verrons cependant (voir l'algorithme proposé au paragraphe 2.5.3) que cela est possible si l'on fait l'hypothèse de causalité intégrale. En revanche, si l'on fait l'hypothèse de causalité mixte, nous sommes obligés de substituer à la notion de sous-système validé par les inversibilités, celle de sous-système orientable, moins forte, pour la raison que les boucles algébriques sont dans ce cas difficilement détectables.

Définition 2.1.5 (Sous-système orientable). Soit M l'ensemble des relations constituant le sous-système surveillable. Soit $var(M)$ l'ensemble des variables apparaissant dans les relations M . Etant donné un ensemble de non-inversibilités, le sous-système surveillable est orientable s'il existe un couplage complet sur $var(M)$ ne comportant que des arcs inversibles.

Les sous-systèmes validés par les inversibilités sont inclus dans les sous-systèmes orientables. En effet, un sous-système validé par les inversibilités possède un couplage validé par les inversibilités. Ce couplage possède trois propriétés :

2.1. Introduction : objectifs et aperçu des développements algorithmiques

1. tous ses arcs sont inversibles ;
2. la graphe orienté qu'il détermine ne comporte pas de boucle algébrique ;
3. le graphe algébrique des boucles algébro-différentielles qu'il induit est sans boucle.

La première propriété définissant un sous-système orientable, les sous-systèmes validés par les inversibilités constituent bien une sous-classe des sous-systèmes orientables. L'inverse n'est pas vrai : il existe des sous-systèmes orientables qui ne sont pas validés par les inversibilités : il suffit pour s'en convaincre d'imaginer que les couplages peuvent générer des boucles algébriques.

Par ailleurs, même si la notion de sous-système orientable est dégradée par rapport à celle de sous-système validé, un gain de temps de calcul reste possible grâce à celle-ci. Cela est naturellement dû au fait que les sous-systèmes orientables sont un sous-ensemble des sous-systèmes surveillables.

Du fait de la difficulté de trouver directement les sous-systèmes validés par les inversibilités quand l'hypothèse de causalité mixte est faite, nous distinguerons dans la suite de ce rapport les méthodes fondées sur les deux hypothèses suivantes :

- causalité intégrale ;
- causalité mixte.

En faisant l'hypothèse d'une causalité mixte, on prêtera tout particulièrement attention à la causalité des contraintes différentielles : celles-ci peuvent être prises dans les deux sens, puisque l'on fait l'hypothèse de causalité mixte, sauf dans une boucle. De plus, un algorithme de tri est nécessaire : parmi les sous-systèmes orientables, nous ne garderons que ceux qui sont validés par les inversibilités. Cette phase de tri n'est pas nécessaire si l'on fait l'hypothèse d'une causalité intégrale, car l'algorithme proposé trouve directement les sous-systèmes validés par les inversibilités.

b) Prise en compte d'un cahier des charges

Le second facteur de réduction de la complexité de la recherche des sous-systèmes surveillables consiste à intégrer un cahier des charges. Au lieu de chercher à exprimer tout ce qui est diagnostiquable, au lieu de chercher l'ensemble des sous-systèmes surveillables, on se donne ici les fautes à diagnostiquer ($f_1, \dots, f_p$) et l'on cherche à savoir si ces fautes sont précisément diagnostiquables et quels sont les tests intéressants pour les discriminer. Ainsi, pour un cahier des charges fixé, on aura pour objectif de ne pas trouver les sous-systèmes non discriminants.

Cela sera réalisé (comme nous le verrons dans la section 2.6) en déterminant les systèmes surcontraints où des résidus insensibles à f_i peuvent être générés, puis en menant une recherche des sous-systèmes surveillables sur ces systèmes.

La figure 2.5 récapitule notre démarche et les algorithmes qui seront développés dans la suite de ce mémoire. Les rectangles grisés représentent les algorithmes classiques existant, qui ont été présentés précédemment dans cette section.

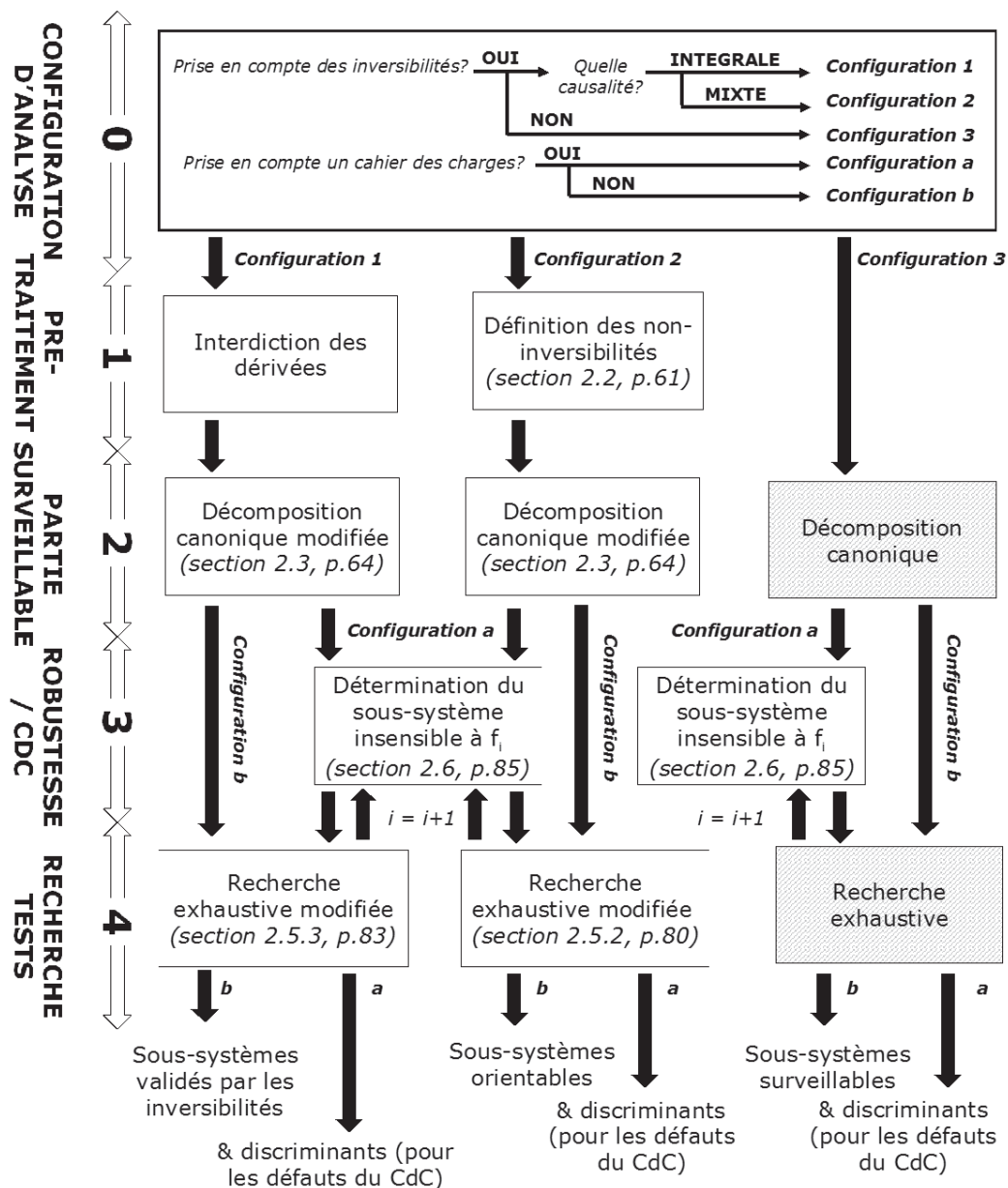


FIGURE 2.5 – Récapitulatif de la démarche

c) Comment choisir la bonne configuration d'analyse ?

La figure 2.5 présente les algorithmes développés, et montre comment les intégrer à la méthode d'analyse structurale une fois la configuration d'analyse choisie.

Le choix de la configuration d'analyse est déterminé par trois questions (en haut de la figure 2.5) :

1. Voulez-vous prendre en compte un cahier des charges de diagnostic ?

2. Voulez-vous intégrer les non-inversibilités à l'analyse ?
3. Voulez-vous une causalité intégrale ou mixte ?

En supposant que l'on ait un cahier des charges de diagnostic, des études d'isolabilité a priori (section 2.4), informant sur la configuration à choisir pour discriminer les défauts du cahier des charges. En effet, suivant les hypothèses faites, deux défauts pourront être isolables ou ne pas l'être. Il se peut, par exemple, que des défauts ne soient pas discriminables si l'on interdit la causalité dérivée, mais qu'ils le soient si celle-ci est autorisée – ce que nous verrons en pratique, sur un exemple, au chapitre 3 de ce rapport. On choisira alors une des configurations permettant de discriminer les défauts.

2.2 Détermination des inversibilités sur les contraintes différentielles

Dans cette partie, aucune hypothèse n'est faite au préalable sur l'orientation des contraintes différentielles ; les deux causalités sont possibles (causalité mixte). La seule restriction que nous faisons concerne l'obligation de prendre une causalité intégrale dans les boucles. Les résultats présentés dans cette section ont fait l'objet d'un article présenté au congrès de diagnostic DX 2010 (de Flaugergues *et al.*, 2010).

a) Exemple introductif

L'idée de l'algorithme présenté, et qu'illustre l'exemple de la figure 2.6, est la suivante : puisque la causalité intégrale est obligatoire dans les boucles, si l'on peut identifier au préalable qu'une contrainte dynamique apparaîtra toujours dans une boucle, alors on peut placer dès le départ une non-inversibilité interdisant la dérivée. Cette contrainte sur l'orientation du graphe nous intéresse, pour la compréhension du système d'une part, et d'autre part, car elle permettra, dans la suite de la démarche, de réduire le temps de calcul des algorithmes de recherche exhaustive.

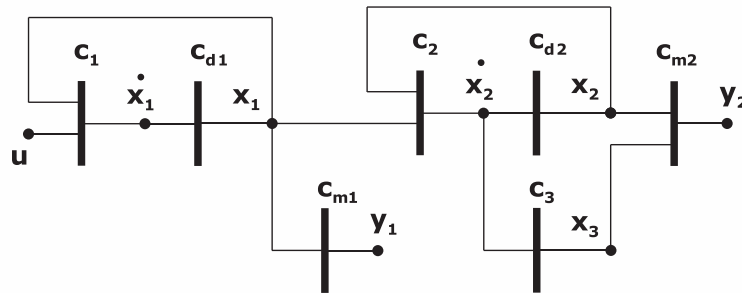


FIGURE 2.6 – Définition de la causalité intégrale pour la contrainte différentielle c_{d2}

Sur l'exemple de la figure 2.6, nous avons utilisé les notations suivantes pour les contraintes : les c_i représentent les contraintes algébriques, les c_{di} représentent les contraintes différentielles, les c_{mi} représentent les équations de mesure. Nous examinons le cas des contraintes différentielles

c_{d1} et c_{d2} pour savoir s'il est possible de les prendre en causalité dérivée.

Concernant c_{d1} , on voit qu'il est possible d'utiliser la causalité dérivée : cela correspond à dériver y_1 pour aller vers u . En revanche, on se rend compte que cela n'est pas possible pour la contrainte différentielle c_{d2} . En effet, celle-ci peut apparaître :

- dans la boucle différentielle constituée des contraintes c_{d2} et c_2 (sens naturel du système) ;
- dans la boucle différentielle constituée des contraintes c_{d2} , c_3 , et c_{m2} (en inversant le système à partir de y_2) ;
- comme contrainte redondante, pour tester la cohérence des estimations de x_2 et de $\dot{x}_2$.

Pour les deux premiers cas, la causalité intégrale est obligatoire. Comme l'inversibilité de la contrainte redondante (dernier cas) n'a, dans nos critères, aucune importance, on considèrera que c_{d2} va toujours apparaître dans une boucle, quel que soit le couplage, et, en conséquence, on imposera la causalité intégrale pour c_{d2} . La raison, que nous développerons par la suite, qui permet d'identifier que c_{d2} ne peut être utilisé que dans une boucle, c'est qu'il est impossible d'atteindre x_2 pour ensuite calculer $\dot{x}_2$. Ici, un chemin permettant d'atteindre x_2 utilise nécessairement $\dot{x}_2$, ce qui crée une boucle.

En résumé, afin de contraindre la recherche des sous-systèmes surveillables, nous nous proposons, comme dans l'exemple de la figure 2.6, de placer la causalité intégrale dès le départ sur les contraintes différentielles dont on sait qu'elles vont toujours apparaître dans une boucle. L'idée-clef est l'étude de l'atteignabilité des variables.

b) Atteignabilité des variables

La notion d'*atteignabilité* des variables, que nous proposons dans ce travail, repose sur la notion de sous-graphe juste-déterminé.

Définition 2.2.1 (Sous-graphe juste-déterminé). Un sous-graphe $(M, var_X(M), \Gamma)$ est juste-déterminé si l'ensemble des contraintes M vérifie $M^0 = M$.

Définition 2.2.2 (Sous-graphe juste-déterminé orientable). Un sous-graphe juste-déterminé orientable possède un couplage complet n'utilisant que des arcs inversibles.

Définition 2.2.3 (Variable atteignable). Une variable x est dite *atteignable* s'il existe un sous-graphe juste-déterminé orientable du graphe biparti la contenant.

c) Inversibilité d'une contrainte dynamique

Le lemme suivant donne un critère permettant d'identifier si une contrainte différentielle peut apparaître en causalité dérivée dans un bloc de Hall de taille 1, auquel cas la causalité dérivée est possible.

Lemme 2. Une contrainte dynamique I liant une variable intégrale et une variable dérivée : $x_i = I(x_d)$ ne peut pas être prise en causalité dérivée si x_i n'est pas atteignable sur le graphe biparti modifié de la façon suivante :

- retrait de la contrainte I ;
- placement de non-inversibilités sur tous les arcs liés à x_d : sur la matrice d'incidence S , la colonne x_d est mise à -1 : $S(:, x_d) = -|S(:, x_d)|$.

2.2. Détermination des inversibilités sur les contraintes différentielles

Preuve 2. S'il est impossible d'atteindre x_i alors que x_d est inconnue et sans utiliser I , alors il est impossible d'utiliser I pour calculer x_d . Deux cas peuvent se produire : soit des non-inversibilités empêchent d'atteindre x_i , soit x_d et I sont nécessaires pour atteindre x_i . Dans chaque cas, la causalité de I sera intégrale. $\square$

L'algorithme suivant teste donc l'atteignabilité des variables dérivées l'une après l'autre ; il faut noter que dès que l'on impose une causalité intégrale, le graphe est contraint dans son orientation, ce qui peut rendre non atteignable une variable préalablement trouvée atteignable. Il faut donc boucler l'algorithme jusqu'à ce qu'il n'y ait plus de modifications d'orientation.

Algorithm 2 $S = \text{Definition_causalite}(S)$

```

1: Identifier l'ensemble des contraintes différentielles  $I$  sur le graphe  $S$ 
2: Initialiser la variable  $MODIF = 1$ 
3: while  $MODIF$  do
4:    $MODIF = 0$ 
5:   for all  $I(k)$  dont la causalité dérivée est permise do
6:     Identifier la variable intégrale  $x_i$  et la variable dérivée  $x_d$  de  $I(k)$ 
7:     if  $I(k)$  ne vérifie pas le lemme 2 then
8:        $I(k)$  n'est pas inversible par rapport à  $x_d$  : modifier  $S$ 
9:        $MODIF = 1$ 
10:    end if
11:  end for
12: end while
13: Renvoyer  $S$ 

```

Nous verrons à la section 2.3 l'outil utilisé pour tester l'atteignabilité des variables sur un graphe pré-orienté.

d) Illustration de l'algorithme 2

Dans l'exemple qui suit (figure 2.7), il est supposé que certaines fonctions ne sont pas inversibles ; elles sont indiquées dans la matrice d'incidence à l'aide de « -1 », et sur la figure, à l'aide d'une flèche précisant le sens dans lequel elles doivent être utilisées. Ainsi, c_{m1} n'est pas inversible par rapport à x_1 , ni c_3 par rapport à $\dot{x}_2$. A première vue, il y a trois sous-systèmes surveillables : le premier liant u et y_1 , le second liant u et y_2 , et le troisième liant y_1 and y_2 . Nous allons montrer que la sous-partie du graphe comprise entre x_1 et y_2 doit être prise en causalité intégrale, et que, par conséquent, il n'est pas possible de générer de résidu liant y_1 et y_2 par propagation – ce qui n'est pas évident à première vue, puisqu'il n'y a qu'une non-inversibilité définie sur cette sous-partie.

L'algorithme 2 va procéder de la façon suivante :

- le premier intégrateur c_{d1} est examiné : sa sortie est atteignable à partir d' y_2 . En effet, le sous-graphe entre y_2 et x_1 est orientable. Aucune modification n'est faite ;
- le deuxième intégrateur c_{d2} est examiné : coupler c_{d2} à $\dot{x}_2$ est impossible, car cela induit une boucle différentielle. L'algorithme l'identifie en testant qu'il n'est pas possible d'atteindre x_2 sans utiliser c_{d2} ni $\dot{x}_2$. La causalité intégrale est donc obligatoire ;
- le premier intégrateur est ré-examiné : à cause des non-inversibilités de c_{d2} et c_3 , sa sortie n'est pas atteignable ($\dot{x}_2$ ne pouvant être couplée en utilisant un arc inversible), par consé-

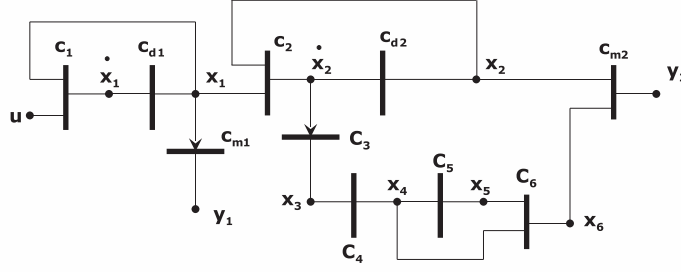


FIGURE 2.7 – Illustration de l’algorithme 2

quent, c_{d1} sera toujours utilisé en causalité intégrale.

e) Complexité de l’algorithme 2 de définition de la causalité

Lemme 3. *L’algorithme 2 de définition de la causalité a une complexité dans le pire des cas en $\mathcal{O}(q^2.d.n^{2,5})$, où n est le nombre de contraintes, d le degré de redondance structural, et q le nombre de contraintes dynamiques.*

Preuve 3. Nous verrons à la section 2.3 que tester l’atteignabilité d’une variable a une complexité dans le pire des cas en $\mathcal{O}(d.n^{2,5})$. Dans le pire des cas, la boucle « WHILE » de l’algorithme 2 est parcourue q fois : on reparcourt cette boucle si une causalité a été modifiée, et il y a au plus q causalités à modifier. Si cette boucle « WHILE » est parcourue q fois, cela veut dire qu’une seule causalité est modifiée à chaque passage dans la boucle. Par conséquent :

- au premier passage, les q contraintes différentielles sont examinées. Ce premier passage a donc une complexité en $\mathcal{O}(q.d.n^{2,5})$;
- au deuxième passage, $q - 1$ contraintes différentielles sont examinées. Ce deuxième passage a donc une complexité en $\mathcal{O}(q - 1.d.n^{2,5})$;
- ...
- au dernier passage, une seule contrainte différentielle est examinée. Ce dernier passage a donc une complexité en $\mathcal{O}(d.n^{2,5})$.

Comme $\sum_{k=1}^q k \leq q^2$, on a, au final, la complexité énoncée dans le lemme. $\square$

2.3 Décomposition canonique modifiée en intégrant des inversibilités

Les travaux et résultats présentés dans cette section ont fait l’objet de la publication (de Flaugergues *et al.*, 2009b).

a) Objectifs.

L’algorithme de décomposition de Dulmage-Mendelsohn ne tient pas compte des contraintes d’inversibilité. Par conséquent, certaines parties du système peuvent être dites surveillables alors qu’elles ne le sont pas en réalité, en raison des problèmes de calculabilité : la décomposition

2.3. Décomposition canonique modifiée en intégrant des inversibilités

canonique surestime les sous-systèmes surveillables et observables. Notre algorithme se propose de mettre en évidence les parties surestimées.

L'objectif de la décomposition canonique modifiée est, à partir de la décomposition d'origine (S^+, S^0, S^-) , de redéfinir chacune des trois parties du système, conformément à la définition de sous-système orientable donnée, afin que la décomposition modifiée vérifie les points suivants :

1. la partie surveillable modifiée S_{mod}^+ ne contient que les contraintes qui apparaîtront dans un sous-système orientable ;
2. la partie observable modifiée S_{mod}^0 ne contient que les variables atteignables à partir des variables connues, et qui ne sont pas dans S_{mod}^+ ;
3. la partie non-observable modifiée contient toutes les contraintes restantes.

En conséquence, nous avons $S_{mod}^+ \subseteq S^+$, et $S_{mod}^- \supseteq S^-$.

b) Exemple introductif

L'exemple présenté à la figure 2.8 est un exemple académique d'un système avec trois capteurs c_{m1} , c_{m2} , et c_{m3} , dont on supposera que c_{m2} et c_{m3} sont non-inversibles. c_{m2} et c_{m3} peuvent être par exemple des détecteurs, c'est-à-dire des capteurs qui détectent un état. On supposera également que l'entrée u est connue et l'entrée v est inconnue.

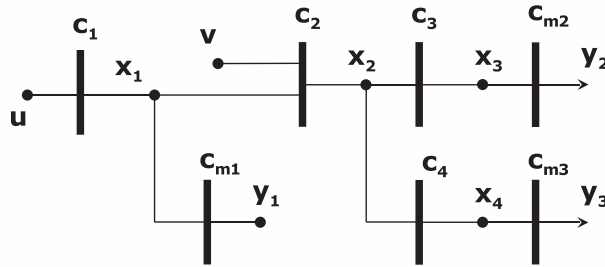


FIGURE 2.8 – Intégration de la notion de sous-système orientable

Sans prendre en compte les non-inversibilités, deux sous-systèmes surveillables peuvent être trouvés : il s'agit des sous-systèmes $\{c_1, c_{m1}\}$, liant u et y_1 , et $\{c_{m2}, c_3, c_4, c_{m3}\}$, liant y_2 et y_3 . La partie surveillable est constituée de la réunion de ces deux ensembles : $S^+ = \{c_1, c_3, c_4, c_{m1}, c_{m2}, c_{m3}\}$. En prenant en considération, à ce niveau, les non-inversibilités du graphe, on se rend compte qu'en réalité, le sous-système surveillable $\{c_3, c_4, c_{m2}, c_{m3}\}$ liant y_2 et y_3 n'est pas calculable du fait de l'impossibilité d'inverser c_{m2} ou c_{m3} : ce sous-système n'est pas orientable. Par conséquent, la partie surveillable du système, au sens des sous-systèmes orientables, n'est constitué que des contraintes c_1 et c_{m1} . Il est possible, contrairement à notre raisonnement sur cet exemple, qui est un raisonnement « a posteriori », de trouver ce sous-système « a priori », sans avoir besoin de connaître les sous-systèmes surveillables : l'algorithme de cette décomposition canonique modifiée par les inversibilités est expliqué par ce qui suit.

c) Identification de la partie surveillable S_{mod}^+ .

Donnons en premier lieu un algorithme basique pour calculer S_{mod}^+ , procédant en deux étapes :

1. décomposition canonique et recherche exhaustive des sous-systèmes surveillables orientables M_i ;
2. exclusion de S^+ des contraintes qui n'apparaissent dans aucun M_i .

Algorithm 3 $S_{mod}^+ = Systeme_surveillable_modifie_O(S)$

- 1: Calculer S^+
 - 2: Trouver tous les sous-systèmes surveillables M_i dans S^+
 - 3: $MS = \phi$
 - 4: **for all** M_i **do**
 - 5: **if** M_i est orientable **then**
 - 6: $MS = MS \cup M_i$
 - 7: **end if**
 - 8: **end for**
 - 9: Former $C_{mod}^+ = \bigcup_{M_j \in MS} C(M_j)$, où $C(M_j)$ représente l'ensemble des contraintes dans M_j
 - 10: Construire V_{mod}^+ , l'ensemble des variables apparaissant dans C_{mod}^+
 - 11: Retourner $S_{mod}^+ = (C_{mod}^+, V_{mod}^+)$
-

Cet algorithme utilise les inversibilités a posteriori, c'est-à-dire qu'une fois que tous les sous-systèmes surveillables M_i ont été trouvés, nous testons s'ils sont orientables. Nous proposons une méthode pour trouver le même S_{mod}^+ que dans l'algorithme 3, mais en utilisant les inversibilités a priori.

Définissons $S^{+,*}$ comme le sous-système surveillable S^+ , dans lequel les arcs non-inversibles ont été supprimés. Soit $(S_+^{+,*}, S_0^{+,*}, S_-^{+,*})$ la DM-décomposition de $S^{+,*}$. $S_-^{+,*}$ donne des informations sur l'atteignabilité des variables de S^+ .

Lemme 4. Si $S_-^{+,*} = \phi$,

1. toutes les variables inconnues de S^+ sont atteignables ;
2. $S_{mod}^+ = S^+$.

Preuve 4. Si $S_-^{+,*} = \phi$, il existe un système juste-déterminé orientable qui contient chaque variable : cela signifie que chaque variable est atteignable. Ce système correspond à un couplage de S^+ et fournit $(n - m)$ sous-systèmes surveillables orientables M_j , (n, m étant respectivement le nombre de contraintes et le nombre de variables de S^+). Chaque contrainte de S^+ apparaît au moins dans un sous-système orientable, i.e., avec les notations de l'algorithme 3, $\bigcup_j C(M_j) = C(S^+)$. □

Lemme 5. Si $S_-^{+,*} \neq \phi$, si $x \in S_-^{+,*}$, alors x n'est pas atteignable à partir des variables connues.

Preuve 5. Si $x \in S_-^{+,*}$ était atteignable à partir d'un sous-ensemble de variables connues, un

2.3. Décomposition canonique modifiée en intégrant des inversibilités

système juste-déterminé de la forme suivante existerait :

$$\begin{cases} x_1 = f_1(x_2, \dots, x, z_1, \dots, z_k) \\ \dots \\ x_{p-1} = f_{p-1}(x_1, \dots, x_{p-2}, x, z_1, \dots, z_k) \\ x = f_p(x_1, \dots, x_{p-1}, z_1, \dots, z_k) \end{cases} \quad (2.2)$$

Retirer les arcs non-inversibles et réaliser une décomposition canonique impliquerait que x appartienne à $S_0^{+,*}$ ou $S_+^{+,*}$. $\square$

Notons qu'une contrainte dans laquelle une variable non atteignable apparaît est inutilisable. Ces deux lemmes posés, l'algorithme complet pour modifier la partie surveillable $S^+ = (C^+, V^+, E^+)$ s'ensuit (algorithme 4).

Algorithm 4 $S_{mod}^+ = \text{Systeme_surveillable_modifié_1}(S)$

- 1: Calculer S^+ (figure 2.9)
 - 2: Construire $S^{+,*}$ en retirant les arcs inversibles
 - 3: Calculer la DM-décomposition de $S^{+,*}$
 - 4: **if** $S^{+,*} = \emptyset$ **then**
 - 5: $S_{mod}^+ = S^+$
 - 6: **else**
 - 7: Identifier l'ensemble des variables non atteignables V_{NA}
 - 8: Identifier l'ensemble de contraintes inutilisables C_{NA} , associé à V_{NA}
 - 9: Former la décomposition de S^+ en $S_A^+ = (C^+ \setminus C_{NA}, V^+ \setminus V_{NA})$, et $S_{NA}^+ = (C_{NA}, V_{NA})$ (figure 2.10){Notons que cette décomposition est une décomposition triangulaire inférieure par blocs}
 - 10: $S_{mod}^+ = \text{Systeme_surveillable_modifié_1}(S_A^+)$ (figure 2.11)
 - 11: **end if**
 - 12: Retourner S_{mod}^+
-

Lors de l'appel récursif de la fonction *Systeme_surveillable_modifié_1* avec S_A^+ (la sous-partie « atteignable » de S^+ – voir étape 9), il est important de réaliser l'étape 1 de nouveau car S_A^+ n'est pas nécessairement surcontraint.

A la fin de cet algorithme, nous avons non seulement défini S_{mod}^+ , mais nous avons également décomposé S^+ en $S_{mod}^+ \cup_i S_i^+$, qui est une décomposition triangulaire inférieure par blocs (figure 2.11). Les sous-systèmes S_i^+ sont soit un bloc inutilisable, correspondant à une sous-partie non atteignable, S_{NA}^+ (étape 8 de l'algorithme, figure 2.10), soit un bloc juste-déterminé ou même sous-déterminé résultant de la DM-Décomposition de S_A^+ (étape 1, figure 2.11).

d) Complexité du calcul de la partie surveillable modifiée.

Lemme 6. *Le calcul de la partie surveillable modifiée, décrit dans l'algorithme 4, a une complexité dans le pire des cas en $\mathcal{O}(d.n^{2,5})$, où n est le nombre de contraintes, et d le degré de redondance structurel.*

Preuve 6. Le calcul de la partie surveillable modifiée s'appuie sur le calcul préalable de la décomposition de Dulmage-Mendelsohn, dont la complexité est en $\mathcal{O}(n^{2,5})$ (Murota, 2000). Dans

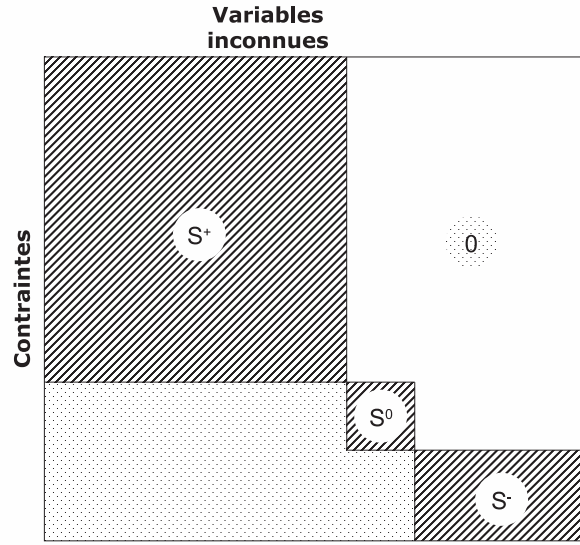


FIGURE 2.9 – *Systeme_surveillable_modifié* : étape 1

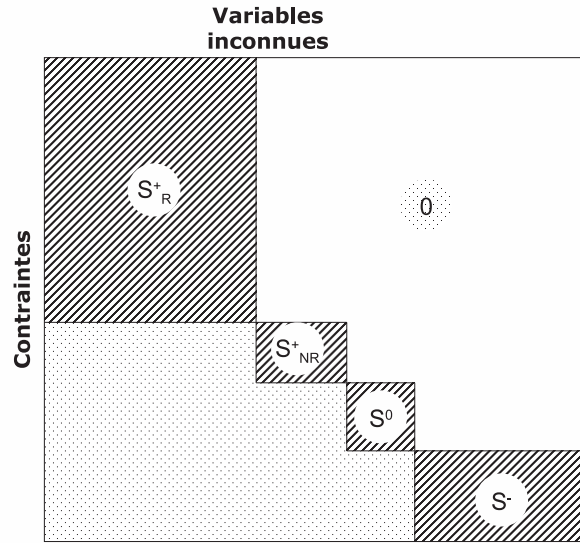


FIGURE 2.10 – *Systeme_surveillable_modifié* : étape 8

l'algorithme 4, une seconde décomposition de Dulmage-Mendelsohn est réalisée, sur le graphe dont on a retiré les arcs non-inversibles, dans le cas où la décomposition doit être modifiée. Puis l'algorithme est appelé récursivement jusqu'à ce que S_{mod}^+ soit calculé.

Notons S_i^+ le S^+ calculé à l'étape 1, au i^e appel de l'algorithme. Notons d_i le degré de redondance de S_i^+ . On a :

1. $d_1 = d$: il s'agit du calcul de la décomposition originale ;
2. $\forall i, d_i \geq 1$, si $S_i^+ \neq \phi$;
3. la suite (d_i) est strictement décroissante, i.e. $d_{i+1} < d_i$.

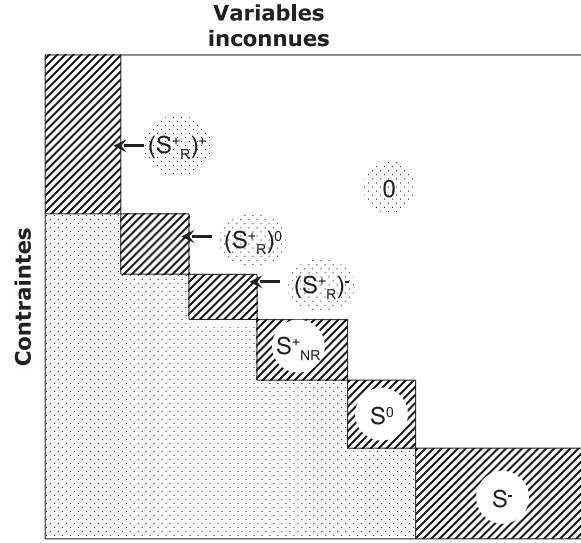


FIGURE 2.11 – *Système_surveillable_modifié* : étape 9

Par conséquent, il y a au plus d appels récursifs, d'où la complexité dans le pire des cas en $\mathcal{O}(d.n^{2,5})$.

Il nous reste à expliquer pourquoi la suite (d_i) est strictement décroissante : l'algorithme est appelé récursivement quand il y a des variables non atteignables $V_{NA} \neq \emptyset$, et des contraintes $C_{NA} \neq \emptyset$. Ces contraintes sont retirées de S^+ afin de former le système S_R^+ . Il a été montré dans (Krysander, 2006) que lorsque l'on retire une contrainte d'un système surcontraint, le degré de redondance décroît d'une unité. A fortiori, quand on retire plusieurs contraintes, le degré de redondance décroît strictement. $\square$

e) Identification du système observable modifié S_{mod}^0 .

Une fois le système décomposé sous la forme triangulaire inférieure par blocs donnée par l'algorithme 3, la partie observable modifiée est obtenue par :

1. décomposition de la partie observable originale en composantes de König-Hall (cette décomposition est unique) S_i^0 . La décomposition de Dulmage-Mendelsohn fournit les composantes de König-Hall ;
2. test de l'atteignabilité de chaque bloc, en construisant le diagramme de Hasse de la décomposition ;
3. test de l'atteignabilité des variables dans chaque bloc.

Clarifions ces trois points :

Critère 2.3.1. Un bloc S_i^+ ou S_i^0 est observable (pour la décomposition modifiée) si les deux conditions suivantes sont vérifiées :

- les variables du bloc sont atteignables, en supposant les variables externes au bloc connues ;
- les variables inconnues entrant dans le bloc sont elles-mêmes atteignables (atteignabilité du bloc).

Définissons la relation d'ordre $\succ$ sur l'ensemble des blocs $B_k = (\cup_i S_i^+) \cup (\cup_j S_j^0) \cup S^-$: nous notons $B_{k_1} \succ B_{k_2}$ s'il y a un arc liant une variable de B_{k_1} à une contrainte B_{k_2} . Le diagramme de Hasse est un diagramme :

- dont les sommets sont les B_k ;
- avec un arc de B_{k_1} à B_{k_2} si $B_{k_1} \succ B_{k_2}$.

En d'autres termes, le diagramme de Hasse est un ordonnancement des blocs qui indique leur ordre de résolution.

Lemme 7. *Si un bloc B_{k_1} n'est pas observable alors tout bloc B_{k_2} tel que $B_{k_1} \succ B_{k_2}$ n'est pas observable.*

Preuve 7. Cela correspond au second point du critère 2.3.1. □

L'algorithme servant à redéfinir la partie observable du système s'ensuit : il s'agit de l'algorithme 5.

Algorithm 5 $(S_{mod}^0, S_{mod}^-) = \text{Système_observable_modifié}(S, S_{mod}^+, B_k)$

Require: Calcul préalable de S_{mod}^+ et identification des B_k

```

1:  $S_{mod}^0 = \phi, S_{mod}^- = \phi$ 
2:  $R = \cup_k B_k$ 
3: while  $R \neq \phi$  do
4:   Prendre le premier  $B_k$ 
5:   Tester l'observabilité de  $B_k$ 
6:   if  $B_k$  est observable then
7:      $S_{mod}^0 = S_{mod}^0 \cup B_k$ 
8:   else
9:     Identifier tous les  $B_l$  tels que  $B_k \succ B_l$ 
10:     $S_{mod}^- = S_{mod}^- \cup (\cup_l B_l)$ 
11:     $R = R \setminus (\cup_l B_l)$ 
12:   end if
13: end while
14: Retourner  $S_{mod}^0, S_{mod}^-$ 

```

f) Illustration des algorithmes 4 et 5

Considérons le modèle structurel représenté à la figure 2.12, où les flèches représentent des non-inversibilités.

D'après la décomposition canonique classique, c_7 et x_7 apparaissent dans la partie juste-contrainte (observable). Toutes les autres contraintes et variables inconnues apparaissent dans la partie sur-contrainte. En examinant la partie sur-contrainte, trois sous-systèmes surveillables peuvent être trouvés, qui correspondent à trois résidus potentiels : le premier lie u et y_1 , le second lie u et y_2 , et le dernier lie y_1 et y_2 . Les deux premiers ne peuvent pas être générés, en raison des non-inversibilités, par propagation des variables connues. Par conséquent, les seules contraintes qui sont effectivement surveillables sont celles utilisées pour construire le dernier résidu (c_{m1} et c_{m2}). Intuitivement, nous pouvons également noter que :

- les variables inconnues x_1 et x_5 sont observables car elles peuvent être atteintes à partir de, respectivement, u et y_2 ;

2.3. Décomposition canonique modifiée en intégrant des inversibilités

- les variables inconnues x_2, x_3, x_4, x_7 ne sont pas observables, car les atteindre suppose d'utiliser c_2 ou c_5 pour calculer x_2 ou x_4 , ce qui est impossible.

Par la suite, nous appliquerons notre méthode afin de montrer comment trouver cette décomposition, que nous avons ici faite intuitivement.

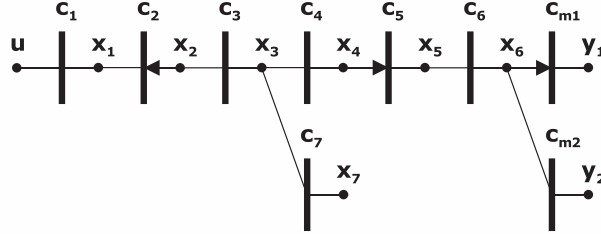


FIGURE 2.12 – Exemple illustratif de la décomposition canonique modifiée

En premier lieu, nous remarquons sur la figure 2.12 que, pour S^+ , il n'y a aucun couplage complet n'utilisant que des arcs inversibles, par conséquent la partie surveillable doit être redéfinie. Retirer les arcs inversibles et effectuer une décomposition canonique (figure 2.13) permet d'identifier des variables non atteignables : x_2, x_3, x_4 .

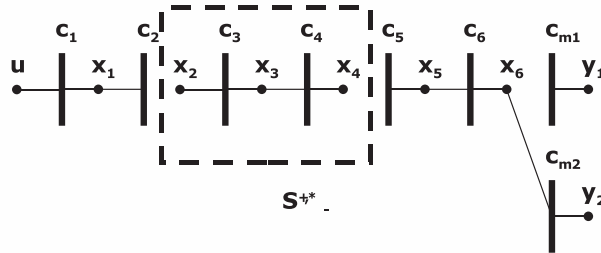


FIGURE 2.13 – Décomposition canonique de $S^{+,*}$

La partie sur-contrainte peut ainsi être décomposée en S_A^+ et S_{NA}^+ (figure 2.14), S_{NA}^+ étant composé :

- des variables non atteignables : x_2, x_3, x_4 ;
- des contraintes dans lesquelles une variable non atteignable apparaît : c_2, c_3, c_4, c_5 .

La méthode est ensuite appliquée récursivement sur S_A^+ : celui-ci est composé d'une partie sur-contrainte (c_{m1}, c_{m2}), et de deux parties observables non connectées c_1 , puis c_6 . La partie sur-contrainte ne doit pas être modifiée. La décomposition finale est donnée à la figure 2.15. Dans celle-ci, nous avons nommé $S_{mod}^+, B_0, B_1, B_2, S^0$ les cinq blocs définis par la décomposition.

Au paragraphe suivant, nous redéfinissons la partie observable.

B_0 est évidemment observable. B_1 est observable car :

1. les variables de B_1 sont atteignables, en supposant les variables externes à B_1 connues ;
2. $S_{mod}^+ \succ B_1$, et S_{mod}^+ est observable.

B_2 n'est pas observable car ses variables ne sont pas atteignables. Par suite, S^0 est non observable car $B_2 \succ S^0$. B_2 et S^0 constituent la partie sous-contrainte du modèle.

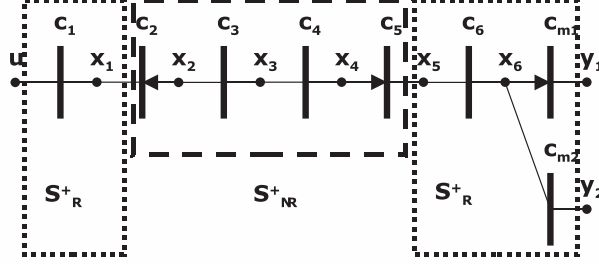


FIGURE 2.14 – Décomposition de S^+ en S_A^+ et S_{NA}^+

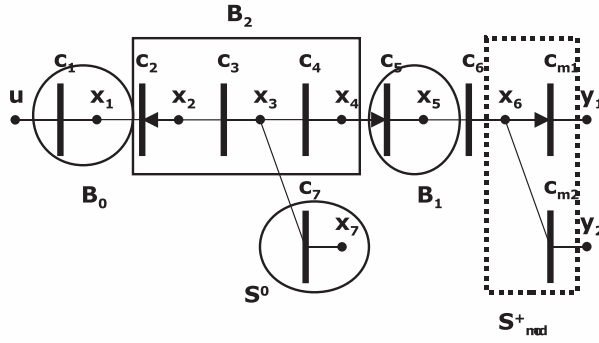


FIGURE 2.15 – Décomposition finale de S

g) Atteignabilité d'une variable

Par construction de la décomposition canonique modifiée, les variables atteignables sont dans S_{mod}^+ et S_{mod}^0 . On peut également tester l'atteignabilité d'une variable inconnue grâce au lemme suivant.

Lemme 8. Soient :

- S un modèle structurel comportant n contraintes $c_1, \dots, c_n$;
- x_k une variable inconnue ;
- S' construit à partir de S en rajoutant une contrainte c_{n+1} telle que $S'(n+1, k) = -1$.
 c_{n+1} est donc une contrainte de type détecteur sur x_k .

Les trois propositions suivantes sont équivalentes :

1. x_k est atteignable pour le modèle S .
2. x_k est une variable de S_{mod}^+ ou S_{mod}^0 .
3. x_k est une variable de $S_{mod}'^+$.

Preuve 8. Les deux premières propositions sont équivalentes, par construction de la décomposition canonique modifiée.

Par ailleurs, si x_k est une variable de S_{mod}^0 , alors il existe un sous-graphe juste-déterminé orientable G contenant x_k . En ajoutant une contrainte c_{n+1} non-inversible sur x_k , on forme un sous-système surveillable orientable : ce sous-système est constitué des contraintes de G et de c_{n+1} . Par conséquent, x_k est une variable de $S_{mod}^{'+}$.

Inversement, si x_k est une variable de $S_{mod}^{'+}$, alors il existe un sous-système orientable G contenant x_k . Si ce sous-système ne contient pas c_{n+1} alors x_k est dans S_{mod}^+ . Si ce sous-système contient c_{n+1} alors, en retirant c_{n+1} , on forme un sous-graphe juste-déterminé orientable. Par conséquent, x_k est une variable de S_{mod}^0 . $\square$

Les conséquences de ce lemme sont :

- qu'il est suffisant de calculer la partie surveillable d'un système modifié « S' » pour tester l'atteignabilité d'une variable ;
- que la complexité du test d'atteignabilité d'une variable est similaire à celle du calcul de la partie sur-contrainte, soit en $\mathcal{O}(d.n^{2,5})$.

2.4 Etudes a priori de l'isolabilité des fautes

2.4.1 Introduction

Nos critères pour l'étude a priori de l'isolabilité des défauts s'appuient sur une idée développée dans (Cocquempot *et al.*, 1991), qui consiste à identifier le sous-système robuste à certains paramètres. Dans le même esprit, un critère d'isolabilité est donné dans (Frisk and Krysander, 2007), que nous rappelons ci-dessous (lemme 9).

Définition 2.4.1 (Isolabilité). Deux fautes f_1 et f_2 sont isolables s'il existe au moins un sous-système surveillable permettant de les discriminer.

Lemme 9 (Isolabilité). Deux fautes f_1, f_2 sont isolables si et seulement si :

$$f_1 \in (S^+ \setminus \{f_2\})^+ \text{ (ce qui est montré équivalent à } f_2 \in (S^+ \setminus \{f_1\})^+)$$

Le critère original de (Frisk and Krysander, 2007) vaut pour un graphe non orienté par les non-inversibilités et indique si, pour deux fautes f_1 et f_2 , on peut trouver des sous-systèmes discriminants.

Nous adaptons ce critère pour prendre en compte la notion de sous-système orientable.

2.4.2 Analyse de l'isolabilité en tenant compte des contraintes d'inversibilité

Définition 2.4.2 (Isolabilité intégrant un critère d'orientation). Deux fautes f_1 et f_2 sont isolables s'il existe au moins un sous-système orientable discriminant pour f_1, f_2 .

Lemme 10 (Isolabilité intégrant un critère d'orientation). Deux fautes f_1, f_2 vérifient la définition 2.4.2 si et seulement si :

$$f_1 \in (S_{mod}^+ \setminus \{f_2\})_{mod}^+ \text{ ou } f_2 \in (S_{mod}^+ \setminus \{f_1\})_{mod}^+$$

Preuve 9. Si $f_1 \in (S_{mod}^+ \setminus \{f_2\})_{mod}^+$, alors f_1 est dans un système surdéterminé orientable qui ne contient pas f_2 : il existe par conséquent un sous-système orientable contenant f_1 et pas f_2 . f_1 et f_2 jouant des rôles symétriques dans le lemme, on en déduira l'inverse si $f_2 \in (S_{mod}^+ \setminus \{f_1\})_{mod}^+$. Réciproquement, si f_1, f_2 vérifient la définition 2.4.2, alors soit il existe un sous-système orientable contenant f_1 et pas f_2 , soit il existe un sous-système orientable contenant f_2 et pas f_1 . Dans le premier cas, il est nécessaire que $f_1 \in (S_{mod}^+ \setminus \{f_2\})_{mod}^+$. Dans le second cas, il est nécessaire que $f_1 \in (S_{mod}^+ \setminus \{f_2\})_{mod}^+$. $\square$

Ce critère d'isolabilité dépend des non-inversibilités qui ont été définies. On pourra l'utiliser dans deux cas : pour la causalité intégrale, comme pour la causalité mixte, après avoir identifié les contraintes différentielles à n'utiliser qu'en causalité intégrale. Rappelons que l'hypothèse sur laquelle est fondée ce critère est qu'un sous-système calculable est un sous-système orientable.

Dans la littérature, on notera l'article (Frisk *et al.*, 2010), qui présente des analyses d'isolabilité a priori pour trois hypothèses :

- causalité mixte ;
- causalité intégrale ;
- causalité dérivée.

A la différence de notre travail, les auteurs considèrent qu'il n'y a pas de problèmes d'inversibilités sur les fonctions algébriques, et que les boucles algébriques sont toutes solvables.

2.5 Intégration des non-inversibilités dans la recherche exhaustive des sous-systèmes surveillables

Nous proposons trois algorithmes dans cette section. Tous trois permettent de rechercher des sous-systèmes surveillables en prenant en considération les non-inversibilités du modèle. Le premier repose sur le principe de recherche des couplages complets, décrit dans (Düstegör *et al.*, 2004a) ; les deux autres réutilisent l'idée de parcours d'un arbre, exposée dans (Krysander *et al.*, 2008). Les deux premiers permettent d'obtenir les sous-systèmes orientables, le troisième fait l'hypothèse de causalité intégrale et ne trouve que les sous-systèmes validés par les inversibilités.

Illustrons tout d'abord sur deux exemples les motivations de ces algorithmes.

a) Exemple introductif pour la recherche des sous-systèmes surveillables orientables

Nous reprenons, pour illustrer les deux premiers algorithmes, l'exemple présenté à la figure 2.8, en supposant cette fois que l'entrée v est connue. Cet exemple est rappelé à la figure 2.16.

Six sous-systèmes surveillables seraient trouvés par une recherche exhaustive. Ils lient :

- u et y_1 ,
- u, v , et y_2 ,
- u, v , et y_3 ,
- v, y_1 , et y_2 ,
- v, y_1 , et y_3 ,

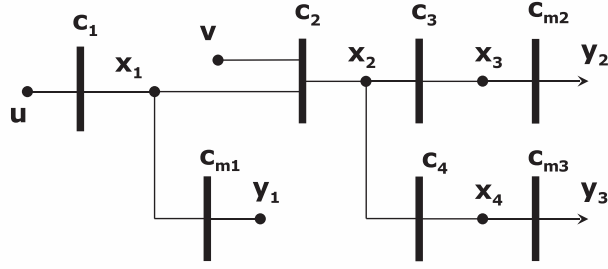


FIGURE 2.16 – Recherche des sous-systèmes orientables

– y_2 et y_3 .

Faire une recherche exhaustive des sous-systèmes orientables permet de ne trouver que les cinq premiers sous-systèmes. Le dernier n'est pas un système orientable, donc ne permet pas de générer un résidu, car c_{m2} et c_{m3} ne sont pas inversibles.

b) Exemple introductif pour la recherche des sous-systèmes surveillables validés (hypothèse de causalité intégrale)

Afin d'illustrer le troisième algorithme, qui permet de trouver les sous-systèmes surveillables validés par les inversibilités, nous utilisons l'exemple de la figure 2.17. Il s'agit également d'un exemple académique d'un système avec deux capteurs c_{m1} , c_{m2} , le premier étant supposé non-inversible. La contrainte nommée « c_d » est une contrainte différentielle, pour laquelle on a imposé la causalité intégrale.

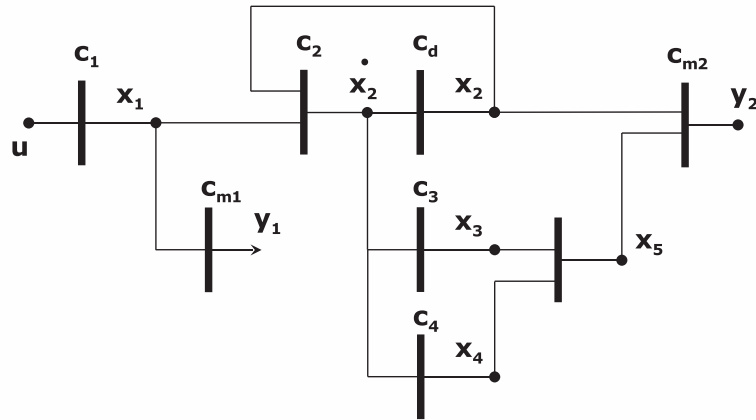


FIGURE 2.17 – Intégration de la notion de sous-système validé par les inversibilités

Une analyse structurelle « classique » fournirait trois sous-systèmes surveillables, reliant u à y_1 , u à y_2 et y_1 à y_2 .

Le sous-système surveillable liant y_1 et y_2 n'est pas validé par les inversibilités. En effet, soit les couplages imposent une orientation incohérente avec les non-inversibilités définies sur c_{m1} et

sur c_d , soit ils font apparaître une boucle algébrique (constituée par les contraintes c_3, c_4, c_5). De fait, il n'existe aucun couplage validé par les inversibilités. L'algorithme de recherche exhaustive des sous-systèmes validés par les inversibilités, qui sera décrit à la section 2.5.3, ne fournira par conséquent que les deux autres sous-systèmes.

Remarquons par ailleurs que cet exemple illustre la différence entre la notion de sous-système orientable et celle de sous-système validé par les inversibilités : en effet, le sous-système liant y_1 et y_2 n'est pas validé par les inversibilités mais il est orientable, car la boucle algébrique est orientable. Comme nous l'avons dit précédemment, la notion de sous-système orientable est une notion moins forte que celle de sous-système validé par les inversibilités.

2.5.1 Algorithme 1 : simplification du système et recherche des couplages complets orientables

Les principes de cet algorithme ont été énoncés dans la publication (de Flaugergues *et al.*, 2009a).

Cet algorithme réutilise en partie la simplification du système proposée dans (Krysander *et al.*, 2008). Nous appelons macro-graphe le graphe simplifié résultant de cette procédure. Cependant, en simplifiant le système, la notion d'inversibilité est perdue. Nous proposons donc d'ajouter à la première procédure, que nous reprenons, une seconde procédure, qui a pour but de redéfinir les non-inversibilités pour le macro-graphe. Puis nous proposons un algorithme adapté des principes énoncés dans (Düstegör *et al.*, 2004a). En effet, le corps de l'algorithme s'appuie sur la recherche des couplages complets :

- sur l'ensemble des variables du système surdéterminé S^+ ,
- qui n'utilisent que des arcs inversibles.

a) Définition du macro-graphe

Nous appelons macro-graphe un graphe réduit du système surdéterminé obtenu en simplifiant des systèmes de contraintes, ce qui résulte en la suppression de certaines variables inconnues. Une fusion de contraintes est appelée macro-contrainte, et les variables non supprimées, restantes, sont appelées variables externes.

Soit c_i une contrainte de S^+ . Le macro-graphe est obtenu en construisant les macro-systèmes MS_i comme suit :

Définition 2.5.1 (Macro-système). Le macro-système MS_i associé à une contrainte c_i est le système ayant C_i pour ensemble de contraintes et $X_{int,i}$ pour ensemble de variables, C_i et $X_{int,i}$ étant définis de la façon suivante :

- $C_i = C^+ \setminus C_i^+$;
- $X_{int,i} = X^+ \setminus X_i^+$.

où C_i^+ et X_i^+ sont respectivement l'ensemble des contraintes et des variables du système $S_i^+ = (S^+ \setminus \{c_i\})^+$.

MS_i possède une contrainte de plus que de variables :

$$\text{Card}(C_i) = \text{Card}(X_{int,i}) + 1. \quad (2.3)$$

Définition 2.5.2 (Variable interne à un macro-système). Une variable de $X_{int,i}$ est dite interne à MS_i .

Définition 2.5.3 (Variable externe à un macro-système). Une variable est dite externe au macro-système MS_i si elle apparaît dans une des contraintes de C_i du macro-système et n'est pas interne.

Définition 2.5.4 (Macro-contrainte). La macro-contrainte MC_i associée à un macro-système MS_i est la contrainte liant les variables externes à MS_i .

Une macro-contrainte MC_i résulte de la simplification des contraintes de C_i composant MS_i . Cette simplification correspond à l'élimination des variables $X_{int,i}$ à partir des contraintes de C_i . En conséquence, les variables internes à MS_i n'apparaissent pas dans MC_i .

Définition 2.5.5 (Macro-graphe). Le macro-graphe d'un système est un graphe biparti $MG = (V = X_{ext} \cup Z, MC, \Gamma')$, où :

- X_{ext} est l'ensemble des noeuds correspondant aux variables inconnues externes, $X_{ext} = \{x_{ext,i}\}_{i=1\dots m'}$;
- Z est l'ensemble des noeuds correspondant aux variables connues ;
- $MC = \{MC_i\}_{i=1\dots n'}$ est l'ensemble des noeuds correspondant aux macro-contraintes ;
- $\Gamma' = \{(MC_i, v_j) \mid v_j \text{ est une variable externe à } MS_i, \text{ connue ou inconnue}\}$ est l'ensemble des arcs.

Ces concepts ont été exposés dans (Krysander *et al.*, 2008). Calculer les macro-systèmes MS_i permet de réordonner les lignes et les colonnes de S^+ , et d'écrire ainsi le système surveillable sous la forme représentée sur la figure 2.18. Cette figure résume par ailleurs les concepts introduits. La simplification des contraintes en macro-contraintes et la suppression des variables internes fournit le macro-graphe du modèle structurel.

Une recherche exhaustive des sous-systèmes surveillables donne les mêmes résultats, qu'elle soit menée sur le macro-graphe ou sur le graphe original.

b) Contraintes d'inversibilité du macro-graphe

Lorsque l'on construit le macro-graphe, les inversibilités définies sont perdues ; dans cette section, les inversibilités des macro-contraintes par rapport aux variables externes sont définies.

Soit $C_i = \{c_i^j\}_{j=1\dots p+1}$ un ensemble de contraintes associé à un macro-système MS_i . De plus, soient $\{x_{int,i}^j\}_{j=1\dots p}$ les variables internes à MS_i . La macro-contrainte MC_i associée à MS_i est une contrainte entre variables externes :

$$MC_i(\dots, x_{ext,k}, \dots, u, y) = 0$$

Définir l'inversibilité de MC_i par rapport à $x_{ext,k}$ revient à répondre à la question suivante : connaissant toutes les variables externes de MC_i sauf $x_{ext,k}$, est-il possible de déterminer $x_{ext,k}$?

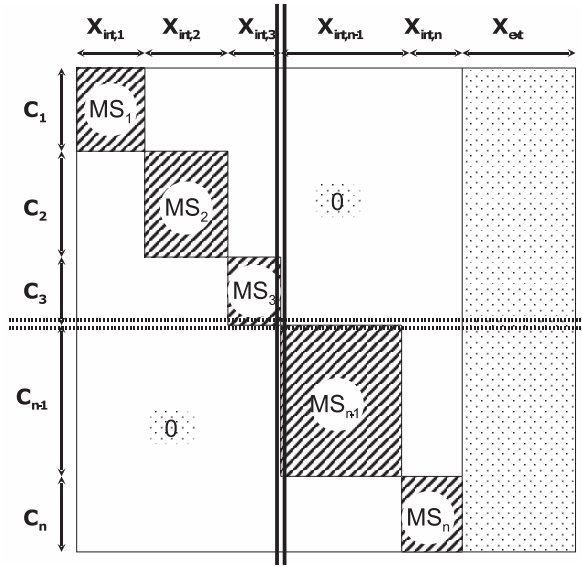


FIGURE 2.18 – Macro-graphe de la partie surveillable

Pour répondre à cette question, réécrivons $MC_i(\dots, x_{ext,k}, \dots, u, y) = 0$ comme suit :

$$\begin{cases} c_i^1(x_{int,i}^1, \dots, x_{int,i}^p, x_{ext,k}, u, y, (x_{ext,l})_{l \neq k}) = 0 \\ \dots \\ c_i^{p+1}(x_{int,i}^1, \dots, x_{int,i}^p, x_{ext,k}, u, y, (x_{ext,l})_{l \neq k}) = 0 \end{cases} \quad (2.4)$$

L'inversibilité de MC_i par rapport $x_{ext,k}$ peut être définie comme l'atteignabilité de $x_{ext,k}$ dans (2.4), $u, y, (x_{ext,l})_{l \neq k}$ étant considérées connues. De cette façon, les inversibilités peuvent être étendues au macro-graphe : l'arc $(MC, x_{ext,k})$ du macro-graphe est :

- inversible si le modèle structurel de (2.4) est orientable ;
- non inversible sinon.

Dans ce qui suit, nous réutilisons le principe de l'algorithme de (Düstegör *et al.*, 2004a), qui recherche tous les couplages complets sur un graphe biparti sur-contraint : il y en a C_n^m , n étant le nombre de contraintes, m étant le nombre de variables. En prenant la notion d'inversibilité en compte, certains couplages sont obligatoires : en effet, notre critère de sous-système orientable interdit de coupler des arcs non-inversibles. Sur la partie sur-contrainte S_{mod}^+ , les couplages obligatoires sont identifiés de la façon suivante :

Lemme 11 (Couplages obligatoires). *Soit $S_{mod}^{+,*}$ défini comme S_{mod}^+ , où les arcs non inversibles ont été supprimés. Il est obligatoire, pour qu'un couplage ne contienne pas d'arc non inversible, de coupler les contraintes de C_{obl} aux variables de V_{obl} , où C_{obl}, V_{obl} , sont définis comme suit :*

- C_{obl} est l'ensemble des contraintes de la partie juste-contrainte de $S_{mod}^{+,*}$;
- V_{obl} est l'ensemble des variables de la partie juste-contrainte de $S_{mod}^{+,*}$.

Preuve 10. Notons que $S_{mod}^{+,*}$ ne possède pas de partie sous-contrainte car, par construction, S_{mod}^+ possède un couplage complet composé d'arcs inversibles. $S_{mod}^{+,*}$ a donc une partie observable (éventuellement vide) et une partie surveillable.

Dans le cas où C_{obl}, V_{obl} ne sont pas vides, si une variable de V_{obl} n'est pas couplée aux contraintes

2.5. Intégration des non-inversibilités dans la recherche exhaustive des sous-systèmes surveillables

de C_{obl} , alors elle doit être couplée aux contraintes de $C_{mod}^+ \setminus C_{obl}$. Comme $S_{mod}^{+,*}$ ne possède pas de partie sous-contrainte, $C_{mod}^+ \setminus C_{obl}$ représente les contraintes de la partie sur-contrainte de $S_{mod}^{+,*}$. D'autre part, comme la décomposition de $S_{mod}^{+,*}$ est triangulaire inférieure par blocs, il n'y a pas d'arc inversible liant une variable de V_{obl} avec une contrainte de $C_{mod}^+ \setminus C_{obl}$. $\square$

Ces couplages obligatoires étant identifiés, ils peuvent servir de base à la construction de tous les couplages : il y en a $C_{n-m'}^{m-m'}$ à considérer ($m' = |V_{obl}| = |C_{obl}|$). Ceci est réalisé par l'algorithme 6.

Algorithm 6 $M = Recherche_exhaustive_1(S_{mod}^+)$

Require: Calcul préalable de S_{mod}^+ , de taille $n * m$

- 1: $M = \phi$
 - 2: Former le système $S_{mod}^{+,*}$ en retirant les arcs inversibles
 - 3: Identifier la partie observable $S_{obl} = (C_{obl}, V_{obl})$ de taille $m' * m'$, et la partie surveillable S_L de taille $n - m' * m - m'$ de $S_{mod}^{+,*}$
 - 4: **for** chaque bloc B carré de taille $m - m'$ de S_L **do**
 - 5: **if** B est juste-contraint **then**
 - 6: Former le bloc juste-contraint B' formé en concaténant B et S_{obl}
 - 7: Identifier les blocs de Hall de B'
 - 8: Déterminer $(n - m)$ sous-systèmes surveillables, notées $M_{B'}$
 - 9: $M = M \cup M_{B'}$
 - 10: **end if**
 - 11: **end for**
 - 12: Retourner M
-

c) Illustration des algorithmes de cette section

A partir de l'exemple de la figure 2.19, il est possible de définir le macro-graphe de la figure 2.20. Sur cette dernière figure, les flèches indiquent les contraintes d'inversibilité du macro-graphe définies.

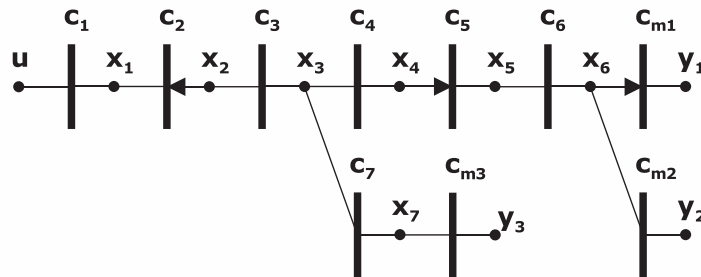


FIGURE 2.19 – Recherche des sous-systèmes orientables

Les deux variables inconnues du macro-graphe, qui comporte cinq contraintes, sont x_3 et x_6 . Il est obligatoire de coupler x_3 à MC_3 à cause des non-inversibilités sur MC_1 et MC_2 . Par conséquent, au lieu de former $C_5^2 = 10$ blocs carrés, il n'y en a que 4 à former. Deux d'entre eux

ne fournissent pas un couplage complet. Les deux couplages complets correspondent aux deux choix suivants pour la variable x_6 : soit x_6 est couplée à MC_2 , soit elle est couplée à c_{m2} . A l'aide des trois contraintes redondantes par couplage, les 4 sous-systèmes surveillables orientables sont trouvés. Ceux ci lient :

1. y_3 à u ;
2. y_3 à y_1 ;
3. y_3 à y_2 ;
4. y_2 à y_1 .

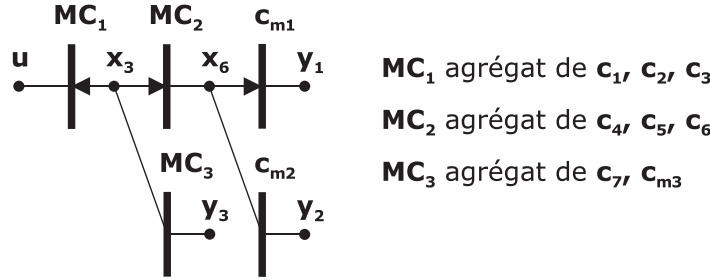


FIGURE 2.20 – Macro-graphe

d) Complexité des algorithmes de cette section

La complexité de l'algorithme 6 est en $\mathcal{O}(n^{2,5} \cdot C_{n-m'}^{m-m'})$. En effet, il y a $C_{n-m'}^{m-m'}$ couplages à examiner, et, pour chaque couplage, une décomposition en blocs de Hall, dont la complexité est en $\mathcal{O}(n^{2,5})$, est réalisée.

La complexité de la construction du macro-graphe est en $\mathcal{O}(n^{3,5})$: une décomposition de Dulmage-Mendelsohn, dont la complexité est en $\mathcal{O}(n^{2,5})$, est nécessaire pour identifier les macro-systèmes, et est réalisée au plus n fois.

Enfin, la redéfinition des inversibilités du macro-graphe consiste à rechercher un couplage complet sur les macro-systèmes. Pour un macro-système avec q_{MS} arcs et m_{MS} variables internes, l'algorithme de recherche d'un couplage complet d'Hopcroft-Karp (voir page 30) a une complexité en $\mathcal{O}(q_{MS} \cdot m_{MS}^{1,5})$.

Cette opération est répétée pour chaque arc de la macro-contrainte associée au macro-système. Au final, soient les notations suivantes pour le graphe $G = (V = X \cup Z, C, \Gamma)$:

- $q = |\Gamma|$, le nombre d'arcs ;
- $m = |X|$, le nombre de variables.

La complexité de la redéfinition des inversibilités sur le macro-graphe est en $\mathcal{O}(q^2 \cdot m^{1,5})$.

2.5.2 Algorithme 2 : Parcours de l'arbre des *PSO* en utilisant la décomposition canonique modifiée

Cet algorithme réalise la descente de l'arbre des *PSO* (section 2.1.2), en utilisant la décomposition canonique modifiée présentée à la section 2.3, au lieu d'utiliser la décomposition classique.

2.5. Intégration des non-inversibilités dans la recherche exhaustive des sous-systèmes surveillables

Cet algorithme retourne les sous-systèmes surveillables orientables.

a) Principe de l'algorithme

Comme l'algorithme *MSO* (section 2.1.2), notre algorithme est un algorithme dichotomique : l'espace des *MSO* orientables est scindé en deux, une fois une contrainte e choisie : l'ensemble des *MSO* orientables contenant e et l'ensemble des *MSO* orientables ne contenant pas e . Pour ce faire, au lieu de descendre l'arbre des *PSO* en utilisant la décomposition de Dulmage-Mendelsohn, on descend l'arbre des *PSO* en utilisant la décomposition modifiée. En effet, l'ensemble des *MSO* orientables ne contenant pas e est contenu dans l'ensemble $(S_{mod}^+ \setminus \{e\})_{mod}^+$.

Les améliorations discutées visant à ne trouver un *MSO* qu'une seule fois sont adaptables :

- le retrait des contraintes de $P \setminus (P \setminus \{e\})^+$ (décomposition originale) aboutit toujours au même *PSO* fils :

$$\forall e_1, \forall e_2, e_2 \in P \setminus (P \setminus \{e_1\})^+ \Rightarrow (P \setminus \{e_1\})^+ = (P \setminus \{e_2\})^+;$$

- si le sous-arbre issu du *PSO* P en retirant la contrainte e , qui a pour racine le *PSO* $(P \setminus \{e\})_{mod}^+$ a déjà été exploré, alors tous les *MSO* ne contenant pas e et contenus dans P ont été trouvés, donc tout autre *PSO* descendant de P ne contenant pas e est inintéressant. L'arbre sera, de même que dans l'algorithme original, tronqué.

Au final, le fonctionnement de l'algorithme modifié est décrit par les algorithmes 7 et 8.

Algorithm 7 $M = Recherche_exhaustive_2(S_{mod}^+)$

Require: Calcul préalable de S_{mod}^+

- 1: $P = \text{contraintes } c \in S_{mod}^+$
 - 2: $M = \text{Sub_Recherche_exhaustive_2}(P, P)$
 - 3: Retourner M
-

b) Illustration du fonctionnement et du gain de l'algorithme 7

Pour illustrer le fonctionnement de l'algorithme 7, nous nous appuyons sur l'exemple situé en haut de la figure 2.21, déjà traité auparavant. Nous nous plaçons dans le cas où nous recherchons les sous-systèmes surveillables ne contenant pas c_{m3} : nous calculons donc le sous-système $(S^+ \setminus \{c_{m3}\})_{mod}^+$, au lieu du sous-système $(S^+ \setminus \{c_{m3}\})^+$ (figure 2.21).

Sur la figure 2.21, il apparaît qu'en utilisant la décomposition canonique de Dulmage-Mendelsohn, on trouve un *PSO* dont le degré de redondance vaut 2. En utilisant la décomposition canonique modifiée, on trouve un *PSO* dont le degré de redondance vaut 1, c'est-à-dire qu'il s'agit d'un sous-système surveillable. La décomposition canonique modifiée permet de sauter des niveaux de l'arbre des *PSO* et ainsi d'accélérer le parcours de cet arbre.

c) Gain de complexité de l'algorithme 7

Dans ce paragraphe, nous allons estimer le gain de complexité de l'algorithme 7, par rapport à l'algorithme *MSO* original, décrit dans (Krysander *et al.*, 2008).

Algorithm 8 $M = \text{Sub_Recherche_exhaustive_2}(P, R)$

Require: P est un PSO

```

1: if  $P$  a un degré de redondance égal à 1 then
2:    $M = M \cup P$ 
3: else
4:   while  $R \neq \emptyset$  do
5:     Choisir une équation  $e$  de  $R : e = R(1)$ 
6:     Calculer  $P_1 = (P \setminus \{e\})^+$ 
7:     Calculer  $P_2 = (P \setminus \{e\})_{mod}^+$ 
8:     Calculer  $E_1 = P \setminus P_1$ 
9:     Calculer  $E_2 = P \setminus P_2$ 
10:    Calculer  $R_1 = R \setminus E_1$ 
11:    Calculer  $R_2 = R \setminus E_2$ 
12:     $M_s = \text{Sub\_Recherche\_exhaustive\_2}(P_2, R_2)$ 
13:     $M = M \cup M_s$ 
14:     $R = R_1$ 
15:   end while
16: end if
17: Retourner  $M$ 

```

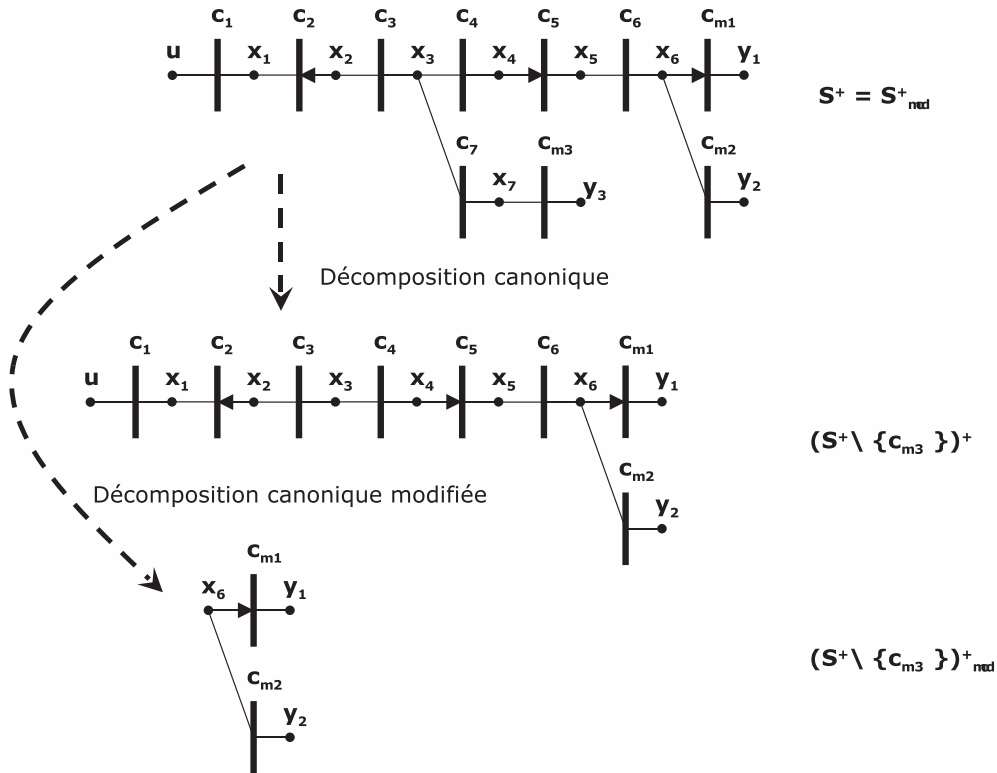


FIGURE 2.21 – Gain apporté par la décomposition canonique modifiée : exemple des sous-systèmes orientables ne contenant pas c_{m3}

2.5. Intégration des non-inversibilités dans la recherche exhaustive des sous-systèmes surveillables

Premièrement, si nous sommes dans le cas où la décomposition canonique et la décomposition canonique modifiée coïncident, il n'y a pas de modification de la complexité de l'algorithme. Un gain de complexité ne peut survenir que lorsque, à partir du *PSO* P , on descend une branche en choisissant une équation e telle que :

$$(P \setminus \{e\})^+ \neq (P \setminus \{e\})_{mod}^+$$

Cela revient à sauter un niveau de l'arbre des *PSO*. On se placera donc dans ce cas, et on notera $P_1 = (P \setminus \{e\})^+$, $P_2 = (P \setminus \{e\})_{mod}^+$. Notons que, par construction de la décomposition canonique modifiée, P_2 est un sous-système de P_1 , c'est-à-dire que $P_2 \subset P_1$.

Supposons que le *PSO* P_i ($i = 1, 2$) ait les propriétés suivantes :

- un nombre d'équations égal à n_i ;
- un nombre de variables égal à m_i ;
- un degré de redondance égal à $d_i = n_i - m_i$;

P_2 étant un sous-système de P_1 , on a $d_2 \leq d_1 - 1$. Comme $d_2 \geq 1$, on a $d_1 \geq 2$.

L'algorithme 7 réalise successivement les deux opérations suivantes :

1. passage de P_1 à $P_2 \subset P_1$ (décomposition canonique modifiée). La complexité de cette étape est en $\mathcal{O}((d_1 - d_2).n^{2,5})$;
2. parcours de l'arbre des *PSO*, à partir de P_2 . La complexité de cette étape est en $\mathcal{O}(n^{d_2+1,5})$ ((Krysander *et al.*, 2008)).

Une étude de fonction montre que la fonction $d_2 \mapsto \frac{(d_1-d_2).n^{2,5}}{n^{d_2+1,5}}$ est croissante, donc bornée par $\frac{n^{2,5}}{n^{d_1-1+1,5}} = n^{2-d_1} \leq 1$. Ainsi, la complexité de notre algorithme est d'ordre $(n^{d_2+1,5})$. Puisque $d_2 \leq d_1 - 1$, la complexité du parcours du sous-arbre des *PSO* à partir de P_1 est en $\mathcal{O}(n^{d_1+0,5})$.

La complexité du parcours de l'arbre des *PSO* à partir de P_1 est en $\mathcal{O}(n^{d_1+1,5})$ (Krysander *et al.*, 2008).

Par conséquent, on a remplacé un parcours de sous-arbre de complexité en $\mathcal{O}(n^{d_1+1,5})$, par un parcours de complexité en $\mathcal{O}(n^{d_1+0,5})$. Le gain réalisé par notre algorithme est dans le pire des cas d'ordre n .

2.5.3 Algorithme 3 : Parcours de l'arbre des *PSO* avec évitement des boucles algébriques (causalité intégrale)

a) Principe

Cet algorithme renvoie les sous-systèmes surveillables qui sont validés par les inversibilités. L'algorithme décrit dans cette section reprend le principe de l'algorithme de la section 2.5.2. En revanche, nous faisons dans cette section l'hypothèse de causalité intégrale, ce qui nous permet d'identifier les contraintes d'un *PSO* qui vont toujours apparaître dans une boucle algébrique. L'algorithme 9, inspiré du « Ranking algorithm », identifie les contraintes algébriques qui apparaissent toujours dans une boucle algébrique.

Algorithm 9 $BA = \text{Identification_boucles_algébriques}(S)$

```
1: Identifier l'ensemble des contraintes différentielles  $c_d$  de  $S : c_d : x_d = \dot{x}_i$ 
2: Former  $S'$  à partir de  $S$  en remplaçant chaque contrainte différentielle  $c_d$  par une contrainte  $c'_d$  de type capteur sur  $x_i : c'_d(x_i)$ 
3:  $BA = \{\text{contraintes de } S'\}$ 
4:  $I = \{\text{contraintes de } S' \text{ ne faisant intervenir qu'une seule variable}\}$ 
5:  $J = \{\text{variables de } S' \text{ intervenant dans } I\}$ 
6: while  $I \neq \emptyset$  do
7:    $BA = BA \setminus I$ 
8:   Modifier  $S'$  en mettant à 0 les lignes  $I$  et les colonnes  $J$ 
9:    $I = \{\text{contraintes de } S' \text{ ne faisant intervenir qu'une seule variable}\}$ 
10:   $J = \{\text{variables de } S' \text{ intervenant dans } I\}$ 
11: end while
12:  $I = \{\text{contraintes de } S' \text{ ne faisant intervenir aucune variable}\}$ 
13:  $BA = BA \setminus I$ 
14: Retourner  $BA$ 
```

Ainsi, puisqu'il est possible d'identifier les boucles algébriques d'un PSO , on peut modifier l'algorithme 8, en indiquant que les contraintes apparaissant toujours dans une boucle algébrique sont inutilisables. En pratique, on supprimera ces contraintes, ce qui conduit à l'algorithme 10.

Algorithm 10 $M = \text{Sub_Recherche_exhaustive_2}'(P, R)$

Require: P est un PSO

```
1:  $BA = \text{Identification\_boucles\_algébriques}(M)$ 
2: Modifier  $P$  en supprimant les contraintes figurant dans  $BA$ 
3:  $P = \text{Systeme\_surveillable\_modifie\_1}(P)$ 
4:  $R = R \cap P$ 
5: if  $P$  a un degré de redondance égal à 1 then
6:    $M = M \cup P$ 
7: else
8:   while  $R \neq \emptyset$  do
9:     Choisir une équation  $e$  de  $R : e = R(1)$ 
10:    Calculer  $P_1 = (P \setminus \{e\})^+$ 
11:    Calculer  $P_2 = (P \setminus \{e\})_{mod}^+$ 
12:    Calculer  $E_1 = P \setminus P_1$ 
13:    Calculer  $E_2 = P \setminus P_2$ 
14:    Calculer  $R_1 = R \setminus E_1$ 
15:    Calculer  $R_2 = R \setminus E_2$ 
16:     $M_s = \text{Sub\_Recherche\_exhaustive\_2}(P_2, R_2)$ 
17:     $M = M \cup M_s$ 
18:     $R = R_1$ 
19:   end while
20: end if
21: Retourner  $M$ 
```

b) Gain de complexité de l'algorithme 10

Dans ce paragraphe, nous allons estimer le gain de complexité permis par l'algorithme 10, par rapport à l'algorithme *MSO* original, décrit dans (Krysander *et al.*, 2008).

Pour cet algorithme, si des contraintes apparaissent toujours dans une boucle algébrique (i.e $BA \neq \phi$), elles sont automatiquement retirées, ce qui revient à descendre l'arbre des *PSO* par une branche imposée, et à ne pas explorer les autres branches de l'arbre. Cette opération revient à sauter un ou plusieurs niveaux de l'arbre des *PSO*. On passe du *PSO* que l'on notera P_1 au *PSO* que l'on notera P_2 , en retirant les contraintes de BA puis en faisant une décomposition canonique modifiée.

Supposons que le *PSO* P_i ($i = 1, 2$) ait les propriétés suivantes :

- un nombre d'équations égal à n_i ;
- un nombre de variables égal à m_i ;
- un degré de redondance égal à $d_i = n_i - m_i$;

P_2 étant un sous-système de P_1 , on a $d_2 \leq d_1 - 1$.

La complexité du parcours de l'arbre des *PSO* à partir de P_1 est en $\mathcal{O}(n^{d_1+1,5})$.

Le « Ranking Algorithm » appliqué à P_1 a une complexité en $\mathcal{O}(n_1.m_1)$.

La complexité du parcours de l'arbre des *PSO* à partir de P_2 est en $\mathcal{O}(n^{d_2+1,5})$.

De même que pour l'algorithme 8 étudié précédemment, le saut d'un niveau se traduit par le fait qu'on a remplacé un parcours de sous-arbre de complexité en $\mathcal{O}(n^{d_1+1,5})$, par un parcours de complexité en $\mathcal{O}(n^{d_1+0,5})$. Le gain réalisé par notre algorithme est dans le pire des cas d'ordre n .

2.6 Prise en compte du cahier des charges

a) Principe

Les critères de la section 2.4 sont utiles pour caractériser les sous-systèmes discriminants, par rapport à un cahier des charges d'isolabilité donné. Ces critères expriment que le sous-ensemble dans lequel il est possible de construire des résidus structurellement insensibles à f_i est soit $(S^+ \setminus \{f_i\})^+$, soit $(S_{mod}^+ \setminus \{f_i\})_{mod}^+$, selon que l'on veuille prendre en considération ou non les non-inversibilités.

Supposons que nous ayons un cahier des charges de diagnostic, exprimé sous la forme d'un ensemble de fautes $\{f_1, \dots, f_p\}$ à discriminer. En ne tenant pas compte des non-inversibilités, les sous-systèmes discriminants sont trouvés :

- en menant une recherche exhaustive sur $(S^+ \setminus \{f_1\})^+$, et en s'assurant qu'au moins une des fautes $\{f_2, \dots, f_p\}$ apparaisse dans chaque sous-système surveillable trouvé. Ainsi, on trouve des sous-systèmes surveillables insensibles à f_1 mais sensibles à certaines autres fautes $f_2, \dots, f_p$;

- en menant une recherche exhaustive sur $(S^+ \setminus \{f_2\})^+$, et en s’assurant qu’au moins une des fautes $\{f_1, f_3, \dots, f_p\}$ apparaisse dans chaque sous-système surveillable trouvé ;
- en menant une recherche exhaustive sur $(S^+ \setminus \{f_i\})^+$, et en s’assurant qu’au moins une des fautes $\{f_1, \dots, f_{i-1}, f_{i+1}, \dots, f_p\}$ apparaisse dans chaque sous-système surveillable trouvé.

Pour la recherche sur chaque sous-système $(S^+ \setminus f_i)^+$, des algorithmes comme ceux présentés aux sections 2.5.2 et 2.5.3 permettent de guider la recherche, c’est-à-dire que chaque *PSO* peut être testé afin :

1. de ne pas trouver les sous-systèmes préalablement trouvés ;
2. d’arrêter la recherche pour éviter de trouver des sous-systèmes insensibles à toutes les fautes.

Un article s’inscrivant dans la même optique a été présenté récemment, au congrès de diagnostic DX’10 (Krysander *et al.*, 2010).

b) Gain de complexité permis par l’intégration d’un cahier des charges

Estimons le gain de complexité obtenu en tenant compte d’un cahier des charges, par rapport à l’algorithme *MSO* original, décrit dans (Krysander *et al.*, 2008).

Pour cet algorithme, mener une recherche exhaustive dans $(S^+ \setminus \{f_i\})^+$ revient à imposer la branche d’exploration de l’arbre des *PSO*.

Calculer $(S^+ \setminus \{f_i\})^+$ a une complexité en $\mathcal{O}(d.n^{2,5})$.

La complexité du parcours de l’arbre des *PSO* à partir de $(S^+ \setminus \{f_i\})^+$ est en $\mathcal{O}(n^{d+0,5})$, car $(S^+ \setminus \{f_i\})^+$ a un degré de redondance $\leq d - 1$.

Ces deux opérations étant répétées au plus p fois, la complexité de la recherche intégrant un cahier des charges est en $\mathcal{O}(p.n^{d+0,5})$. La complexité de l’algorithme original était en $\mathcal{O}(n^{d+1,5})$.

2.7 Récapitulatif

Ce chapitre, consacré aux algorithmes d’analyse structurelle développés, s’appuie sur plusieurs algorithmes actuels, dont nous avons rappelé le fonctionnement. La figure 2.22, déjà présentée, récapitule ce chapitre, et illustre qu’il est possible de substituer aux algorithmes actuels d’analyse structurelle des variantes, parfois plus restrictives – quand elles s’appuient sur des critères de calculabilité par simulation – mais dont les performances, en terme de temps de calcul, sont accrues.

L’algorithme central de ce chapitre est la décomposition de Dulmage-Mendelsohn modifiée. Cet algorithme permet de trouver les parties du graphe qui peuvent être atteintes en suivant des chemins orientés, avec la contrainte que ces chemins n’utilisent pas d’arc non-inversible. La définition des arcs inversibles et non-inversibles a donc son importance dans le résultat de la décomposition, et c’est ce qui motive la phase de définition des non-inversibilités (dans laquelle s’inscrit notamment l’algorithme de la section 2.2). Nous avons ainsi été amenés à définir plusieurs niveaux d’analyse, qui déterminent la façon dont les non-inversibilités sont définies. Le

premier niveau d'analyse est de ne pas définir d'arcs non-inversibles ; il correspond à l'analyse structurelle classiquement réalisée. Le deuxième niveau est de ne définir des arcs inversibles que sur les contraintes différentielles dont on est certain qu'elles ne peuvent être utilisées qu'en tant qu'intégrateurs. Le troisième niveau est d'imposer une causalité intégrale.

Par ailleurs, de nombreux algorithmes actuels utilisant la décomposition canonique, il est possible de les adapter en utilisant la décomposition canonique modifiée. Ainsi, les critères d'isolabilité ont pu être adaptés pour prendre en compte les non-inversibilités définies préalablement ; ainsi, les algorithmes de recherche exhaustive ont été modifiés.

Au chapitre suivant, ces algorithmes seront comparés sur un exemple de taille et de complexité moyenne. Cette comparaison illustrera le gain de performance qu'il est possible de réaliser. Cependant, c'est sur les systèmes les plus complexes que la réduction du temps de calcul est la plus patente. La deuxième partie de ce rapport, présentant l'analyse structurelle de la boucle d'air, en fera la démonstration, puisque nos algorithmes ont permis de mener une analyse structurelle en un temps fini, ce qui n'était pas possible en utilisant les algorithmes existant.

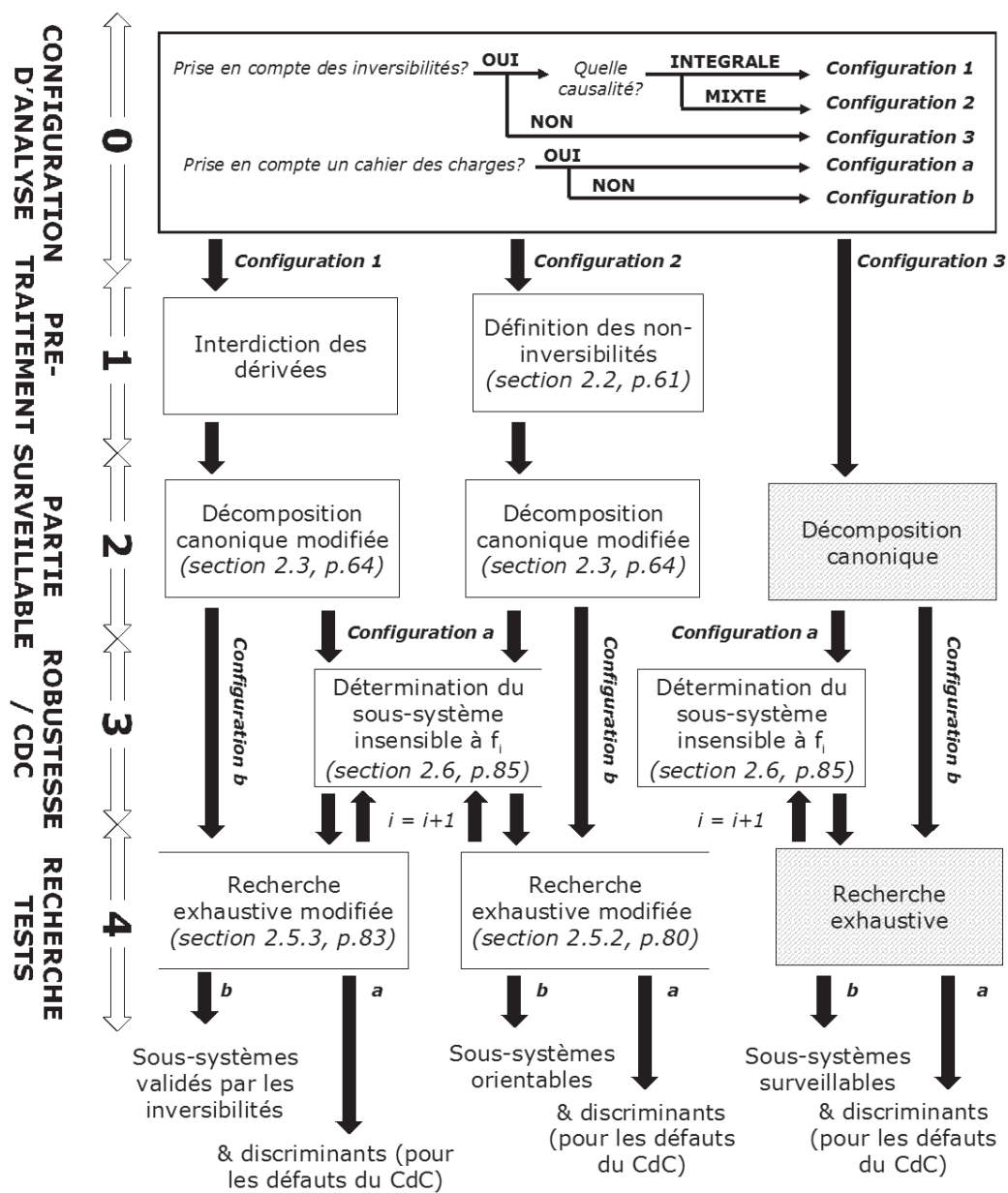


FIGURE 2.22 – Récapitulatif des algorithmes développés

Chapitre 3

Présentation de la plate-forme logicielle développée et application sur un exemple

Résumé

Ce chapitre montre l'application, sur un exemple, des algorithmes présentés au deuxième chapitre. L'application servant à illustrer le fonctionnement des algorithmes est un benchmark du réseau européen *DAMADICS* (*Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems*).

Nous présentons ainsi des résultats sur le temps de calcul des algorithmes, et nous vérifions que les algorithmes du chapitre 2 permettent effectivement de réduire ce temps de calcul, par rapport aux algorithmes existant.

Nous verrons ensuite que différentes hypothèses de calculabilité des résidus (causalité mixte ou causalité intégrale) conduisent à différentes tables d'isolabilité. Nous en tirerons donc la conclusion que des hypothèses de calculabilité restrictives ne permettent parfois pas d'atteindre l'isolabilité désirée, et qu'il faut lever ces hypothèses (par exemple : admettre la causalité mixte, ou admettre que les boucles algébriques sont solvables) si l'on veut discriminer les fautes.

Nous débutons ce chapitre par la présentation de l'outil logiciel développé, de ses fonctionnalités et de son interface graphique.

Abstract

This chapter exemplifies the algorithms described in the second chapter. The application case which is chosen to illustrate the use of the algorithms is a benchmark of the European Research Training Network *DAMADICS* (Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems).

Results on the computation time of the algorithms are thus presented, which will show that the computation time can indeed be reduced, comparatively to existing algorithms, thanks to the algorithms presented in chapter 2.

It will be shown that different hypotheses on residuals' calculability (mixed causality or integral causality) lead to different isolability tables. The conclusion which will be drawn is that the calculability hypotheses may sometimes be too restrictive for the isolability requirements to be fulfilled. Therefore, if one wants to discriminate faults, one has to untight the calculability hypotheses, by, for instance, allowing mixed causality or supposing that algebraic loops are solvable.

This chapter starts with a presentation of the functionalities and interface of the software which was developed during the thesis.

3.1. Présentation des fonctionnalités de l'outil logiciel

L'outil logiciel développé au cours de cette thèse permet d'analyser les modèles codés sous la forme d'un schéma-bloc *MATLAB/Simulink*, aussi l'une des fonctionnalités de l'outil consiste à extraire un modèle structurel d'un modèle Simulink – l'intérêt résidant en la possibilité d'analyser rapidement les modèles développés et partagés au sein de l'entreprise. Il n'est bien sûr pas indispensable de disposer d'un modèle Simulink : les algorithmes présentés au chapitre 2, incorporés au logiciel, permettent de faire l'analyse de tout système dont on connaît la structure sous forme d'une matrice d'incidence. Les fonctionnalités de l'outil logiciel sont présentées en premier lieu (section 3.1), avant l'application des algorithmes au modèle du benchmark *DAMA-DICS* (section 3.2).

3.1 Présentation des fonctionnalités de l'outil logiciel

L'outil logiciel développé (figure 3.1) au cours de cette thèse est construit autour de deux fonctions principales : la première consiste à extraire un modèle structurel d'un modèle Simulink (section 3.1.1), la seconde consiste à réaliser l'analyse de tout système dont on connaît la structure sous forme d'une matrice d'incidence (section 3.1.2).

Une interface graphique a été développée afin de simplifier l'utilisation des algorithmes (voir figures 3.1, 3.2, et 3.4).

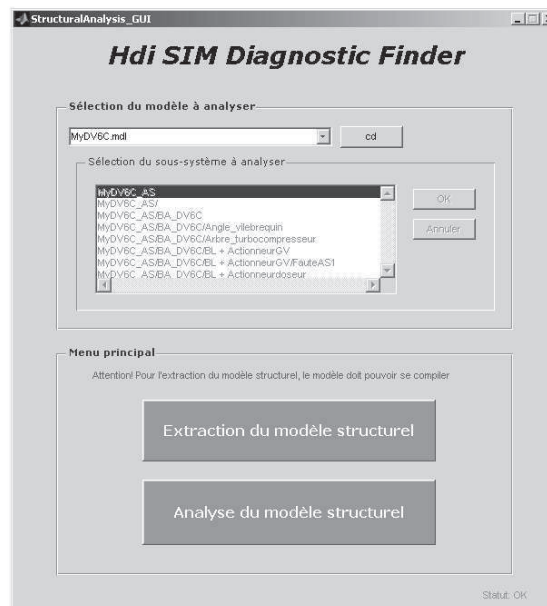


FIGURE 3.1 – Interface de l'outil logiciel développé

3.1.1 Ecriture du modèle structurel

a) Conditionnement du modèle Simulink

Il convient de faire, avant d'extraire une matrice d'incidence à partir du modèle *Simulink*, une analyse critique de ce modèle, et, éventuellement, d'en changer l'écriture afin d'éviter certains



FIGURE 3.2 – Interface pour l'extraction du modèle structurel

problèmes lors de l'analyse structurelle. C'est ce que nous appelons le conditionnement du modèle *Simulink*. Ainsi, on portera une attention particulière aux points suivants :

- ne pas écrire la même fonction deux fois dans le modèle, car cela peut éventuellement masquer des liens ;
- éviter les blocs génériques « fonction », « s-function », car ils masquent les opérations qui s'effectuent en interne de ces blocs. On préférera utiliser des fonctions élémentaires, dont on connaît l'inversibilité.

b) Structure

Afin d'écrire un modèle structurel à partir d'un modèle *Simulink*, on procède en plusieurs étapes :

1. mise en forme du modèle : l'utilisateur renseigne sur le modèle les variables mesurées et les variables de faute. A cet effet, des blocs spéciaux sont utilisés (voir figure 3.3) ;
2. automatiquement, l'algorithme réalise tour à tour les actions suivantes :
 - identification des blocs engendrant des contraintes : on identifie particulièrement les contraintes dynamiques, les variables en causalité intégrale et les variables en causalité dérivée ;
 - écriture d'une matrice d'incidence type pour chaque bloc ;
 - suivi des variables à travers les blocs qui ne génèrent pas de contraintes, tels les « mux », « demux », « selector » ;
 - identification des variables d'entrées (constantes, paramètres, commandes) ;
 - identification des variables mesurées et de fautes.

c) Inversibilités

Dans ce qui précède, lorsque l'on écrit une matrice d'incidence « type » pour chaque bloc, au deuxième tiret de l'étape 2, des choix par défaut sont faits sur certaines fonctions, par exemple :

- les *switchs* sont considérés comme des fonctions non-inversibles ;

3.1. Présentation des fonctionnalités de l'outil logiciel

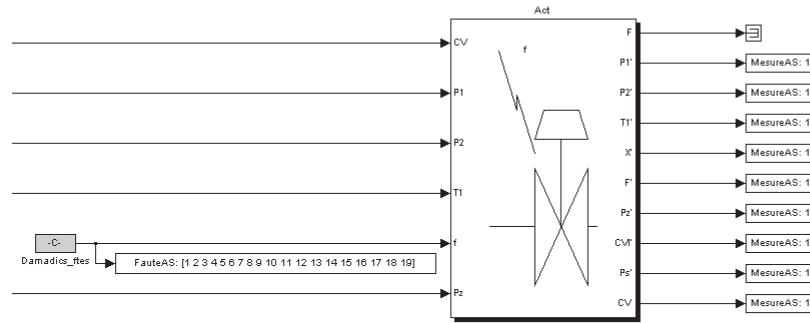


FIGURE 3.3 – Renseignement des variables mesurées et des variables de faute du modèle

- les saturations sont considérées comme des fonctions non-inversibles ;
- les tables sont considérées comme des fonctions non-inversibles.

Un fichier de configuration est créé lors de l'extraction du modèle pour indiquer les choix par défaut. Ce fichier peut-être modifié manuellement, c'est-à-dire qu'il est possible pour l'utilisateur d'indiquer :

- pour un *switch*, la branche du *switch* qui est utilisée (ce qui le rend inversible) ;
- pour une saturation, si la saturation est inversible, c'est-à-dire si l'on se trouve dans la partie linéaire de la saturation ;
- pour une table, si la table est inversible.

Le modèle structurel sera par la suite modifié afin que ces choix soient pris en compte.

La seconde possibilité permise par le logiciel est, si les paramètres du modèle sont identifiés, de récupérer des informations de la simulation, en particulier la position des *switchs* et l'inversibilité ou non des saturations, puis d'écrire ces informations dans le fichier de configuration, informations qui seront de même impactées sur le modèle structurel.

3.1.2 Analyse du modèle structurel

En premier lieu, il est demandé à l'utilisateur de choisir une configuration (notée 1, 2, 3, a, b, sur la figure 2.22). Cette configuration reflète les paramètres d'analyse structurelle voulus eu égard à la causalité et à l'utilisation d'un cahier des charges. Un menu principal permet ensuite de choisir entre plusieurs options (voir figure 3.4).

1. Analyse de l'isolabilité des fautes a priori ;
2. Recherche des sous-systèmes surveillables ;
3. Validation des sous-systèmes surveillables ;
4. Analyse a posteriori de l'isolabilité des fautes ;
5. Aide au choix d'un sous-système surveillable ;
6. Aide à la construction d'un test fonctionnel ;
7. Recherche des conditions initiales estimables.

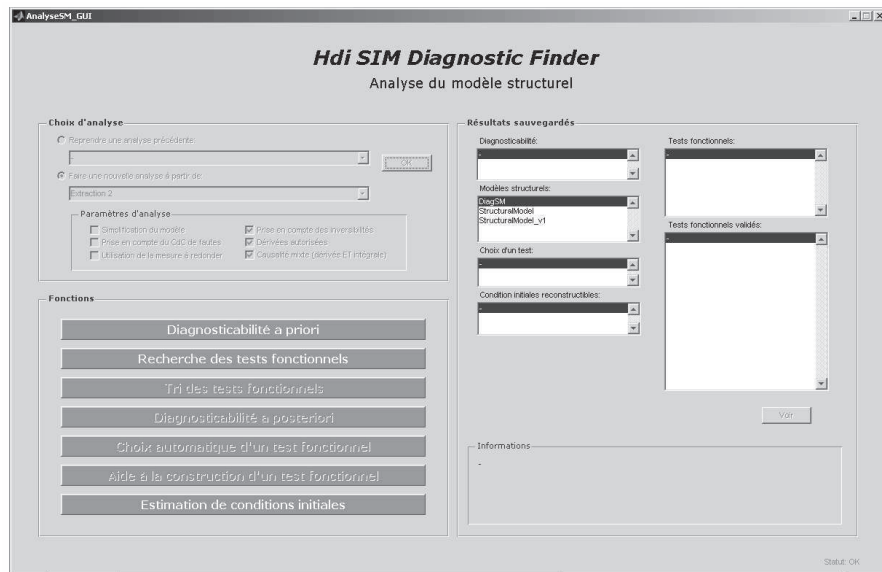


FIGURE 3.4 – Interface pour l’analyse du modèle structurel

a) Analyse de l’isolabilité des fautes a priori

Voir le récapitulatif des algorithmes utilisés à la section 2.1.3.

b) Recherche des sous-systèmes surveillables

Voir le récapitulatif des algorithmes utilisés à la section 2.1.3. On mentionnera de plus qu’une possibilité du logiciel, non mentionnée dans la section 2.1.3, consiste à imposer que le test de diagnostic se termine par un capteur que l’on aura préalablement indiqué. Pratiquement, il suffit de considérer que le capteur que l’on aura choisi comme redondant est non-inversible, et d’imposer que les tests trouvés l’utilisent.

c) Validation des sous-systèmes surveillables

Nous gardons les sous-systèmes surveillables validés par les inversibilités, quelle que soit l’hypothèse (causalité intégrale, mixte, ou dérivée).

d) Analyse a posteriori de l’isolabilité des fautes

Il s’agit d’une analyse classique de l’isolabilité permise par les sous-systèmes surveillables trouvés et validés.

e) Aide au choix d’un sous-système surveillable

Cette fonction permet, parmi l’ensemble des sous-systèmes surveillables validés, d’en choisir un à tester. Les critères retenus pour la sélection sont :

- la taille des sous-systèmes surveillables : les sous-systèmes les plus petits sont intéressants, car on peut penser a priori qu'étant sensibles à moins de paramètres du modèle, ils sont potentiellement plus robustes aux erreurs de modélisation ;
- l'absence d'intégrateurs purs (intégrateurs ne figurant pas dans une boucle) : les chemins contenant des intégrateurs purs sont fortement pénalisés ;
- le non-mélange des causalités : on préférera les sous-systèmes en causalité toute différentielle ou toute intégrale ;
- la similarité avec le modèle initial : les chemins contenant peu d'inversions sont favorisés.

f) Aide à la construction d'un test fonctionnel

Il s'agit de fournir à l'utilisateur une visualisation du chemin induit par le couplage, et de lui indiquer les modifications à apporter au modèle initial pour reconstruire un test de diagnostic (contraintes à inverser, contraintes conservées telles quelles, et contraintes à supprimer).

g) Recherche des conditions initiales estimables

Le graphe structurel permet de calculer certaines conditions initiales, que l'on pourra réinjecter dans les schémas de simulation des résidus. La causalité utilisée pour le calcul des conditions initiales est une causalité dérivée, et l'algorithme utilisé est similaire au « Ranking Algorithm » (Blanke *et al.*, 2006).

3.2 Application au benchmark DAMADICS

Le système faisant l'objet du benchmark est un actionneur de vanne, permettant de modifier l'ouverture de celle-ci afin de commander le débit passant. La modification de l'ouverture de la vanne est accomplie par un servomoteur pneumatique à membrane : un fluide sous pression agit sur le diaphragme pour fournir le mouvement linéaire de la tige du servomoteur. Un dispositif « positionneur » est utilisé dans le but d'éliminer les erreurs de position de la vanne : il s'agit d'une boucle interne de contrôle de la position de la tige, réalisée par un correcteur proportionnel.

La figure 3.5 représente le système : le contour de l'étude est figuré par des pointillés, et les trois ensembles de l'actionneur (vanne, servomoteur, et positionneur) sont distingués. En dehors du contour pointillé, $V1$, $V2$, $V3$ sont des vannes manuelles, et les sigles TT , PT , et FT représentent des capteurs. A l'intérieur du contour en pointillé :

- la position de la tige est repérée par X ,
- ZT représente le capteur de position de la tige,
- ZC représente le contrôleur de position (boucle interne),
- CV est la consigne calculée par la boucle de régulation externe (régulation du débit),
- CVI est la signal électrique de sortie du contrôleur,
- E/P représente le convertisseur électro-pneumatique, dont la pression d'alimentation Pz est fournie par l'organe PSP
- Ps est la pression dans la chambre du servomoteur.

19 fautes de tous types sont considérées pour ce benchmark (défauts de mesure, fuites, perforation du diaphragme pour en citer quelques unes). Pour plus d'informations, le lecteur est

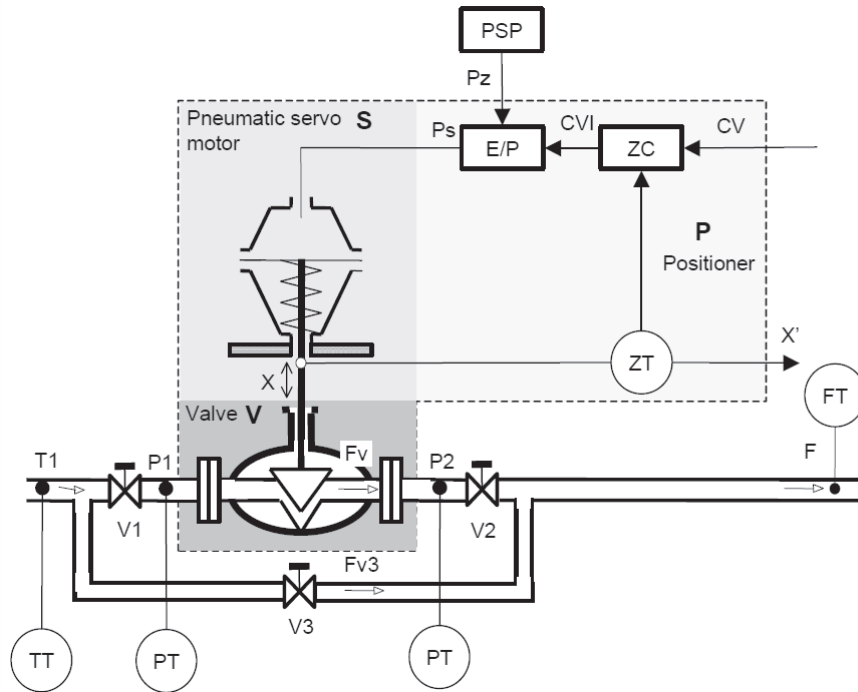


FIGURE 3.5 – Schéma de l'actionneur

invité à se référer à la page internet du benchmark (WUT, 2002).

Le benchmark *DAMADICS* dans la littérature : présentation et comparaisons

Nous présentons dans cette section un court état de l'art sur la détection et l'isolation des fautes du benchmark *DAMADICS*. Faire une comparaison est difficile ; elle doit être menée avec prudence du fait des nombreuses différences entre les travaux présentés : ces différences portent sur les modèles (modèles analytiques, qualitatifs, réseau de neurones, ou modèles structurels), sur les approches mises en œuvre (approches sans modèles, observateurs, automates ...) et sur les mesures considérées : nous avons, dans notre cas d'application, utilisé des capteurs supplémentaires pour justifier pleinement l'utilisation de l'analyse structurelle (celle-ci s'appliquant sur des systèmes comportant de nombreux capteurs) et illustrer les apports en terme de temps de calcul de nos algorithmes. Au final, on se rendra compte qu'il y a peu de points communs entre ces méthodes, mais nous essaierons d'en faire ressortir quelques un.

Dans (Uppal *et al.*, 2006), deux méthodes sont comparées : la première met en œuvre un observateur étendu à entrée inconnue, l'autre s'appuie sur un modèle neuro-flou. Les fautes considérées sont les fautes f_1 , f_2 , f_7 , f_{13} , f_{18} , f_{19} . La possibilité de détecter les fautes dépend de leur force, de leur gravité. Pour la méthode de détection par observateur, seuls la position de la tige et le débit sont mesurés. Par ailleurs, toutes les fautes considérées sont détectables pour des défauts importants. Si les fautes sont moyennes ou petites, seules les fautes f_1 , f_7 , f_{13} , f_{18} , f_{19} sont détectables.

Concernant l'approche reposant sur un modèle neuro-flou, toutes les fautes considérées sont détectables et isolables, quelle que soit l'échelle de la faute.

Dans (Puig *et al.*, 2006), sont construits des observateurs à intervalles pour la détection des fautes $f_1, f_{10}, f_{11}, f_{13}$. Il s'agit de détection uniquement, et la réussite de celle-ci dépend de la taille de la faute.

L'article (Witczak *et al.*, 2006) décrit une approche fondée sur un réseau de neurones dit « GMDH ». En ce qui concerne la détectabilité, f_5, f_9, f_{14} sont non détectables. La détectabilité comme l'isolabilité dépendent de l'importance du défaut, et de sa vitesse d'apparition. f_4 et f_{15} sont toujours non isolables. Il en est de même pour $f_2, f_3, f_6, f_{18}, f_{19}$. Ces deux ensembles, en revanche sont isolables entre eux.

Dans (Supavatanakul *et al.*, 2006) sont employées des méthodes discrètes de diagnostic, utilisant des automates temporisés : Les défauts f_{16}, f_{18}, f_{19} sont détectés.

Dans (Previdi and Parisini, 2006), la détection uniquement est considérée, par des méthodes d'estimation spectrale : $f_1, f_2, f_7, f_8, f_{10}, f_{11}, f_{12}, f_{14}, f_{15}, f_{18}, f_{19}$ sont détectés, et f_{17} également, mais seulement si les pressions de part et d'autres de la vanne sont mesurées.

Les auteurs de (Bocaniala and da Costa, 2006) ne considèrent que les fautes $f_1, f_2, f_7, f_{10}, f_{11}, f_{13}, f_{15}, f_{16}, f_{17}, f_{18}, f_{19}$. Les résultats du classifieur dépendent de l'importance du défaut. Des groupes de fautes sont néanmoins identifiés : il s'agit des groupes $\{f_1, f_7\}$, $\{f_2, f_{19}\}$, $\{f_7, f_{10}\}$, $\{f_{11}, f_{15}, f_{16}\}$, $\{f_{13}, f_{18}\}$ et $\{f_{17}\}$.

(Calado *et al.*, 2006) emploie un raisonnement qualitatif couplé à des réseaux de neurones flous. La méthode présente des difficultés pour isoler les défauts naissants. En mettant de côté ce cas, les fautes $f_2, f_7, f_{10}, f_{11}, f_{13}, f_{17}, f_{18}, f_{19}$ sont isolables. Pour le reste, le module d'isolation ne donne pas de résultat.

Au final, sur les sept articles présentés, toute faute est détectable par au moins une approche, mais il n'y a pas de point commun évident entre les approches.

3.2.1 Modèle structurel du benchmark DAMADICS

Le modèle *Simulink* de l'actionneur est disponible sur la page internet du benchmark (WUT, 2002). Sur le système réel, les grandeurs Pz , CVI et Ps ne sont pas mesurées mais sont indiquées susceptibles de l'être. Nous faisons, comme dans (Düstegör *et al.*, 2004d), l'hypothèse qu'elles le sont.

L'analyse du modèle *Simulink* de l'actionneur nous fournit un graphe structurel avec 311 contraintes et 307 variables inconnues. La partie surveillable du modèle contient 283 contraintes et 279 variables inconnues. Le degré de redondance du système est donc égal à 4.

Le modèle contient 4 blocs « *switch* », 8 tables, et 102 saturations. Il contient par ailleurs 4 intégrateurs, 2 fonctions de transfert du premier ordre (ce qui donne 2 autres contraintes différentielles). Seul un de ces intégrateurs peut être pris en causalité différentielle.

3.2.2 Sous-systèmes surveillables du benchmark *DAMADICS*

Au total, le nombre de sous-systèmes surveillables (sans prendre en compte les inversibilités, sans chercher les sous-systèmes discriminants) est égal à 265. Sur ceux-ci, 189 sont structurellement discriminants pour les fautes $f_1, \dots, f_{19}$ (au sens de la définition 1.2.7). Par ailleurs, notons qu'il n'existe aucun sous-système sans boucle : tous les sous-systèmes surveillables trouvés contiennent au moins une boucle différentielle.

Le tableau 3.1 récapitule le nombre de sous-systèmes surveillables trouvés et validés, pour différentes hypothèses de causalité.

	causalité mixte	causalité intégrale
Sous-systèmes orientables	26	-
(dont discriminants)	24	-
Sous-systèmes validés	16	16
(dont discriminants)	16	16

TABLE 3.1 – Nombre de sous-systèmes surveillables du modèle

On remarque sur le tableau 3.1 que le nombre de sous-systèmes surveillables validés est identique pour la causalité mixte ou pour la causalité intégrale. Cela tient aux nombreuses non-inversibilités (saturations, hysteresis...) : en considérant toutes ces saturations non-inversibles, on contraint très fortement le modèle dans son orientation entrée-sortie (intégrale) : la causalité mixte est, dans ce cas, presque une causalité intégrale.

En revanche, on peut changer les paramètres par défaut d'Analyse Structurale, de sorte à écrire un modèle pour lequel on relâche les contraintes de non-inversibilité sur certaines saturations, et pour lequel on identifie la position des blocs « *switch* ». Nous qualifierons ce modèle de « nominal », pour la raison que les paramètres de commutation ou de non-inversibilité que nous avons choisis sont caractéristiques d'un fonctionnement normal, ou « nominal ». Nous avons par exemple supposé qu'au nominal, les capteurs ne saturent pas, ou encore que les positions de certains *switches* sont connues. Le modèle *Simulink* peut du reste être utilisé pour gagner de la connaissance sur les commutations et sur les saturations du systèmes : une des vocations d'un modèle de simulation est en effet d'instruire son utilisateur sur le fonctionnement du système. Le tableau 3.2 présente les résultats pour ce modèle dit « nominal ».

	causalité mixte	causalité intégrale
Sous-systèmes orientables	32	-
(dont discriminants)	30	-
Sous-systèmes validés	22	16
(dont discriminants)	22	16

TABLE 3.2 – Nombre de sous-systèmes du modèle nominal

Chacun des sous-systèmes validés contient une boucle différentielle. Les résidus peuvent être obtenus en simulation pourvu que les conditions initiales soient connues.

Comparaison des temps de calcul des algorithmes développés, entre eux et par rapport aux algorithmes de la littérature

Nous donnerons dans cette section les temps de calcul des algorithmes suivants :

1. Pour une hypothèse où l'on ne prend pas du tout en compte les inversibilités :
 - l'algorithme *MSO* de (Krysander *et al.*, 2008), qui permet de trouver 265 sous-systèmes surveillables ;
 - l'algorithme *MSO* de (Krysander *et al.*, 2008), modifié par nos soins, pour ne trouver que les 189 sous-systèmes discriminants.
 Cette comparaison permettra de montrer le gain réalisé en se limitant aux fautes du cahier des charges.
2. Pour une hypothèse de causalité mixte sur le modèle nominal :
 - l'algorithme de recherche des sous-systèmes surveillables sur les macro-graphes de la section 2.5.1 ;
 - l'algorithme de recherche des sous-systèmes surveillables de la section 2.5.2.
 Ces algorithmes permettent de trouver les 32 sous-systèmes orientables. Nous verrons le gain réalisé par rapport à une approche exhaustive.
3. Pour une hypothèse de causalité intégrale sur le modèle nominal, l'algorithme de la section 2.5.3 donnant les 16 sous-systèmes validés par les inversibilités. Nous verrons également le gain réalisé par rapport à une approche exhaustive, et par rapport à l'algorithme de la section 2.5.2.

L'ordinateur utilisé pour ces comparaisons est une station de travail HP avec Windows XP Pro X64, version 2003. Le processeur est un Intel Core 2 Duo E8400 à 3GHz, et la version de MATLAB utilisée est la version 7.7.0.471.

Les codes des algorithmes ne sont pas optimisés, mais sont semblables. Par conséquent, les temps de calcul sont comparables.

La table 3.3 récapitule les résultats.

Algorithme	Hypothèse(s)	Sous-systèmes	Nbre	Tps calcul
MSO	pas d'inversibilités	tous	265	3.36s
MSO discriminants	pas d'inversibilités	discriminants	189	0.75s
section 2.5.1	causalité mixte, modèle simplifié	orientables	32	1.42s
section 2.5.2	causalité mixte, modèle simplifié	orientables	32	0.52s
section 2.5.3	causalité intégrale, modèle simplifié	validés	16	0.44s

TABLE 3.3 – Temps de calcul des algorithmes développés

Il faut de surcroît noter que les sous-systèmes surveillables trouvés par l'algorithme de la section 2.5.3 sont validés, par conséquent il n'est pas utile de lancer un algorithme de validation des sous-systèmes à la suite. De manière générale, moins il y a de sous-systèmes trouvés en premier lieu, moins la phase de validation sera coûteuse en temps de calcul. Le gain de temps de calcul des algorithmes développés par rapport à l'algorithme « MSO » s'en trouve accru, et l'économie de temps de calcul se ressent également à ce niveau.

3.2.3 Isolabilité des fautes du benchmark *DAMADICS*

La décomposition de Dulmage-Mendelsohn nous permet d'analyser la détectabilité des fautes du système : toutes les fautes sont structurellement détectables, à l'exception de trois d'entre elles : il s'agit des fautes f_7, f_{16}, f_{17} . Ces défauts représentent respectivement :

- l'évaporation du fluide traversant la vanne, modélisée par une augmentation de T_1 ;
- la chute de la pression d'alimentation P_z fournie, modélisée par une chute de P_z ;
- un changement de pression inattendu de part et d'autre de la vanne, modélisé par un changement de la différence des pressions de part et d'autre de la vanne $P_1 - P_2$.

Remarquons que les résultats de l’analyse structurelle concordent avec ceux établis dans les deux articles présentant une approche à base d’observateurs, en ce sens qu’il n’y a pas de défaut non structurellement détectable qui soit réellement détectable. Ils concordent également avec les résultats présentés dans (Düstegör *et al.*, 2006) ou (Düstegör *et al.*, 2004d), malgré quelques différences qui sont dues aux différences de modélisation.

Sans prendre en compte la notion d'inversibilité, l'isolabilité des fautes du benchmark est présentée à la figure 3.6.

f_1	DAMADICS_AS/Damadics_ftes/Output#1/1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Output#1/3	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Output#1/2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Output#1/8	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Output#1/4	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Output#1/19	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Output#1/18	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Output#1/6	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Output#1/5	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Output#1/12	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Output#1/9	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
f_{10}	DAMADICS_AS/Damadics_ftes/Output#1/10	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
f_{11}	DAMADICS_AS/Damadics_ftes/Output#1/11	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	DAMADICS_AS/Damadics_ftes/Output#1/13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	DAMADICS_AS/Damadics_ftes/Output#1/14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	DAMADICS_AS/Damadics_ftes/Output#1/15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

FIGURE 3.6 – Table d’isolabilité des fautes du benchmark

Si l'on introduit les inversibilités pour faire le tri des sous-systèmes, l'isolabilité des fautes est naturellement moins forte. Nous avons considéré deux cas : celui de paramètres d'analyse dits « nominaux » et celui de paramètres d'analyse par défaut. Pour chacun de ces deux cas, les hypothèses de causalité mixte et de causalité intégrale sont examinées, et des tables d'isolabilité sont obtenues et présentées aux figures 3.7, 3.8, 3.9, et 3.10.

Remarquons que si les tables 3.7, 3.8, et 3.9 sont identiques, c'est parce que, comme nous l'avons dit précédemment, la causalité mixte est presque une causalité intégrale. Enfin, pour conclure cette section, faisons quelques commentaires sur ce qu'indiquent ces différentes tables.

3.2. Application au benchmark DAMADICS

		Table d'isolabilité															
Fautes	f_{11} :	DAMADICS_AS/Damadics_ftes/Outport#1/11	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	f_{10} :	DAMADICS_AS/Damadics_ftes/Outport#1/10	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
		DAMADICS_AS/Damadics_ftes/Outport#1/8	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
		DAMADICS_AS/Damadics_ftes/Outport#1/4	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	f_1 :	DAMADICS_AS/Damadics_ftes/Outport#1/1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
		DAMADICS_AS/Damadics_ftes/Outport#1/3	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
		DAMADICS_AS/Damadics_ftes/Outport#1/2	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
		DAMADICS_AS/Damadics_ftes/Outport#1/19	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
		DAMADICS_AS/Damadics_ftes/Outport#1/18	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
		DAMADICS_AS/Damadics_ftes/Outport#1/6	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
		DAMADICS_AS/Damadics_ftes/Outport#1/5	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
		DAMADICS_AS/Damadics_ftes/Outport#1/12	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
		DAMADICS_AS/Damadics_ftes/Outport#1/9	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
		DAMADICS_AS/Damadics_ftes/Outport#1/13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
		DAMADICS_AS/Damadics_ftes/Outport#1/14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	DAMADICS_AS/Damadics_ftes/Outport#1/15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

FIGURE 3.7 – Table d'isolabilité des fautes du benchmark – Paramètres d'analyse par défaut, causalité intégrale

		Table d'isolabilité															
Fautes	f_{11} :	DAMADICS_AS/Damadics_ftes/Output#1/11	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	f_{10} :	DAMADICS_AS/Damadics_ftes/Output#1/10	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
		DAMADICS_AS/Damadics_ftes/Output#1/8	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
		DAMADICS_AS/Damadics_ftes/Output#1/4	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	f_1 :	DAMADICS_AS/Damadics_ftes/Output#1/1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
		DAMADICS_AS/Damadics_ftes/Output#1/3	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
		DAMADICS_AS/Damadics_ftes/Output#1/2	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
		DAMADICS_AS/Damadics_ftes/Output#1/19	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
		DAMADICS_AS/Damadics_ftes/Output#1/18	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
		DAMADICS_AS/Damadics_ftes/Output#1/6	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
		DAMADICS_AS/Damadics_ftes/Output#1/5	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
		DAMADICS_AS/Damadics_ftes/Output#1/12	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
		DAMADICS_AS/Damadics_ftes/Output#1/9	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
		DAMADICS_AS/Damadics_ftes/Output#1/13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
		DAMADICS_AS/Damadics_ftes/Output#1/14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	DAMADICS_AS/Damadics_ftes/Output#1/15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

FIGURE 3.8 – Table d'isolabilité des fautes du benchmark – Paramètres d'analyse par défaut, causalité mixte

A propos de f_{11} (défaut du ressort du servo-moteur)

D'après la table d'isolabilité structurale présentée à la figure 3.6, f_{11} est isolable par rapport à tous les autres défauts. Ce n'est pas le cas si l'on regarde les tables d'isolabilité prenant en compte les non-inversibilités : ainsi f_{11} n'est pas isolable pour les hypothèses suivantes :

		Table d'isolabilité															
f_{11}	DAMADICS_AS/Damadics_ftes/Outport#1/11	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
f_{10}	DAMADICS_AS/Damadics_ftes/Outport#1/10	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/8	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/4	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
f_1	DAMADICS_AS/Damadics_ftes/Outport#1/1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/3	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/2	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
Fautes	DAMADICS_AS/Damadics_ftes/Outport#1/19	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/18	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/6	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/5	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/12	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/9	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	DAMADICS_AS/Damadics_ftes/Outport#1/15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

FIGURE 3.9 – Table d'isolabilité des fautes du benchmark – Paramètres d'analyse nominaux, causalité intégrale

		Table d'isolabilité															
f_{10}	DAMADICS_AS/Damadics_ftes/Outport#1/10	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/8	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/4	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
f_1	DAMADICS_AS/Damadics_ftes/Outport#1/1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/3	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/2	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
Fautes	DAMADICS_AS/Damadics_ftes/Outport#1/19	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/18	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/6	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/5	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/12	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/9	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
	f_{11}	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	DAMADICS_AS/Damadics_ftes/Outport#1/14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	DAMADICS_AS/Damadics_ftes/Outport#1/15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

FIGURE 3.10 – Table d'isolabilité des fautes du benchmark – Paramètres d'analyse nominaux, causalité mixte

- paramètres d'analyse par défaut, causalité intégrale (figure 3.7) ;
- paramètres d'analyse par défaut, causalité mixte (figure 3.8) ;
- paramètres d'analyse nominaux, causalité intégrale (figure 3.9).

Néanmoins, au vu de la figure 3.10), on peut dire que si l'on veut isoler f_{11} , il faut à la fois se placer dans le mode de fonctionnement nominal et envisager la causalité mixte.

A propos de f_1 (encrassement de la vanne) et f_{10} (perforation du diaphragme du servo-moteur)

La table d'isolabilité de la figure 3.6 montre que ces fautes sont structurellement discriminables. Les autres tables montrent que les sous-systèmes surveillables validés ne permettent a priori pas de les discriminer, quelles que soient la causalité prise ou les simplifications faites. Il peut être possible de discriminer f_1 et f_{10} mais, à première vue, ce n'est pas évident : discriminer structurellement ces deux fautes nécessite de construire des résidus sur les sous-systèmes qui n'ont pas été validés. Par conséquent, la solvabilité des boucles algébriques doit être étudiée, les algorithmes d'élimination doivent être considérés. Ainsi, au niveau de l'analyse structurelle, on peut évaluer les moyens qu'il faudra mettre en œuvre pour générer les résidus.

Chapitre 4

Conclusion et perspectives

Dans cette première partie, nous nous sommes consacrés aux développements d’algorithmes entrepris lors de cette thèse. Au premier chapitre, le cadre de ce travail a été posé, ainsi que les notions fondamentales du diagnostic à base de modèles et de l’analyse structurelle. Une étude bibliographique a permis de situer les problématiques industrielles que nous avons rencontrées par rapport à l’existant. Ces problématiques concernent la génération des tests de diagnostic, mais principalement la mise en œuvre des algorithmes d’Analyse Structurelle sur des systèmes complexes.

En ce qui concerne la génération des tests de diagnostic, nous avons fait le choix, compte-tenu de la nature de notre modèle, d’utiliser l’analyse structurelle pour construire des schémas de simulation des résidus. Le deuxième chapitre traite de la difficulté de mener une analyse structurelle sur un système complexe. Pour ce faire, nous avons travaillé sur les algorithmes existant afin de les améliorer. Les améliorations que nous apportons portent principalement sur l’intégration des inversibilités dans les algorithmes actuels, puis, dans un second temps, sur l’intégration d’un cahier des charges de diagnostic. Le benchmark *DAMADICS* traité au troisième chapitre, illustre notre travail.

Dans ce troisième chapitre, nous avons présenté le gain de temps de calcul que peut apporter l’intégration de critères de calculabilité dans les algorithmes actuels. Ce cas d’application nous a également permis d’illustrer que l’isolabilité structurelle pouvait être différente suivant les hypothèses faites, concernant la causalité, mais également concernant l’inversibilité de fonctions comme les saturations.

Plusieurs questions ressortent donc de ce travail et de ce cas d’application, que nous allons développer dans les sections suivantes. Ce sont des perspectives de ce travail.

4.1 Simulation des résidus et robustesse

Une première perspective de ce travail serait de travailler à la robustesse des estimateurs construits grâce aux algorithmes du chapitre 2. Ces estimateurs ne sont robustes ni vis-à-vis des bruits ni vis-à-vis des conditions initiales. Dans cette section, nous revenons donc en premier lieu sur la génération des résidus en simulation et apportons quelques éléments concernant la possibilité de les rendre robustes aux bruits ou aux conditions initiales.

Les règles de calculabilité des schémas de simulation comportant des boucles différentielles, énoncées à la section 1.3.3 conduisent à l'écriture d'équations différentielles semi-explicites et à leur résolution par une méthode « ODE ».

Cependant, le cadre de ces travaux est plus large que celui de la résolution numérique : en effet, résoudre une équation numérique par la méthode « ODE » conduit à écrire des systèmes sous forme semi-explicite, transformables en équation différentielle ordinaire. La forme obtenue est donc familière et utilisable, car elle s'apparente à une forme d'état.

$$\begin{cases} \dot{x} &= f(x, z) \\ 0 &= h(x, z) \end{cases} \quad (4.1)$$

Le lecteur notera que cette forme d'état est, dans le cas général, une représentation non causale d'une partie du système à surveiller.

Obtenir pratiquement cette forme d'état n'est pas toujours immédiat. En effet, quelques cas peuvent poser problème, notamment :

- si le schéma de simulation comporte des fonctions non analytiques : tables, saturations, hysteresis ;
- si le schéma de simulation comporte une boucle algébrique pour laquelle il existe une solution numérique, mais pas de solution analytique ;
- si le schéma de simulation comporte une fonction de transfert.

Néanmoins, si l'on part du principe que l'on peut, à partir d'un schéma de simulation, réécrire l'équivalent d'une forme d'état, alors on peut imaginer utiliser n'importe quelle méthode classique de génération des résidus s'appuyant sur une forme d'état, par exemple :

- calcul de relations de parités ;
- conception d'un observateur fondé sur le modèle (4.1). Ceci est réalisé dans (Svärd and Wassén, 2006), où les résidus sont générés de deux façons :

1. par simulation de la forme d'état (comme vu précédemment) ;
2. en utilisant un observateur, quand la simulation de la forme d'état ne donne pas de bons résultats. Le modèle utilisé pour concevoir l'observateur est la forme d'état (4.1) issue du schéma de simulation du sous-système surveillable. On notera à cet égard les travaux exposés dans (Áslund and Frisk, 2006), (Von Wissel *et al.*, 1996), dans lesquels sont considérés des observateurs sur des formes d'état spécifiques en ceci que l'équation de mesure est une équation algébrique, telle la forme (4.1).

4.2 Choix d'un sous-système surveillable

Une deuxième perspective de ce travail concerne le choix des tests de diagnostic. Parmi tous les sous-systèmes surveillables trouvés, peut-on trouver des critères qui permettent d'affirmer que tel chemin d'estimation est meilleur que tel autre ? L'intégration de critères de sensibilité, comme cela est décrit dans (Düstegör *et al.*, 2004c) ou (Düstegör *et al.*, 2004b), est assurément une piste intéressante, mais que nous n'avons pas pu explorer dans ce travail, par manque de temps aussi bien qu'en raison de difficultés de mise en œuvre. Ces approches reposent soit sur une pondération des arcs, soit sur la définition d'une relation

4.3. Comment prendre en compte les modes de fonctionnement ?

d'ordre, l'une et l'autre traduisant un ordre préférentiel de calcul. Les questions soulevées par une telle approche sont :

- En vertu de quels critères définit-on un ordre préférentiel de calcul ?
- Comment intègre-t-on les notions de dynamique des système, voire de stabilité, dans le choix d'un chemin de calcul ?

4.3 Comment prendre en compte les modes de fonctionnement ?

Le modèle de vanne traité au troisième chapitre comprend des blocs de type « *switch* ». Ces *switchs* définissent différents modes de fonctionnement du système. Comment les modéliser ? Nous considérons dans ce mémoire qu'il s'agit de fonctions liant la sortie, les entrées du *switch*, et la condition du *switch*. Nous pourrions également faire plusieurs analyses structurelles selon les différentes configurations des *switchs*. Il s'agit d'une question ouverte.

Un second aspect doit être rattaché à cette question sur les modes de fonctionnement du système : la définition des non-inversibilités. Quelques questions d'ordre pratique se posent : une saturation peut-elle être inversée ? Faut-il lier la question de l'inversibilité d'une saturation au mode de fonctionnement ? De manière plus générale, une fonction peut-elle être inversée dans un mode de fonctionnement, et ne pas l'être dans un autre mode de fonctionnement ? Il en ressort, à notre avis, des questions sur ce qu'est et ce que n'est pas une non-inversibilité, sur la façon de les définir dans chaque mode de fonctionnement, et globalement, sur la modélisation du système et de ses modes de fonctionnement.

4.4 Analyse structurelle par niveaux

Nous avons montré au troisième chapitre que les tables d'isolabilité pouvaient différer selon les hypothèses de calculabilité faites. Analyser un modèle selon différentes hypothèses de calculabilité et savoir ainsi les fautes qu'il est possible de diagnostiquer pour chaque hypothèse, permet d'apprécier la difficulté de concevoir des générateurs de résidus. Nous pourrions considérer plus d'hypothèses que nous ne l'avons fait dans ce mémoire, où nous nous sommes bornés à générer des résidus par propagation numérique des grandeurs connues ; un outil logiciel complet pourrait en effet incorporer d'autres hypothèses, plus ou moins complexes, parfois énoncées dans la littérature : par exemple :

1. les résidus sont obtenus autrement qu'en suivant un chemin de calcul donné par un couplage ;
2. les résidus sont obtenus en suivant le chemin de calcul donné par un couplage :
 - (a) dont la causalité est :
 - i. intégrale ;
 - ii. dérivée ;
 - iii. mixte.
 - (b) dont les boucles algébriques :
 - i. ne sont pas solvables ;

Chapitre 4. Conclusion et perspectives

- ii. sont solvables ;
- iii. sont solvables seulement si elles sont linéaires.

Deuxième partie

Etude pour la conception du diagnostic fonctionnel de la boucle d'air d'un moteur diesel (confidentiel)

Conclusion

Ce mémoire de doctorat comporte deux parties. La première concerne la présentation de la méthodologie, l'analyse structurelle, des résultats théoriques obtenus, et l'illustration de ceux-ci sur un exemple. La seconde présente l'analyse du système boucle d'air.

L'objectif de cette thèse est de construire tous les tests fonctionnels possibles afin de déterminer les potentialités de diagnostic du système. Nous nous sommes placés dans un contexte où le système est modélisé sous forme d'un schéma de simulation comportant de nombreux blocs, dont certains représentent des relations non analytiques.

Du fait de la nature et de la taille des modèles considérés, ainsi que de la volonté de réduire le temps de calcul de la recherche exhaustive des tests fonctionnels, notre travail s'est porté sur la définition et l'utilisation de critères traduisant la possibilité de générer un test implantable. Le second aspect permettant de réduire ce temps de calcul concerne l'intégration d'un cahier des charges. Les algorithmes développés ont été présentés au chapitre 2.

Deux exemples sont traités dans ce rapport : l'un porte sur un modèle de vanne qui fût l'objet d'un benchmark, dans le cadre du projet européen *DAMADICS* (chapitre 3) ; l'autre porte sur la boucle d'air du moteur *DV6C* (chapitre 6). Dans chacun des cas, nous présentons différentes hypothèses d'analyse, qui reflètent notre volonté de hiérarchiser et de classer les tests trouvés.

Parmi les résultats les plus intéressants, d'un point de vue industriel, de ce travail, nous avons montré, au chapitre 6, qu'il était possible de discriminer deux types d'encrassement de la boucle d'air. Nous avons également réussi, grâce aux algorithmes développés, à mener une analyse structurelle sur un système de grande taille. Cette maîtrise de la complexité a été atteinte par la prise en compte des inversibilités et d'un cahier des charges, par le rejet des boucles algébriques, par le choix de la contrainte redondante, et enfin, par le non-mélange des causalités.

A l'issue de ce travail, bien des perspectives sont envisageables. Nous ne reviendrons pas en détail sur celles-ci ; elles sont exposées dans les conclusions de chacune des parties. Elles consistent en l'amélioration des algorithmes développés (par la prise en compte d'autres hypothèses de calculabilité que celles énoncées dans ce mémoire), autant qu'en la simplification des modèles, l'étude du diagnostic actif, le choix des tests de diagnostic ou la robustesse des estimateurs.

Conclusion

Bibliographie

- (Armengol *et al.*, 2009) J. Armengol, A. Bregon, T. Escobet, E. Gelso, M. Krysander, M. Nyberg, X. Olive, B. Pulido, and L. Travé-Massuyès. Minimal Structurally Overdetermined sets for residual generation : A comparison of alternative approaches. *Proceedings of the IFAC-Safeprocess 2009*, 2009.
- (Åslund and Frisk, 2006) J. Åslund and E. Frisk. An observer for non-linear differential-algebraic systems. *Automatica*, 42(6) :959–965, 2006.
- (Basseville and Nikiforov, 1993) M. Basseville and I.V. Nikiforov. *Detection of abrupt changes : theory and application*, volume 10. Citeseer, 1993.
- (Blanke *et al.*, 2006) M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer Verlag, 2006.
- (Bocaniala and da Costa, 2006) C.D. Bocaniala and J.S. da Costa. Application of a novel fuzzy classifier to fault detection and isolation of the damadics benchmark problem. *Control Engineering Practice*, 14(6) :653 – 669, 2006.
- (Boukhobza and Hamelin, 2007) T. Boukhobza and F. Hamelin. Observability analysis for structured bilinear systems : a graph-theoretic approach. *Automatica*, 43(11) :1968–1974, 2007.
- (Boukhobza *et al.*, 2006) T. Boukhobza, F. Hamelin, and D. Sauter. Observability of structured linear systems in descriptor form : a graph-theoretic approach. *Automatica*, 42(4) :629–635, 2006.
- (Brunet *et al.*, 1990) J. Brunet, D. Jaume, M. Labarrère, A. Rault, and M. Vergé. *Détection et diagnostic de pannes : approche par modélisation*. Hermès, 1990.
- (Calado *et al.*, 2006) J.M.F. Calado, J.M.G. Sa da Costa, M. Bartys, and J. Korbicz. Fdi approach to the damadics benchmark problem based on qualitative reasoning coupled with fuzzy neural networks. *Control Engineering Practice*, 14(6) :685 – 698, 2006.
- (Calderón-Espinoza *et al.*, 2007) G. Calderón-Espinoza, J. Armengol, J. Vehí, and E.R. Gelso. Dynamic diagnosis based on interval analytical redundancy relations and signs of the symptoms. *AI Communications*, 20(1) :39–47, 2007.
- (Cassar *et al.*, 1994) J.P. Cassar, RG Litwak, V. Cocquempot, and M. Staroswiecki. Approche structurelle de la conception de systèmes de surveillance pour des procédés industriels complexes. *Revue Européenne de Diagnostic et Sécurité de fonctionnement*, 4(2) :179–202, 1994.
- (Chen and Patton, 1999) J. Chen and R.J. Patton. *Robust model-based fault diagnosis for dynamic systems*. Kluwer Academic Publishers Norwell, MA, USA, Boston, Dordrecht, London, 1999.

Bibliographie

- (Cocquempot *et al.*, 1991) V. Cocquempot, J.P. Cassar, and M. Staroswiecki. Generation of robust analytical redundancy relations. 1991.
- (Commault and Dion, 2007) C. Commault and J.M. Dion. Sensor location for diagnosis in linear systems : a structural analysis. *IEEE Transactions on Automatic Control*, 52(2) :155–169, 2007.
- (Conrard *et al.*, 2009) B. Conrard, V. Cocquempot, and M. Bayart. Sensor and Actuator Placement with Dependability Constraints and a Cost Criterion. *Proceedings of the IFAC-Safeprocess 2009*, 2009.
- (Cordier *et al.*, 2000) M.O. Cordier, P. Dague, M. Dumas, F. Levy, J. Montmain, M. Staroswiecki, and L. Travé-Massuyès. A comparative analysis of AI and control theory approaches to model-based diagnosis. *14th European Conference on Artificial Intelligence*, pages 136–140, 2000.
- (Cordier *et al.*, 2004) M.O. Cordier, P. Dague, F. Levy, J. Montmain, M. Staroswiecki, and L. Travé-Massuyès. Conflicts versus analytical redundancy relations : a comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives. *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, 34(5) :2163–2177, 2004.
- (de Flaugergues *et al.*, 2009a) V. de Flaugergues, V. Cocquempot, M. Bayart, and N. Lefebvre. Exhaustive search of residuals computation schemes using macro-graphs and invertibility constraints. *7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SafeProcess09*, 2009.
- (de Flaugergues *et al.*, 2009b) V. de Flaugergues, V. Cocquempot, M. Bayart, and M. Pengov. A modified, invertibility-based canonical decomposition. *20th International Workshop on Principles of Diagnosis, DX09*, 2009.
- (de Flaugergues *et al.*, 2010) V. de Flaugergues, V. Cocquempot, M. Bayart, and M. Pengov. On non-invertibilities for Structural Analysis. *21st International Workshop on Principles of Diagnosis, DX10*, 2010.
- (De Kleer and Williams, 1987) J. De Kleer and B.C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1) :97–130, 1987.
- (Declerck and Staroswiecki, 1991) P. Declerck and M. Staroswiecki. Characterization of the canonical components of a structural graph for fault detection in large scale industrial plants. *Proceedings of ECC 91*, pages 298–303, 1991.
- (Dion *et al.*, 2003) J.M. Dion, C. Commault, and J. van der Woude. Generic properties and control of linear structured systems : a survey. *Automatica*, 39(7) :1125–1144, 2003.
- (Djeziri *et al.*, 2009) MA Djeziri, B. Ould Bouamama, and R. Merzouki. Modelling and robust FDI of steam generator using uncertain bond graph model. *Journal of Process Control*, 19(1) :149–162, 2009.
- (Dulmage and Mendelsohn, 1958) AL Dulmage and NS Mendelsohn. Coverings of bipartite graphs. *Canad. J. Math*, 10 :517–534, 1958.
- (Düstegör *et al.*, 2004a) D. Düstegör, V. Cocquempot, and M. Staroswiecki. Structural analysis for fault detection and identification : an algorithmic study. *Proceedings of the 2nd Symposium on System Structure and Control 2004 (SSSC'04), Oaxaca, Mexique, 7-11 december*, 2004.
- (Düstegör *et al.*, 2004b) D. Düstegör, V. Cocquempot, and M. Staroswiecki. Structural Analysis for Residual Generation : Implementation Issues Consideration. 2004.

- (Düstegör *et al.*, 2004c) D. Düstegör, V. Cocquempot, and M. Staroswiecki. Structural analysis for residual generation : towards implementation. 2, 2004.
- (Düstegör *et al.*, 2004d) D. Düstegör, V. Coquempot, M. Staroswiecki, and E. Frisk. Isolabilité structurelle des défaillances - Application à un modèle de vanne. *Journal européen des systèmes automatisés*, 39 :103–124, 2004.
- (Düstegör *et al.*, 2006) D. Düstegör, E. Frisk, V. Cocquempot, M. Krysander, and M. Staroswiecki. Structural analysis of fault isolability in the DAMADICS benchmark. *Control Engineering Practice*, 14(6) :597–608, 2006.
- (Düstegör, 2005) D. Düstegör. *Aspects Algorithmiques de l'Analyse Structurelle pour la Surveillance*. PhD thesis, Université des Sciences et Technologies de Lille, LAGIS, 2005.
- (Frisk and Krysander, 2007) E. Frisk and M. Krysander. Sensor placement for maximum fault isolability. *Proceedings DX-2007, Nashville, USA*, 2007.
- (Frisk *et al.*, 2010) E. Frisk, A. Bregon, and J. Aslund. Diagnosability Analysis Considering Causal Interpretations for Differential Constraints. *21st International Workshop on Principles of Diagnosis, DX10*, 2010.
- (Frisk, 2000) E. Frisk. Residual Generator Design For Non-Linear, Polynomial Systems – A Grobner Basis Approach. pages 979–984, 2000.
- (Gelso *et al.*, 2008) E.R. Gelso, S.M. Castillo, and J. Armengol. An algorithm based on structural analysis for model-based fault diagnosis. page 138, 2008.
- (Gentil and Montmain, 2004) S. Gentil and J. Montmain. Hierarchical representation of complex systems for supporting human decision making. *Advanced Engineering Informatics*, 18(3) :143–159, 2004.
- (Gentil *et al.*, 2004) S. Gentil, J. Montmain, and C. Combastel. Combining FDI and AI approaches within causal-model-based diagnosis. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 34(5) :2207–2221, 2004.
- (Guernez *et al.*, 1997) C. Guernez, J.P. Cassar, and M. Staroswiecki. Extension of parity space to nonlinear polynomial dynamic systems. *Proceedings of Safeprocess 1997*, 1997.
- (Hopcroft and Karp, 1973) J.E. Hopcroft and R.M. Karp. An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs. *SIAM Journal on Computing*, 2 :225, 1973.
- (Isermann and Ballé, 1997) R. Isermann and P. Ballé. Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5(5) :709–719, 1997.
- (Izadi-Zamanabadi, 1999) R. Izadi-Zamanabadi. *Fault-tolerant Supervisory Control : System Analysis and Logic Design*. PhD thesis, Aalborg University, 1999.
- (Kallesoe *et al.*, 2006) CS Kallesoe, V. Cocquempot, and R. Izadi-Zamanabadi. Model based fault detection in a centrifugal pump application. *Control Systems Technology, IEEE Transactions on*, 14(2) :204–215, 2006.
- (Krysander *et al.*, 2008) M. Krysander, J. Aslund, and M. Nyberg. An efficient algorithm for finding minimal overconstrained subsystems for model-based diagnosis. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 38(1) :197–206, 2008.
- (Krysander *et al.*, 2010) M. Krysander, J. Åslund, and E. Frisk. A Structural Algorithm for Finding Testable Sub-models and Multiple Fault Isolability Analysis. *21st International Workshop on Principles of Diagnosis, DX10*, 2010.

Bibliographie

- (Krysander, 2006) M. Krysander. *Design and analysis of diagnostic systems utilizing structural methods*. Department of Electrical Engineering, Linköpings universitet, 2006.
- (Maquin *et al.*, 1997) D. Maquin, V. Cocquempot, JP Cassar, M. Staroswiecki, and J. Ragot. Generation of Analytical Redundancy Relations for FDI purposes. *IFAC Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives, SDEM-PED'97, Carry-le Rouet, France*, pages 86–93, 1997.
- (Montmain and Gentil, 2000) J. Montmain and S. Gentil. Dynamic causal model diagnostic reasoning for online technical process supervision. *Automatica*, 36 :1137–1152, 2000.
- (Murota, 2000) K. Murota. *Matrices and Matroids for Systems Analysis*. Springer, 2000.
- (Patton *et al.*, 2000) R. Patton, P.M. Frank, and R. Clark. *Issues of fault diagnosis for dynamic systems*. Springer Verlag, 2000.
- (Ploix and Giuliani, 2010) S. Ploix and S. Giuliani. Glossaire des termes usuels du diagnostic de défauts. <http://www.univ-valenciennes.fr/GDR-MACS/local/Grenoble/www-lag.ensieg.inpg.fr/ploix/glossaire/DiagnosticGlossaire.htm>, 2010.
- (Ploix *et al.*, 2003) S. Ploix, S. Touaf, and JM Flaus. A logical diagnostic method for complex dynamic systems. *Qualita 2003, Nancy, France*, 2003.
- (Ploix, 1998) S. Ploix. *Diagnostic des systèmes incertains : l'approche bornante*. PhD thesis, National Polytechnic Institute of Lorraine (INPL), 1998.
- (Previdi and Parisini, 2006) F. Previdi and T. Parisini. Model-free actuator fault detection using a spectral estimation approach : the case of the damadics benchmark problem. *Control Engineering Practice*, 14(6) :635 – 644, 2006.
- (Puig *et al.*, 2006) V. Puig, A. Stancu, T. Escobet, F. Nejjari, J. Quevedo, and R.J. Patton. Passive robust fault detection using interval observers : Application to the damadics benchmark problem. *Control Engineering Practice*, 14(6) :621 – 633, 2006.
- (Pulido and Gonzalez, 2004) B. Pulido and CA Gonzalez. Possible conflicts : a compilation technique for consistency-based diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 34(5) :2192–2206, 2004.
- (Pulido *et al.*, 2008) B. Pulido, A. Bregón, and C. Alonso. Combining state estimation and simulation in consistency-based diagnosis using possible conflicts. pages 339–346, 2008.
- (Rosich *et al.*, 2009) A. Rosich, E. Frisk, J. Åslund, R. Sarrate, and F. Nejjari. Sensor placement for fault diagnosis based on causal computations. 9 :402–407, 2009.
- (Samantaray *et al.*, 2006) AK Samantaray, K. Medjaher, B. Ould Bouamama, M. Staroswiecki, and G. Dauphin-Tanguy. Diagnostic bond graphs for online fault detection and isolation. *Simulation Modelling Practice and Theory*, 14(3) :237–262, 2006.
- (Shampine *et al.*, 1999) L.F. Shampine, M.W. Reichelt, and J.A. Kierzenka. Solving index-I DAEs in MATLAB and Simulink. *Siam Review*, 41(3) :538–552, 1999.
- (Staroswiecki and Comtet-Varga, 2001) M. Staroswiecki and G. Comtet-Varga. Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica*, 37(5) :687–699, 2001.
- (Staroswiecki, 2000) M. Staroswiecki. Surveillance : le cadre linéaire et quelques extensions. *Conférence Internationale Francophone d'Automatique CIFA 2000*, 2000.

- (Supavatanakul *et al.*, 2006) P. Supavatanakul, J. Lunze, V. Puig, and J. Quevedo. Diagnosis of timed automata : Theory and application to the damadics actuator benchmark problem. *Control Engineering Practice*, 14(6) :609 – 619, 2006.
- (Svärd and Nyberg, 2008) C. Svärd and M. Nyberg. A Mixed Causality Approach to Residual Generation Utilizing Equation System Solvers and Differential-Algebraic Equation Theory. *19th International Workshop on Principles of Diagnosis, DX08*, 2008.
- (Svärd and Wassén, 2006) C. Svärd and H. Wassén. *Development of Methods for Automatic Design of Residual Generators*. PhD thesis, Linköping University, Department of Electrical Engineering, 2006.
- (Touaf, 2005) S. Touaf. *Diagnostic logique des systèmes complexes et dynamiques dans un contexte multi-agent*. PhD thesis, Université Joseph Fourier, Grenoble 1, France, 2005.
- (Travé-Massuyès *et al.*, 2006) L. Travé-Massuyès, T. Escobet, X. Olive, and T. CNRS. Diagnosability analysis based on component-supported analytical redundancy relations. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 36(6) :1146–1160, 2006.
- (Uppal *et al.*, 2006) F.J. Uppal, R.J. Patton, and M. Witczak. A neuro-fuzzy multiple-model observer approach to robust fault diagnosis based on the damadics benchmark problem. *Control Engineering Practice*, 14(6) :699 – 717, 2006.
- (Von Wissel *et al.*, 1996) D. Von Wissel, R. Nikoukhah, SL Campbell, and F. Delebecque. Nonlinear observer design using implicit system descriptions. *Proc. of CESA'96, Lille, France*, 1996.
- (Witczak *et al.*, 2006) M. Witczak, J. Korbicz, M. Mrugalski, and R.J. Patton. A gmdh neural network-based approach to robust fault diagnosis : Application to the damadics benchmark problem. *Control Engineering Practice*, 14(6) :671 – 683, 2006.
- (WUT, 2002) WUT. DAMADICS (Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems) RTN Information Web site. [http ://diag.mchtr.pw.edu.pl/damadics/](http://diag.mchtr.pw.edu.pl/damadics/), 2002.
- (Zwingelstein, 1995) G. Zwingelstein. *Diagnostic des défaillances : Théorie et pratique pour les systèmes industriels*. Hermès, 1995.