



HAL
open science

Contribution de l'outil bond graph pour la conception de systèmes de supervision des processus industriels

Kamal Medjaher

► **To cite this version:**

Kamal Medjaher. Contribution de l'outil bond graph pour la conception de systèmes de supervision des processus industriels. Sciences de l'ingénieur [physics]. Université des sciences et technologies de Lille 1, 2005. Français. NNT: . tel-01736573

HAL Id: tel-01736573

<https://hal.science/tel-01736573v1>

Submitted on 17 Mar 2018

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

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'Ordre : 3654

Université des Sciences et Technologies de Lille

Laboratoire d'Automatique Génie Informatique et Signal
UMR CNRS 8146

THÈSE

Présentée en vue de l'obtention du grade de

DOCTEUR DE L'UNIVERSITÉ

Spécialité : Automatique et Informatique Industrielle

par

Kamal MEDJAHER

Ingénieur Université M. MAMMERI.

Directeur de thèse : Prof. Belkacem OULD BOUAMAMA

CONTRIBUTION DE L'OUTIL BOND GRAPH POUR LA CONCEPTION DE SYSTÈMES DE SUPERVISION DES PROCESSUS INDUSTRIELS

Soutenue publiquement le 7 juillet 2005 devant la commission d'examen :

Président	Geneviève DAUPHIN-TANGUY	Professeur, Ecole Centrale de Lille.
Rapporteurs	José RAGOT	Professeur, Institut National Polytechnique de Lorraine, CRAN, Nancy.
	Kamal YUCEF-TOUMI	Professeur, MIT, Cambridge, USA.
Examineurs	Belkacem OULD BOUAMAMA	Professeur, Ecole Polytechnique Universitaire de Lille, Directeur de thèse.
	Arun Kumar SAMANTARAY	Maître assistant, Institut Indien de Technologie, Kharagpur, Inde,
	Bertrand BRAUNSCHWEIG	Docteur, HDR, Directeur Expert, Institut Français du Pétrole.
	Jean-Charles CADIOU	Professeur, Université de Versailles (UVSQ), Versailles.

Table des matières

Introduction générale	1
1 Cadre et objectifs	1
2 Positionnement du travail de thèse	1
2.1 Ce qui existe	1
2.2 Ce que nous proposons	2
3 Résultats des travaux de thèse	4
4 Organisation du mémoire	4
1 Génération de modèles dynamiques en génie des procédés	7
1.1 Introduction	7
1.2 Classification fonctionnelle des procédés thermofluides	8
1.2.1 Le tétraèdre d'état et les procédés énergétiques	8
1.2.2 Classes de procédés thermofluides	10
1.3 Modèles bond graphs des classes thermofluides	11
1.3.1 Sources thermofluides	12
1.3.2 Le multiport thermofluide R	13
1.3.2.1 Conduite avec restriction	13
1.3.2.2 Conduite connectée à une pompe	16
1.3.3 Multiport thermofluide C	19
1.3.3.1 Accumulateur thermofluide sous-saturé	20
1.3.3.2 Accumulateur thermofluide saturé	22
1.3.4 Classes détecteurs et régulateurs	25
1.3.5 Classe connecteurs	26
1.4 Syntaxe d'interconnexion des classes de procédés	26
1.4.1 Règles d'interconnexion	26
1.4.2 Règles d'équivalence	28
1.4.3 Exemple de chaînes	29

1.4.4	Délimitation structurelle	29
1.5	Génération du modèle global	31
1.5.1	Exemple	31
1.5.2	Présence de boucles algébriques	35
1.6	Contribution au placement de capteurs	37
1.7	Conclusion	41
2	Génération d’algorithmes pour le diagnostic	43
2.1	Introduction	43
Partie I	Génération de Relations de Redondance Analytique	45
2.2	Représentation graphique des systèmes	47
2.2.1	Représentation par un modèle structurel	47
2.2.2	Représentation par un digraphe	49
2.2.3	Représentation par un graphe biparti	49
2.2.4	Représentation par bond graph	51
2.2.4.1	Les contraintes	51
2.2.4.2	Les variables	54
2.2.4.3	Les paramètres	55
2.3	Génération de RRAs	55
2.3.1	Génération de RRAs en utilisant le graphe biparti	55
2.3.2	Génération de RRAs en utilisant le bond graph	57
2.3.2.1	Algorithme de génération de RRAs par bond graph	58
2.3.2.2	Matrice de signature des défaillances	64
2.4	Procédure de détection et d’isolation des défaillances	65
2.5	Conclusion	66
Partie II	Cas de contraintes causales non-inversibles	67
2.6	Introduction	69
2.7	Contraintes non-inversibles	69
2.8	Détecteurs de résidus numériques	71
2.8.1	Modifications sur le modèle bond graph en causalité dérivée	72
2.8.2	Génération des résidus	75
2.8.2.1	Exemple d’application	76
2.8.2.2	Matrice de signature des défaillances	81

2.8.2.3	Génération de résidus numériques	83
2.8.2.4	Construction du modèle de simulation	84
2.8.2.5	Résultats de simulation	87
2.9	Conclusion	92
Partie III Bond Graph de Diagnostic Étendu		93
2.10	Bond Graph de Diagnostic Étendu	95
2.10.1	Choix de la causalité	98
2.10.2	Problèmes de couplage entre variables en bond graph	100
3	Informatisation des procédures développées	103
3.1	Introduction	103
3.2	État de l'art des logiciels existants	103
3.3	Représentation du P&ID par bond graph	106
3.4	Choix du logiciel pour l'implémentation	111
3.5	Implémentation de la boîte à outils	111
3.6	Validation du modèle	113
3.6.1	Simulation de défaillances et réponses des résidus	116
3.7	Exemple d'illustration: banc d'essai du projet CHEM	118
3.8	Conclusion	125
4	Application	127
4.1	Introduction	127
4.2	Application: le générateur de vapeur	128
4.2.1	Description du processus	128
4.2.2	Modèle bond graph des composants du générateur de vapeur	130
4.2.3	Modèle bond graph et architectural du générateur de vapeur	134
4.2.4	Modèle dynamique et sa validation	136
4.2.5	Génération de RRAs	142
4.2.6	Surveillabilité et placement de capteurs	145
4.2.7	Implémentation en temps réel	148
4.2.7.1	Matrice pratique de signature des défaillances	148
4.2.7.2	Procédure de décision	150
4.2.7.3	Résultats expérimentaux	151
4.3	Conclusion	158

Conclusions et perspectives	159
1 Conclusions générales	159
2 Perspectives	160
Bibliographie	163

Introduction générale

1 Cadre et objectifs

Les travaux de la présente thèse ont été réalisés dans le cadre du projet Européen CHEM [1]. Le but de ce projet consiste à développer et implémenter un système d'aide à la décision sous forme modulaire pour la supervision des processus de raffinage, chimiques et pétrochimiques. Le projet combine les expertises de cinq établissements académiques (dont le LAGIS), et dix industriels. Le projet est divisé en 12 groupes de travail et composé de 23 boîtes à outils.

Sur le plan académique, les travaux de recherche ont été développés au sein d'une opération transversale 'Bond Graph-Surveillance' entre les équipes 'Bond Graph' et 'Sûreté de Fonctionnement des Systèmes Dynamiques (SFSD)' du Laboratoire d'Automatique, Génie Informatique et Signal (LAGIS).

L'objectif de cette thèse est la conception d'un système de supervision à base de modèles en utilisant un seul outil: le bond graph. Sur le plan méthodologique, le travail consiste à automatiser les procédures de génération de modèles et d'indicateurs de défaillances sous forme formelle et interchangeable pour les intégrer dans le système de supervision. Sur le plan industriel, les résultats obtenus ont été appliqués sur des installations réelles: le générateur de vapeur du LAGIS et le pilote de FCC (Fluid Catalytic Cracking) de l'Institut Français du Pétrole.

2 Positionnement du travail de thèse

2.1 Ce qui existe

L'outil bond graph [2] a été initialement utilisé pour la modélisation des systèmes physiques. La démarche habituelle des utilisateurs de cet outil consiste à considérer le modèle bond graph comme un modèle de connaissance pour la simulation et la commande des systèmes dynamiques [3, 4, 5, 6]. L'idée d'utiliser une seule représentation (le bond graph) pour la modélisation, l'analyse et la synthèse des lois de commande en exploitant la causalité est récente. Plusieurs travaux ont été développés dans ce domaine [7, 8, 9, 10].

La surveillance des processus physiques, avec ses aspects détection et localisation des défaillances, mais aussi le choix et le placement des capteurs trouve aussi un intérêt à l'existence d'un modèle bond graph. L'idée d'utiliser l'outil bond graph pour la surveillance est née de la collaboration entre les équipes "Bond Graph" et "SFSD" du LAGIS dans le cadre de la thèse de Tagina [11]. Fort des résultats obtenus, l'intérêt de poursuivre cette collaboration a été concrétisé par la création en 1998 de l'opération transversale au sein du laboratoire.

Concernant l'utilisation des bond graphs pour les problèmes de surveillance, on trouve quelques équipes qui utilisent cette démarche dans le monde. On peut citer par exemple les travaux récents sur le diagnostic des systèmes à l'aide des bond graphs qualitatifs [12, 13, 14]. Les difficultés de ces approches résident dans le fait que les modèles qualitatifs peuvent avoir plusieurs comportements possibles (l'augmentation d'un débit d'eau dans un réservoir peut aussi bien augmenter que diminuer la température stockée dans l'accumulateur). Il n'est pas aisé de déterminer les valeurs limites inférieures et supérieures de déviation. D'autre part un problème combinatoire peut apparaître lors des procédures d'inférences pour les systèmes complexes.

Les premiers travaux relatifs à la surveillance des systèmes à l'aide des modèles bond graphs quantitatifs reposent essentiellement sur l'analyse des propriétés causales et structurelles du modèle [15]. Ce dernier, qui est avant tout un graphe, est utilisé pour la génération d'une manière générique des indicateurs de défaillances (communément appelés Relations de Redondance Analytique: RRA) et pour analyser les conditions de surveillabilité des sources (actionneurs) directement sur le bond graph. Les travaux actuellement existant dans la littérature sur l'analyse structurelle des systèmes sont basés sur un modèle bond graph monoénergie décrivant des processus physiques régis par une seule énergie à la fois (généralement mécanique ou électrique).

Dans le cas où les processus à surveiller sont décrits par des pseudo-bond graphs multi-énergies (modélisant les processus thermofluides présents dans de très nombreuses industries chimiques et pétrochimiques souvent à risques), l'utilisation directe de ces approches n'est plus possible. En effet, les modèles bond graphs décrivant ces processus sont non linéaires, et "multiports": les lois caractérisant les phénomènes physiques modélisés ne sont plus scalaires. Le LAGIS a déjà développé les outils de modélisation par bond graph de ce type de systèmes. Ces travaux ont fait l'objet de publications d'un certain nombre de travaux ainsi que d'applications industrielles [16][17][18].

2.2 Ce que nous proposons

Dans les approches de surveillance à base de modèles quantitatifs les performances des procédures de détection et de localisation de défaillances dépendent fortement du modèle utilisé. Une fois ce dernier généré, les indicateurs de défaillances peuvent être déduits. L'obtention du

modèle est une tâche complexe et difficile plus particulièrement pour les processus en génie des procédés en raison de leur diversité et du couplage des énergies qui les caractérisent.

L'objectif du travail de recherche développé dans ce mémoire est donc: l'utilisation des bond graphs pour la conception, sous forme d'une boîte à outils, d'un générateur de modèles dynamiques formels des procédés énergétiques afin de les surveiller. Nous proposons de mettre à la disposition de l'utilisateur un outil lui permettant de surveiller le processus sans avoir à recourir ni à la modélisation ni à la génération manuelle des RRAs qui reste une tâche complexe surtout pour les modèles non linéaires. Ces modèles générés sous forme formelle seront utilisés hors ligne par d'autres boîtes à outils pour leur intégration dans le système global de supervision développé dans CHEM. Ces procédures ont été alors automatisées à l'aide d'une boîte à outils appelée ModelBuilder [19]. Cette dernière est présentée avec une interface graphique de telle sorte que l'utilisateur puisse aisément construire les modèles dynamiques et architecturaux de la plupart des processus thermofluides (en se basant sur le Plan des Instruments Détaillé (P&ID) de l'installation) et générer les RRAs et la surveillabilité du processus (définir à l'opérateur les composants pouvant être surveillés en fonction du cahier des charges spécifié). L'objectif recherché de notre travail peut donc être résumé à la réponse à la question suivante: comment peut-on surveiller le processus énergétique à partir d'une simple interconnexion graphique de ses composants? Les différentes étapes pour la réalisation de cette tâche sont présentées sur la figure 1.

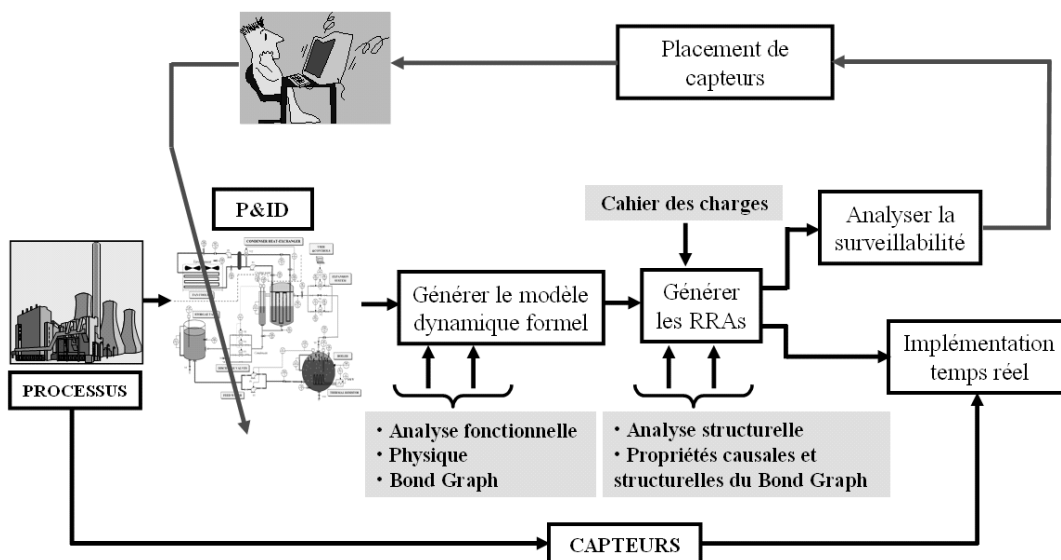


FIG. 1 – Conception intégrée: du P&ID à l'analyse de la surveillabilité

3 Résultats des travaux de thèse

Au regard des travaux existant sur ce thème, notre contribution en tenant compte des contraintes posées par le caractère applicatif du travail se situe à plusieurs niveaux.

1. La démarche développée est une approche complète pour la conception d'un système de supervision : elle consiste à générer des modèles dynamiques (en mode normal et défaillant), des algorithmes de surveillance sous forme symbolique ou numérique à partir non pas des équations mathématiques mais du procédé physique à surveiller. La démarche est générique et flexible et n'utilise qu'une seule représentation: le bond graph.
2. Le cahier des charges fixé pour le diagnostic concerne la détection et la localisation des défaillances au niveau des capteurs, des actionneurs, des composants physiques, des contrôleurs et des algorithmes de commande grâce à une démarche modulaire.
3. L'algorithme de génération des RRAs à partir du modèle bond graph n'est pas seulement limité à des formes particulières du modèle (polynomiale pour la théorie de l'élimination ou linéaire pour la méthode par projection dans le cas de l'espace de parité) mais aussi à des modèles donnés sous forme empiriques (cas des processus chimiques).
4. Enfin, la thématique développée dans ce projet est une recherche finalisée. Les outils développés ont été implémentés dans le logiciel SYMBOLS 2000 sous forme d'une boîte à outil nommée "ModelBuilder" générant les modèles dynamiques, les RRAs et l'analyse de la surveillabilité des processus industriels. Une base de données générique constituée des classes de processus, de capteurs, d'actionneurs et de contrôleurs permet à l'utilisateur d'introduire, sous forme de Plan des Instruments Détaillés (P&ID), le processus à superviser. La sortie de ModelBuilder est utilisée par d'autres modules du superviseur global pour l'intégration finale. Enfin, la boîte à outils développée est appliquée à des procédés énergétiques (générateur de vapeur) et pétrochimiques (craquage catalytique).

Les résultats de recherche dans le cadre de ma thèse ont fait l'objet de quatre publications dans deux revues [19, 20, 21] et [22].

4 Organisation du mémoire

Le présent mémoire est composé, en plus de l'introduction générale, de quatre chapitres et d'une conclusion.

Chapitre 1: Génération de modèles dynamiques en génie des procédés

Dans ce chapitre nous proposons d'abord une classification fonctionnelle des différents composants en génie des procédés. Ensuite, une syntaxe et des règles de connexion entre les différentes classes de composants ont été établies. Ces règles, basées sur une analyse fonctionnelle et comportementale (modèle bond graph), garantissent la validité du modèle architectu-

ral global. Enfin, est exposée la méthode de génération des modèles dynamiques en utilisant l'outil bond graph.

Chapitre 2: Génération d'algorithmes pour le diagnostic

Ce chapitre présente les algorithmes et les procédures pour la génération d'indicateurs de défaillances permettant la surveillance du processus (RRAs, matrice de signature des défaillances, analyse de la surveillabilité). Nous l'avons divisé en trois parties principales. Dans la première partie, après avoir revu les différentes représentations graphiques des systèmes dynamiques, nous présenterons un algorithme pour la génération de RRAs et de la matrice des signatures correspondante. Les RRAs générées sont alors présentées sous forme de relations symboliques obtenues en éliminant les variables inconnues du modèle bond graph. Toutefois, dans certains cas (présence de relations non inversibles) l'élimination des variables inconnues n'est pas possible à réaliser. Nous proposerons alors dans la deuxième partie une nouvelle méthode de génération de résidus numériques sans passer par l'élimination des variables inconnues. Pour cela, des modifications sont apportées au modèle bond graph initial pour introduire ce que nous appelons les détecteurs de résidus numériques. Enfin, dans la troisième partie du chapitre nous proposons les schémas et méthodes pour leur implémentation en temps réel.

Chapitre 3: Informatisation des procédures

Les procédures et les algorithmes développés dans les deux premiers chapitres sont implémentés dans une boîte à outils appelée ModelBuilder et intégrée au logiciel SYMBOLS 2000. Dans ce chapitre, nous montrerons à travers un exemple simple (le banc d'essai du projet CHEM) les différentes fonctionnalités de la boîte à outils.

Chapitre 4: Applications

Dans le quatrième et dernier chapitre du présent mémoire, nous présenterons l'application des outils développés sous forme d'une boîte à outils sur deux types de procédés industriels différents. Le premier est un procédé énergétique (le générateur de vapeur) du LAGIS et le second est un procédé physico-chimique (le craquage catalytique) de l'Institut Français du Pétrole. Ces deux applications nous permettront de tester les fonctionnalités de la boîte à outils et de valider l'approche et les algorithmes développés.

Enfin, une conclusion générale et des perspectives sont données à la fin du mémoire.

Chapitre 1

Génération de modèles dynamiques en génie des procédés

1.1 Introduction

Le but de ce chapitre est de développer une méthodologie pour la génération de modèles dynamiques formels des composants thermofluides. Ces modèles seront utilisés pour fournir à l'utilisateur les outils et les algorithmes nécessaires pour la détection et la localisation des défaillances sur des procédés réels. La démarche consiste, en premier lieu, à établir une classification fonctionnelle des composants et ensuite de générer les modèles bond graphs correspondants permettant d'obtenir les équations dynamiques de chaque composant ainsi que les Relations de Redondance Analytique (RRAs). Une syntaxe pour établir et vérifier les règles de connexion entre les différentes classes de composants et pour garantir la validité du modèle architectural est exposée sur la base d'une analyse fonctionnelle [23] et comportementale. Une fois la bibliothèque des composants construite, l'utilisateur pourra sélectionner les éléments formant le Plan des Instruments Détaillé (P&ID) de son système pour générer le modèle bond graph, les équations dynamiques et les outils de détection et de localisation des défaillances (RRAs et matrice de signature de défaillances). En outre, dans le cas où le cahier des charges (relatif à la surveillance) n'est pas satisfait, l'utilisateur pourra, grâce à l'analyse du modèle bond graph global et des RRAs, optimiser le placement de capteurs pour améliorer la surveillabilité du processus industriel.

La modélisation des systèmes thermofluides est une tâche difficile à cause du couplage des phénomènes hydraulique et thermique. L'outil bond graph [2, 3, 6, 4] en tant qu'outil flexible et fédérateur est alors utilisé, dans une démarche intégrée, pour la modélisation, l'analyse, la synthèse et la surveillance de tels procédés multi-énergies. Les propriétés causales et structurelles des bond graphs permettent d'un côté de vérifier la validité et la consistance des modèles dynamiques générés et de l'autre côté d'analyser la commandabilité, l'observabilité

et la surveillabilité du système sans faire de calcul formel. En outre, à partir du modèle bond graph global, on peut facilement délimiter des zones autour des structures des jonctions pour créer des sous-modèles bond graphs représentant les composants élémentaires du système.

Ce chapitre est organisé comme suit: après avoir défini les différentes classes de composants thermofluides, la première partie présente les modèles bond graphs et les équations de comportement de chaque classe. Ensuite, dans une deuxième partie, on présentera la syntaxe et les règles d'interconnexion des différentes classes (ou composants) afin de créer des modèles fonctionnels et dynamiques valides. Enfin, nous proposerons dans la dernière partie une contribution au placement de capteurs afin d'améliorer les propriétés structurelles et la surveillabilité du système. Les résultats de recherche de ce chapitre ont fait l'objet de deux publications dans deux revues [22, 19].

1.2 Classification fonctionnelle des procédés thermofluides

1.2.1 Le tétraèdre d'état et les procédés énergétiques

Dans une plateforme de supervision, l'intégration des modèles fonctionnels est très utile pour l'opérateur parce qu'il lui est plus facile de considérer le fonctionnement du système en terme de ces fonctions qu'en terme d'un modèle mathématique souvent compliqué et abstrait. En s'inspirant des travaux de Murakami [24] et [23], notre système physique sera défini comme un ensemble de procédés interconnectés. Dans cet ensemble, on associe à chaque procédé un composant élémentaire du système représenté par un modèle bond graph. Soit E_{Σ} l'environnement du système, i.e. l'ensemble des entités qui le composent, à savoir, la matière (E^m), l'énergie (E^e) et l'information (E^i). Un procédé est alors considéré comme étant l'ensemble des changements affectant une ou plusieurs entités du système. Pour cela, considérons le tétraèdre d'état, figure 1.1(a), proposé initialement par Paynter [2] puis repris par Thoma [3]. Le tétraèdre d'état consiste en quatre variables généralisées: l'effort (e), le flux (f), le déplacement (q) et le moment (p); et cinq relations de contraintes entre les variables généralisées. Ces dernières permettent d'obtenir (par intégration du flux et de l'effort) les variables de déplacement et de moment généralisés. L'élément C introduit une relation fonctionnelle entre le déplacement et l'effort, l'élément I entre le moment et le flux; et enfin l'élément R entre l'effort et le flux. Ainsi, le tétraèdre d'état est constitué de quatre variables généralisées et cinq relations, à savoir trois relations constitutives ($\Phi_R(e,f,R)$, $\Phi_C(e,q,C)$ et $\Phi_I(e,p,I)$, relatives aux éléments R , C et I), et deux relations structurelles correspondant aux opérateurs de type intégrale et dérivée. Sur le tétraèdre, le produit de l'effort et du flux donne lieu à une puissance. Le produit du déplacement et de l'effort fournit une énergie potentielle tandis que le produit du moment et du flux correspond à une énergie cinétique. A partir du tétraèdre d'état, le système de la figure 1.1(b) représentant la transformation de la puissance et de

l'énergie peut être obtenu.

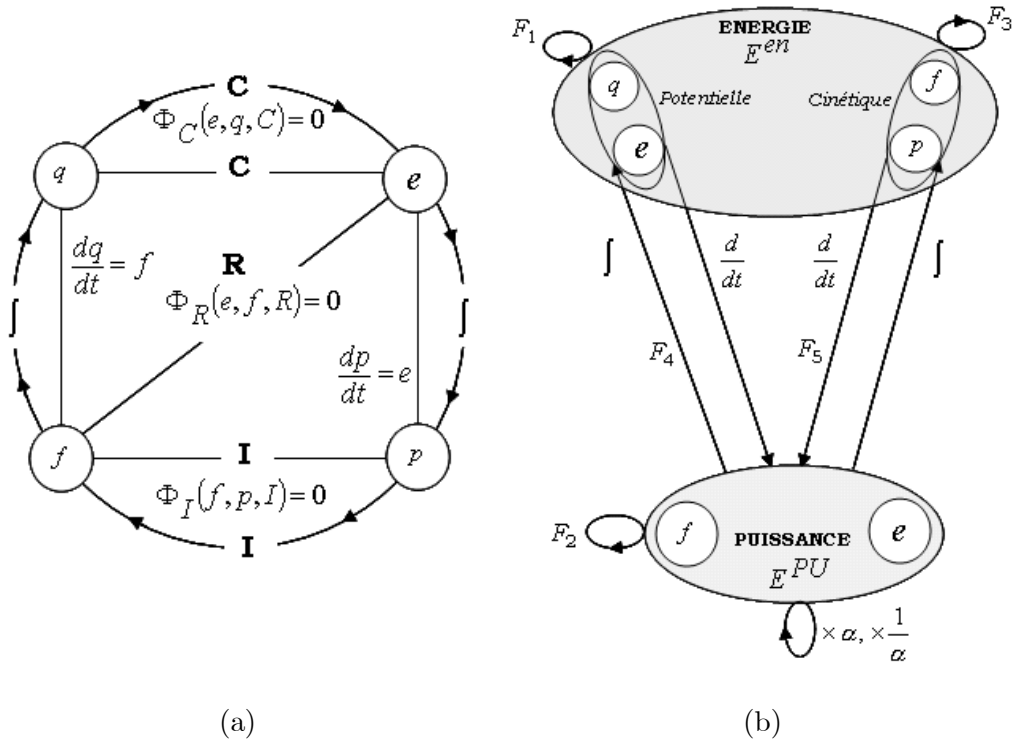


FIG. 1.1 – (a) *Tétraèdre d'état* (b) *transformation de l'énergie et de la puissance*

Sur le graphe de la figure 1.1(b), trois transitions, correspondant aux relations structurelles du tétraèdre, sont associées à trois procédés: la transition F_4 représente l'accumulation de la puissance pour produire, après intégration, une énergie (Stockage: **Procédé A**), F_5 représente la production de la puissance à partir de la dérivée de l'énergie (Transport: **Procédé T**); et $(\times \alpha, \times \frac{1}{\alpha})$ montre la transformation (ou conversion) de la puissance (Transformation: **Procédé C**) en utilisant des relations de contraintes algébriques. Par conséquent, trois classes de procédés de type **A**, **T** et **C** peuvent être définies pour représenter la partie énergétique d'un système. Les trois autres relations F_1 , F_2 et F_3 , correspondant respectivement aux équations constitutives $\Phi_{\mathbf{R}}(e, f)$, $\Phi_{\mathbf{C}}(e, q)$ et $\Phi_{\mathbf{I}}(f, p)$, mettent en évidence les contraintes entre les états du système.

Remarque 1.2.1 Dans ce mémoire le procédé de transformation **C** est noté en caractère gras pour le différencier de l'élément bond graph **C**.

- Le procédé **A** reçoit une puissance en entrée et produit une énergie (potentielle et/ou cinétique) par stockage en sortie. Ce procédé appartient à la classe de transformation de la puissance en énergie par intégration.

$$E^{Pu} \xrightarrow{f} E^{en}, (e, f) \rightarrow \{(q, e), (p, f)\}. \quad (1.1)$$

- Le procédé **T** reçoit une énergie en entrée et produit une puissance par transport (flux d'énergie) en sortie. Ce procédé appartient à la classe de transformation de l'énergie en puissance par dérivation.

$$E^{en} \xrightarrow{\frac{d}{dt}} E^{Pu}, \{(q, e), (p, f)\} \rightarrow (e, f). \quad (1.2)$$

- Le procédé **C** reçoit une puissance en entrée et produit, par conversion, transformation ou encore dissipation, une puissance. Ce procédé appartient à la classe de transformation de la puissance d'un domaine physique en un autre en utilisant des relations algébriques.

$$E^{Po} \xrightarrow{\alpha, \frac{1}{\alpha}} E^{Pu}, (e, f) \rightarrow (e, f). \quad (1.3)$$

La figure 1.2 donne les blocs diagrammes correspondant aux trois classes de procédés.

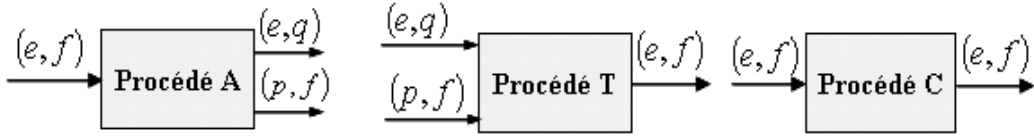


FIG. 1.2 – Blocs diagrammes des procédés **A**, **T** et **C**

1.2.2 Classes de procédés thermofluides

Du point de vue fonctionnel, les procédés de type **A** peuvent être des réservoirs, des chaudières, etc., qui correspondent à des composants de stockage. En bond graph, ces éléments sont associés à un multiport capacitif C . Ainsi, le modèle dynamique du procédé **A** est donné par la loi constitutive de l'élément multiport bond graph C qui lie les variables d'effort et de déplacement.

Les composants résistifs (ou dissipatifs) et de transformation associés au procédé de type **C** sont les pompes, les moteurs, les turbines, les transformateurs électriques, etc., qui transforme une puissance d'un domaine physique à un autre. Dans les procédés de transport **T** la force qui provoque le déplacement de la matière est due à une différence d'effort (gradient d'effort, différence de pression, de tension, de potentiel chimique, de température, etc.). Le procédé de transport **T** reçoit en entrée l'énergie associée aux paires (e, q) ou (f, p) et produit en sortie une puissance. Puisqu'on considère seulement les procédés de transport réels (avec dissipation d'énergie), un procédé de type **T** est toujours groupé avec un procédé de type **C** pour former

un procédé de type **TC**. Les procédés de transformation et résistifs **TC** sont décrits par des relations algébriques reliant les variables d'effort et de flux.

En se basant sur les fonctions réalisées par les composants des trois classes sus-mentionnées, une classification non-exhaustive des procédés thermofluides est donnée en figure 1.3. Les sources d'énergie thermofluides, similaires à celles données sur les bond graphs sont ajoutées aux classes des procédés thermofluides. Les sources peuvent être modulées par des signaux de commande externes. Ces sources thermofluides sont des flux hydrauliques et thermiques conjugués véhiculés par le fluide.

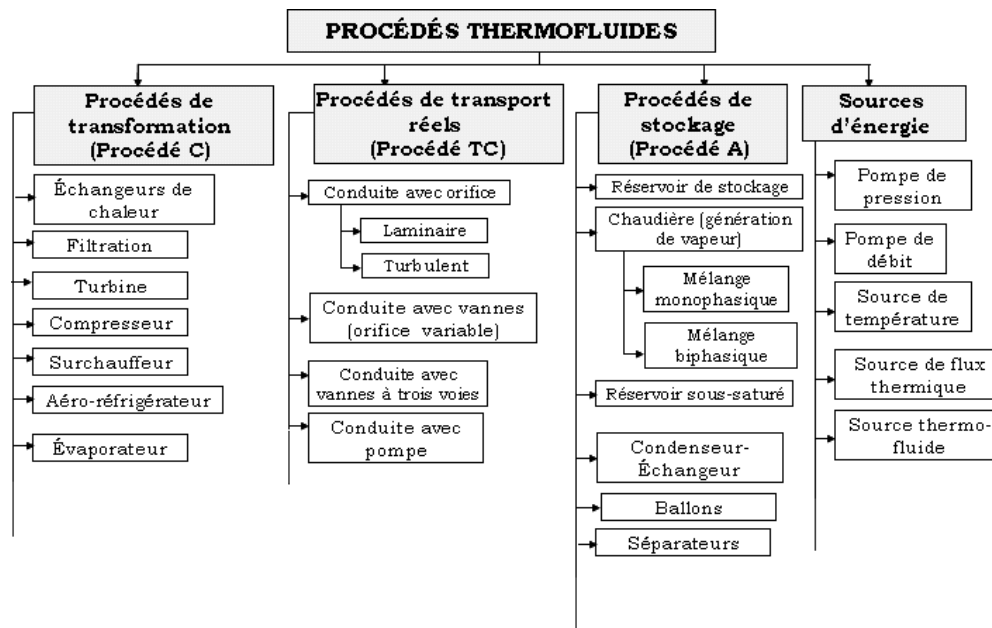


FIG. 1.3 – Différents composants des classes de procédés thermofluides

1.3 Modèles bond graphs des classes thermofluides

Les procédés thermofluides font intervenir les deux phénomènes de convection et de conduction de la chaleur. Nous définissons dans ce travail par conduction tout transfert thermique entre le fluide et la paroi et par convection l'énergie thermique véhiculée par le transport du fluide. Ce dernier phénomène est bien sûr caractérisé par le couplage de deux énergies (hydraulique et thermique); il sera donc modélisé par des bond graphs couplés multi-énergies. Par contre les phénomènes de conduction seront représentés par des bond graphs mono-énergie. Dans cette catégorie de systèmes physiques, pour modéliser par les bonds graphs le phénomène de convection thermique, il est plus pratique d'utiliser le flux d'enthalpie à la place du flux d'entropie [25, 26]. Les variables de puissance dans ce cas sont celles données par l'Eq. (1.4).

$$e = \begin{bmatrix} e_h & e_t \end{bmatrix} = \begin{bmatrix} P & T \end{bmatrix}, f = \begin{bmatrix} f_h & f_t \end{bmatrix} = \begin{bmatrix} \dot{m} & \dot{H} \end{bmatrix}. \quad (1.4)$$

Dans un procédé thermofluide, les énergies thermique et hydraulique sont couplées. Ce couplage est représenté, dans un modèle bond graph, par un lien multiple comportant un cercle comme le montre la figure 1.4(a) ou bien par deux liens parallèles [27] comme l'illustre la figure 1.4(c). L'utilisation de lignes continues parallèles pour les liens hydrauliques et thermiques (généralement en pointillés), figure 1.4(b), a été aussi proposée pour distinguer le type d'énergie mise en jeu [4]. Une autre méthode pour représenter les phénomènes de convection utilisant des *liens de convection* a été proposée dans [28]. Dans ce chapitre, les représentations de la figure 1.4(a) et figure 1.4(b) sont utilisées.

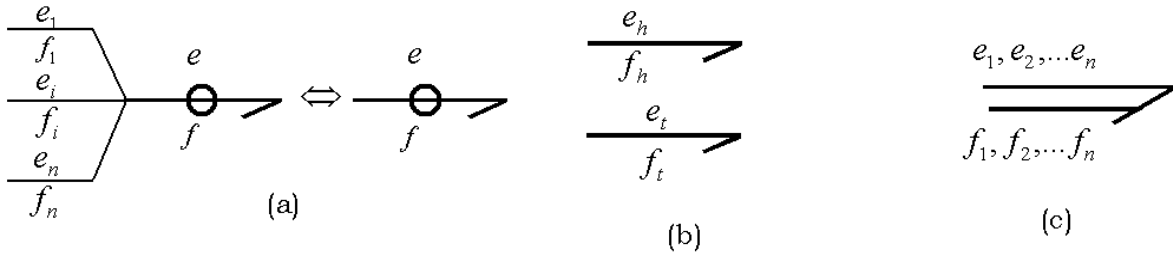


FIG. 1.4 – Différentes représentations des liens en bond graph multi-énergie

1.3.1 Sources thermofluides

Les sources thermofluides sont des flux thermiques et hydrauliques véhiculés par la convection du fluide. Elles sont caractérisées par le couplage de l'énergie thermique et hydraulique représenté par l'Eq. (1.5) dans le cas d'un fluide saturé et par l'Eq. (1.6) si le fluide est sous-saturé. Ces sources peuvent être représentées par le modèle bond graph de la figure 1.5

$$\dot{H} = \dot{m}.h, \quad (1.5)$$

$$\dot{H} = \dot{m}c_p T. \quad (1.6)$$

Cette représentation est intéressante du point de vue physique et aussi importante pour l'analyse du système. En effet, l'actionneur physique est un flux massique $Sf_h : \dot{m}$. La source de flux d'enthalpie \dot{H} est considérée comme un actionneur fictif puisqu'elle est la conséquence du flux massique, Eq.(1.6). La source de température $Se_t : T$ peut être considérée comme un paramètre ou une variable externe (si elle est donnée par une source externe).

Les sources d'énergie thermofluides et leurs modèles bonds graphs, sont ajoutées aux classes des procédés thermofluides. A noter que ces sources peuvent être modulées par un

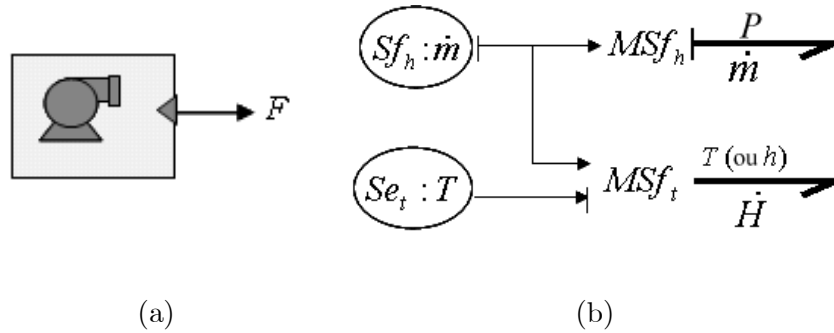


FIG. 1.5 – (a) Modèle iconique (b) modèle bond graph d'une source thermo-fluide

signal de commande externe. Le modèle bond graph de la figure 1.5(b) peut par exemple être associé à une source de pression ou de débit massique fournie par une simple pompe qui n'est pas connectée à une conduite, ce qui permet de négliger la réaction du système en aval de la pompe.

1.3.2 Le multiport thermofluide R

Dans les procédés thermodynamiques, le phénomène de dissipation (évaporation, condensation de la vapeur, convection le long d'une conduite, etc.) est modélisé par un élément multiport R . Pour les composants thermofluides associés aux procédés de dissipation, la force qui provoque la dissipation de l'énergie ou le déplacement de la matière est due à la différence entre les efforts en amont et en aval du composant (différence de pression, de potentiel chimique, de température, etc.). En modélisation bond graph, et concernant la causalité, ce composant thermofluide reçoit en entrée l'effort et retourne en sortie le flux. Mais en pratique, la causalité du multiport thermofluide R n'est pas fixe. Elle dépend des entrées et sorties imposées par les composants qui lui sont connectés en amont et en aval et par conséquent, le multiport R peut avoir une causalité conductance ou résistance.

1.3.2.1 Conduite avec restriction

D'après la causalité résistance affectée à l'élément R de la figure 1.6(a), les sorties obtenues sont le flux thermique \dot{H} et le débit massique \dot{m} . Les suffixes 'in' et 'out' représentent respectivement l'entrée et la sortie physique du composant modélisé. Les flux, thermique et massique, retournés par l'élément R , sont donnés par l'Eq.(1.7). A noter qu'en appliquant le principe de la continuité, on a $F_{in} = F_{out}$.

$$F = \begin{bmatrix} \dot{m} \\ \dot{H} \end{bmatrix} = \begin{bmatrix} \Phi_{RH}(T_{in}, P_{in}, P_{out}) \\ \Phi_{RT}(T_{in}, P_{in}, P_{out}) \end{bmatrix}, \quad (1.7)$$

où Φ_{RH} et Φ_{RT} sont deux fonctions nonlinéaires qui dépendent principalement de l'état thermodynamique du fluide et des hypothèses de modélisation prises en compte.

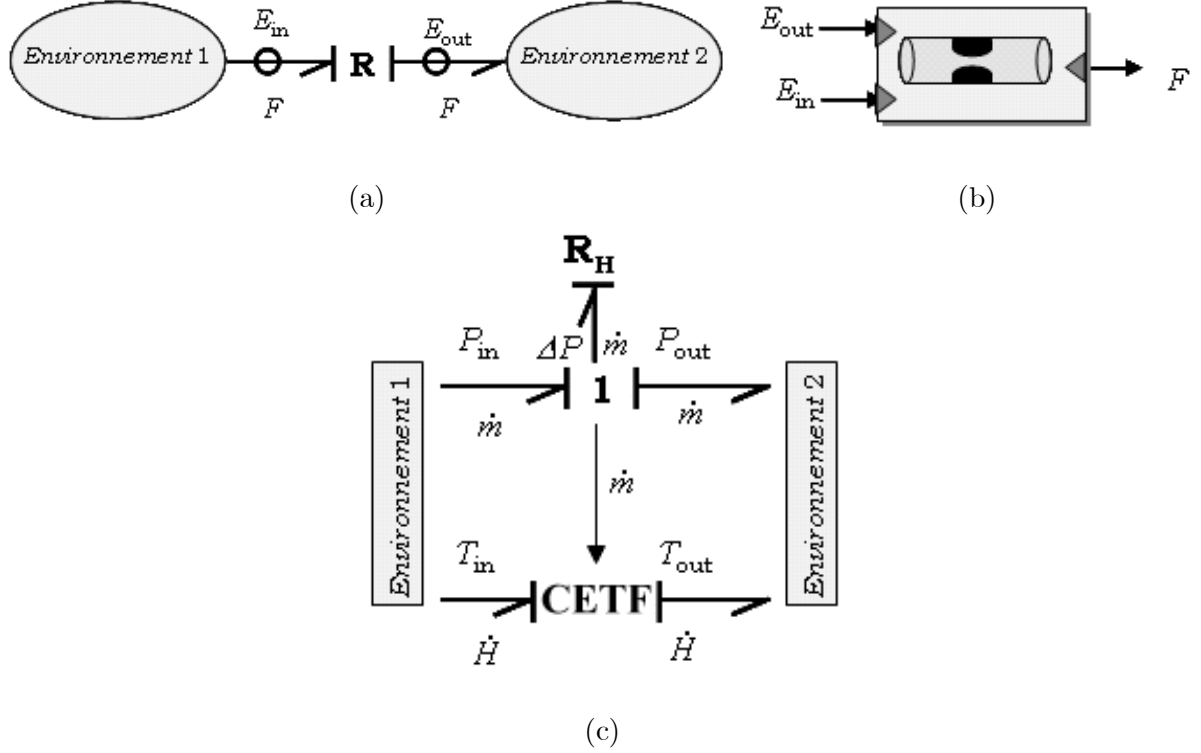


FIG. 1.6 – Modèle bond graph pour (a) flux supersonique et (c) subsonique d'un composant conduite (b)

La forme générique du modèle bond graph de la figure 1.6(a) est utile dans le cas où la restriction peut augmenter à cause d'un débit supersonique. Dans ce cas, le débit massique, \dot{m} , dépend non seulement de la différence de pression mais aussi du rapport entre la pression d'entrée P_{in} et la pression de sortie P_{out} . Il est alors possible de prédire, sous l'hypothèse d'un fluide isentropique [4], le débit en utilisant les mesures des pressions en amont et en aval (Eq. (1.8)).

$$\dot{m} = A \frac{P_{in}}{\sqrt{T_{in}}} \sqrt{\frac{2\gamma}{R_G(\gamma-1)}} \sqrt{P_r^{2/\gamma} - P_r^{\gamma+1/\gamma}}, \quad (1.8)$$

avec

$$P_r = \frac{P_{out}}{P_{in}}, \quad (1.9)$$

ou

$$P_r \leq P_{crit} = \left(\frac{2}{\gamma + 1} \right)^{\frac{\gamma}{\gamma - 1}}; \quad (1.10)$$

pour un débit critique (choked flow). P_r et T_{in} sont respectivement le rapport de pression et la température d'entrée, R_G est une constante du gaz, A est la section à la sortie de la restriction, $\gamma = c_p/c_v$, et c_p et c_v sont respectivement les capacités thermiques spécifiques à pression constante et à volume constant.

L'inconvénient de cette modélisation est qu'il est difficile de calculer l'effort lorsqu'on veut inverser la causalité du composant (Eq. (1.8)). Dans la pratique, on suppose un débit subsonique pour pouvoir considérer que le débit massique \dot{m} est fonction de la différence de pression. Le modèle bond graph de la figure 1.6(c) peut alors être utilisé. On suppose que les propriétés du débit du fluide sont constantes le long de la conduite et que l'accumulation de la masse et de l'énergie sont négligeables. En outre, on considère que l'écoulement du fluide est régulier et Newtonien. Le modèle bond graph est donné en figure 1.6. Sur cette figure, le couplage des deux phénomènes hydraulique et thermique, Eq. (1.5) et Eq. (1.6), est modélisé par l'élément *de couplage thermofluide CETF* [20] (en anglais Coupling Element for ThermoFluide) de la figure 1.7.

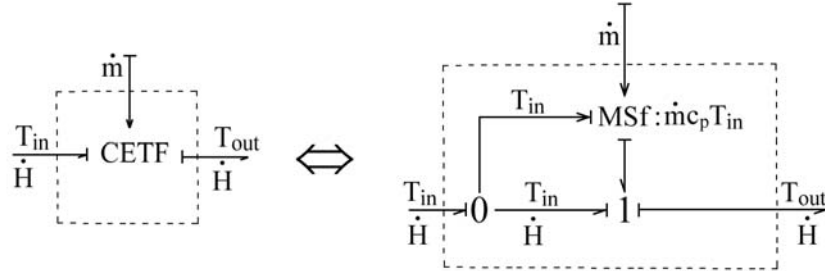


FIG. 1.7 – L'élément CETF

La forme résistance de la causalité affectée à l'élément R (figure 1.6(a)), permet de calculer, selon l'Eq. (1.11), le débit massique \dot{m} et le flux thermique \dot{H} dans la conduite dans le cas d'un fluide sous-saturé.

$$\begin{bmatrix} \dot{m} \\ \dot{H} \end{bmatrix} = \begin{bmatrix} C_d \cdot \text{sign}(P_{in} - P_{out}) \cdot \sqrt{|P_{in} - P_{out}|} \\ C_d \cdot \text{sign}(P_{in} - P_{out}) \cdot \sqrt{|P_{in} - P_{out}|} \cdot c_p \cdot T_{in} \end{bmatrix}, \quad (1.11)$$

avec

$$\frac{1}{R_H} = C_d = \sqrt{\frac{\rho \cdot D^5 \cdot \pi^2}{8 \cdot L \cdot \lambda}}, \quad (1.12)$$

pour un régime turbulent. En régime laminaire $C_d = \frac{D^2 \cdot \pi \cdot 1000}{128 \cdot L \nu}$.

C_d est le coefficient de perte de charge dans la conduite, ρ la masse volumique du fluide, ν la viscosité dynamique et enfin D et L sont respectivement le diamètre et la longueur de la conduite. Dans le cas d'un fluide saturé, la température à l'entrée de la conduite T_{in} peut être calculée en fonction de la pression en utilisant les tables thermodynamiques [29].

1.3.2.2 Conduite connectée à une pompe

Une conduite couplée à une pompe est un composant largement utilisé dans les procédés industriels et constitue en général un système d'alimentation (eau, matière première, etc.). Dans [30], une pompe est considérée comme une source d'effort qui génère une différence de pression ΔP . Pour Thoma [31], la principale action hydrostatique de la pompe est représentée par un transformateur TF (ou un transformateur modulé MTF dans le cas d'une vitesse de rotation variable) entre la puissance mécanique et la puissance hydraulique. Dans d'autres travaux, la pompe est considérée comme un élément gyrateur (GY) [6].

Sur la partie hydraulique du système de la figure 1.8, le débit massique f_C du fluide à travers la conduite, considérée comme une résistance R , est déterminé par le point d'intersection entre la caractéristique de la pompe (le débit massique f_P en fonction de la différence de pression ΔP et de la vitesse angulaire ω) et celle de la conduite (f_{Cj} , $j = 1..n$ en fonction de la valeur de la résistance de la conduite).

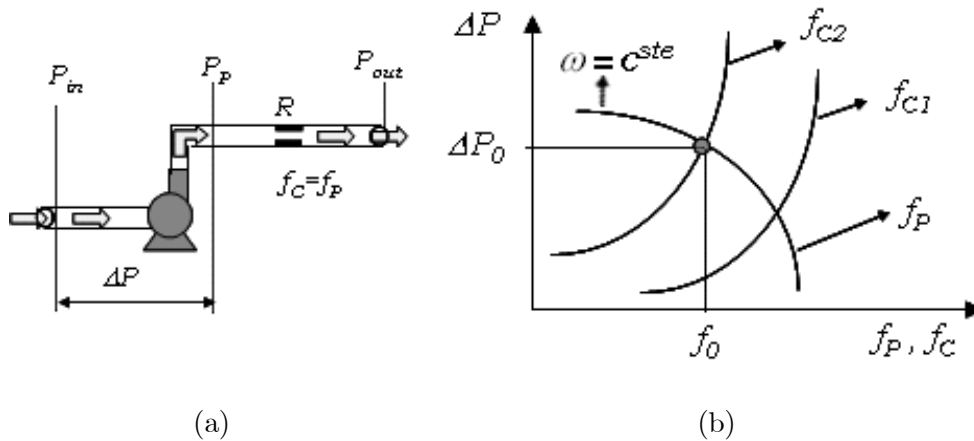


FIG. 1.8 – Détermination du débit à travers une conduite couplée à une pompe

Le modèle bond graph d'une pompe centrifuge, prenant en compte les aspects mécanique, hydraulique et thermique; est donné par la figure 1.9(b). Dans ce modèle nous n'avons pas modélisé la partie électrique de la pompe mais nous l'avons juste considérée comme une source d'effort Se . La transformation électromécanique est représentée par le gyrateur GY . I_{17} est le moment d'inertie de l'arbre en rotation et l'élément RS [17] représente ici les frottements puisqu'on considère qu'ils génèrent de la chaleur \dot{Q}_{13} , qui contribue à l'augmentation du flux

d'enthalpie \dot{H}_{16} convecté par le fluide dans la conduite. Cependant, l'évaluation de l'énergie dissipée est compliquée et dépend de plusieurs paramètres souvent non disponibles.

La transformation de l'énergie mécanique en énergie hydraulique est modélisée par un élément *MTF*. Le débit massique \dot{m}_4 délivré par la pompe est fonction de la caractéristique de la pompe:

$$\dot{m}_4 = b.f_P(\Delta P, \omega), \tag{1.13}$$

avec

$$\Delta P = P_6 - P_8. \tag{1.14}$$

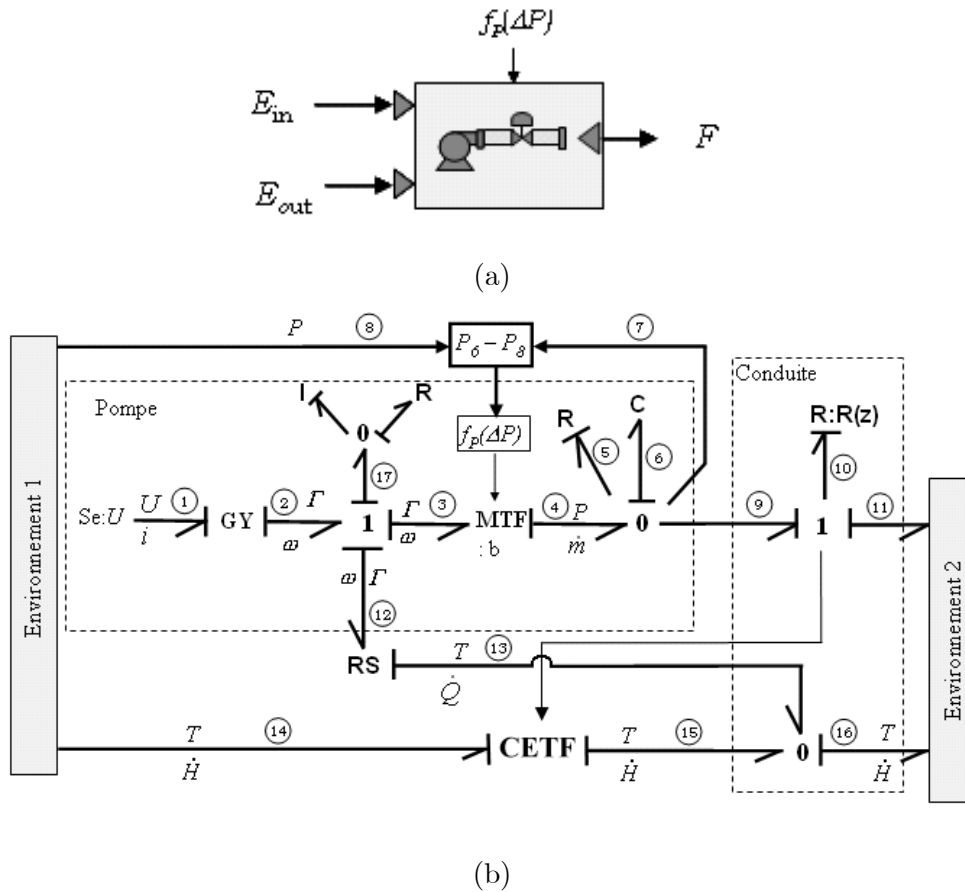


FIG. 1.9 – Modèle bond graph détaillé d'une pompe centrifuge

L'élément *MTF* est modulé par une variable booléenne b introduite par une source externe (e.g. sortie d'un régulateur). La pression P_6 , à la sortie de la pompe, est calculée par l'élément de stockage C_6 et par la relation conservative de la jonction 0 comme suit:

$$P_6 = \frac{1}{C_6} \int (\dot{m}_4 - \dot{m}_5 - \dot{m}_9) dt. \quad (1.15)$$

C_6 représente ici la compressibilité du fluide à la sortie de la pompe qui peut être prise en compte dans le modèle mais, en pratique, son calcul n'est pas trivial [6, 5]. R_5 représente les fuites dans la pompe (une partie du débit idéal s'échappe du corps de l'hélice). Le rendement total de la pompe dépend des rendements volumétrique η_{slip} (en fonction de la valeur de R_5) et mécanique η_{drag} (selon RS). Si ces deux rendements sont connus, on peut alors, à partir du modèle bond graph, déterminer la valeur de l'élément R . Les rendements η_{slip} et η_{drag} sont donnés par les relations suivantes:

$$\eta_{slip} = 1 - \frac{\dot{m}_5/\rho}{\dot{m}_9/\rho}, \quad \eta_{drag} = 1 - \frac{\Gamma_3}{\Gamma_2}. \quad (1.16)$$

Le débit massique, \dot{m}_{11} dans la conduite modélisée par l'élément $R : R(z)$ est calculé par la loi de Bernoulli suivante

$$\dot{m}_{11} = \frac{1}{R(z)} \sqrt{P_9 - P_{11}}. \quad (1.17)$$

Le coefficient de décharge, $R(z)$, est fonction des paramètres géométriques ainsi que de l'ouverture z de la conduite. Ce coefficient peut être calculé mais il est plus facile en pratique de le déterminer expérimentalement.

– Modèle thermofluide

Le flux thermique véhiculé par le débit du fluide peut être exprimé par la relation suivante:

$$\dot{H}_{16} = \dot{Q}_{13} + \dot{m}_{11} \cdot c_P \cdot T_{14}. \quad (1.18)$$

Le modèle final de la pompe qui prend en compte les phénomènes hydraulique, thermique, et mécanique doit refléter au mieux et avec une plus grande précision le composant. Cependant, l'utilisation d'un tel modèle pour la conception d'un générateur de modèles dynamiques conduit à plusieurs difficultés dues à sa complexité. En supposant un rendement idéal et en négligeant la compressibilité du fluide et la dynamique des parties mécaniques; une pompe couplée à une conduite peut être modélisée par un simple modèle bond graph avec des éléments résistifs, comme le montre la figure 1.10.

Sur le modèle bond graph de la figure 1.10(a), la pompe et la conduite sont représentées séparément par deux éléments R (R_P et $R(z)$, respectivement). La relation constitutive de l'élément R_P est donnée par la caractéristique de la pompe (Eq. (1.13)). En fait l'élément R_P est une résistance négative puisque la pompe fournit de l'énergie (voir figure 1.8). L'inconvénient de ce modèle est que le débit f_p de la pompe est obtenu en résolvant la boucle algébrique engendrée par la mise en série de deux éléments R (le chemin causal

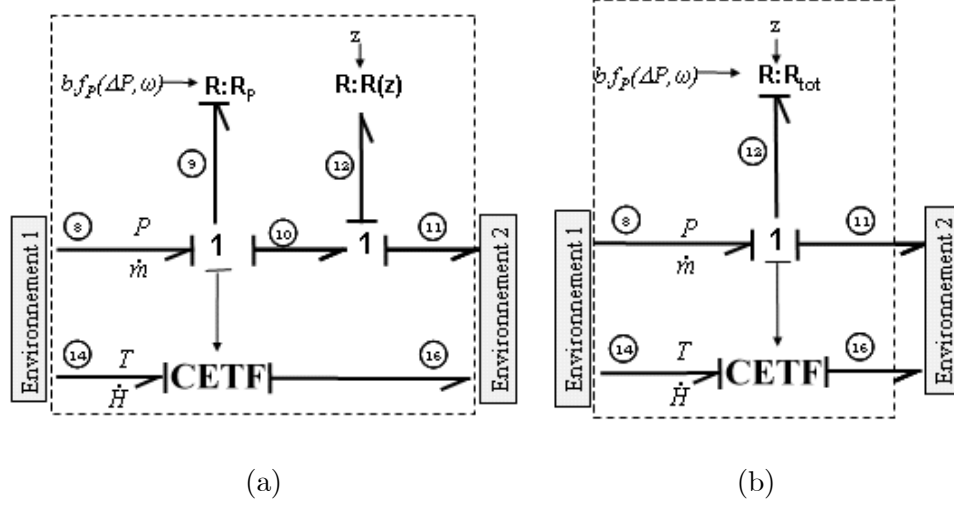


FIG. 1.10 – Modèle bond graph simplifié de la pompe couplée à une conduite

$9 \rightarrow R_P \rightarrow 9 \rightarrow 10 \rightarrow 12 \rightarrow R(z) \rightarrow 12 \rightarrow 10 \rightarrow 9$). Ainsi, les deux résistances sont groupées en une seule, R_{tot} , pour obtenir le modèle bond graph de la figure 1.10(b). Si on suppose que la caractéristique de la pompe est un premier ordre alors l'Eq. (1.13) devient, pour une vitesse angulaire constante:

$$\dot{m}_4 = f_P(\Delta P) = K_1 \Delta P + K_2 \quad (1.19)$$

où K_1 et K_2 sont deux paramètres constants (avec $K_1 < 0$).

Le débit de fluide f_0 à travers la conduite est déterminé par la solution de l'équation $\dot{m}_{11} = \dot{m}_4$ et sa valeur est donnée par la relation suivante:

$$\dot{m}_{11} = \left(\frac{R(z)^2 - R(z) \sqrt{R(z)^2 - 4K_1(K_2 + K_1(P_{11} - P_8))}}{2K_1} \right) .b. \quad (1.20)$$

1.3.3 Multiport thermofluide C

Dans le cas du multiport thermofluide C , les variables flux sont imposées (figure 1.11). La jonction vectorielle '0' représente la loi de conservation de masse et d'énergie dans le composant. F_{in} et F_{out} sont respectivement le flux d'entrée et de sortie, et n et r correspondent aux numéros respectifs des flux. Les indices h et t dans l'Eq. (1.21) renvoient respectivement aux domaines hydraulique et thermique.

$$F_{in} = \left[\sum f_{hin} \quad \sum f_{tin} \right]^T, \quad F_{out} = \left[\sum f_{hout} \quad \sum f_{tout} \right]^T. \quad (1.21)$$

Les sorties du multiport, P et T , sont calculées par l'équation constitutive qui lie les variables du pseudo-déplacement (la masse m et l'enthalpie totale H) aux variables d'effort

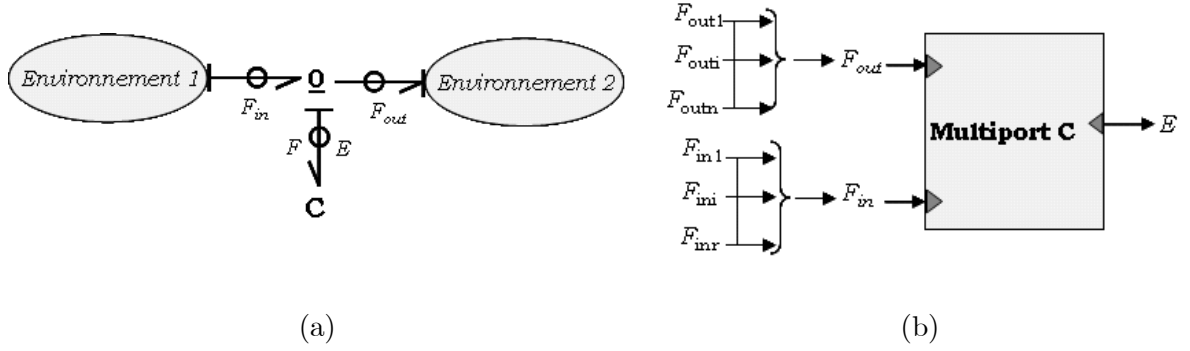


FIG. 1.11 – (a) Modèle bond graph et (b) modèle iconique du multiport C

(température T et pression P) selon la formule suivante:

$$E = \begin{bmatrix} P \\ T \end{bmatrix} = \begin{bmatrix} \Phi_{CH} (\int (F_{in}, F_{out}) dt) \\ \Phi_{CT} (\int (F_{in}, F_{out}) dt) \end{bmatrix} = \begin{bmatrix} \Psi_{ch} (m, H) \\ \Psi_{ct} (m, H) \end{bmatrix}, \quad (1.22)$$

avec E comme vecteur de variable d'effort et Φ_{CH} , Φ_{CT} , Ψ_{ch} et Ψ_{ct} sont des relations nonlinéaires qui dépendent de l'état thermodynamique du fluide (saturé, sous-saturé, monophasique, biphasique, etc.).

1.3.3.1 Accumulateur thermofluide sous-saturé

Cette sous-classe de composants est utilisée dans l'industrie afin de stocker du fluide sous-saturé. Les modèles, iconique et bond graph, de ce composant sont donnés par la figure 1.12.

Le modèle dynamique du composant de stockage de la figure 1.12(c) est généré sous forme d'équations d'état. Dans ce modèle, les variables d'état (x) sont associées à l'intégrale des variables de flux massique (\dot{m}) et thermique (\dot{H}).

$$\dot{x} = \begin{bmatrix} \dot{m} \\ \dot{H} \end{bmatrix}^T. \quad (1.23)$$

Le vecteur d'entrée, u , du composant est composé des débits massiques d'entrée (\dot{m}_{in}) et de sortie (\dot{m}_{out}) ainsi que de la température d'entrée (T_{in}) du fluide.

$$u = \begin{bmatrix} \dot{m}_{in} & \dot{m}_{out} & T_{in} \end{bmatrix}. \quad (1.24)$$

Comme il a été déjà décrit dans les travaux précédents [4, 26, 6, 32], le flux d'enthalpie n'est pas considéré ici comme une source contrôlée mais plutôt comme une source dépendant du débit massique et de la température. Le fluide sort du réservoir de stockage avec une enthalpie spécifique égale à $\frac{H}{m}$ (lien 7 sur la figure 1.12(c)). Les flux d'enthalpie d'entrée

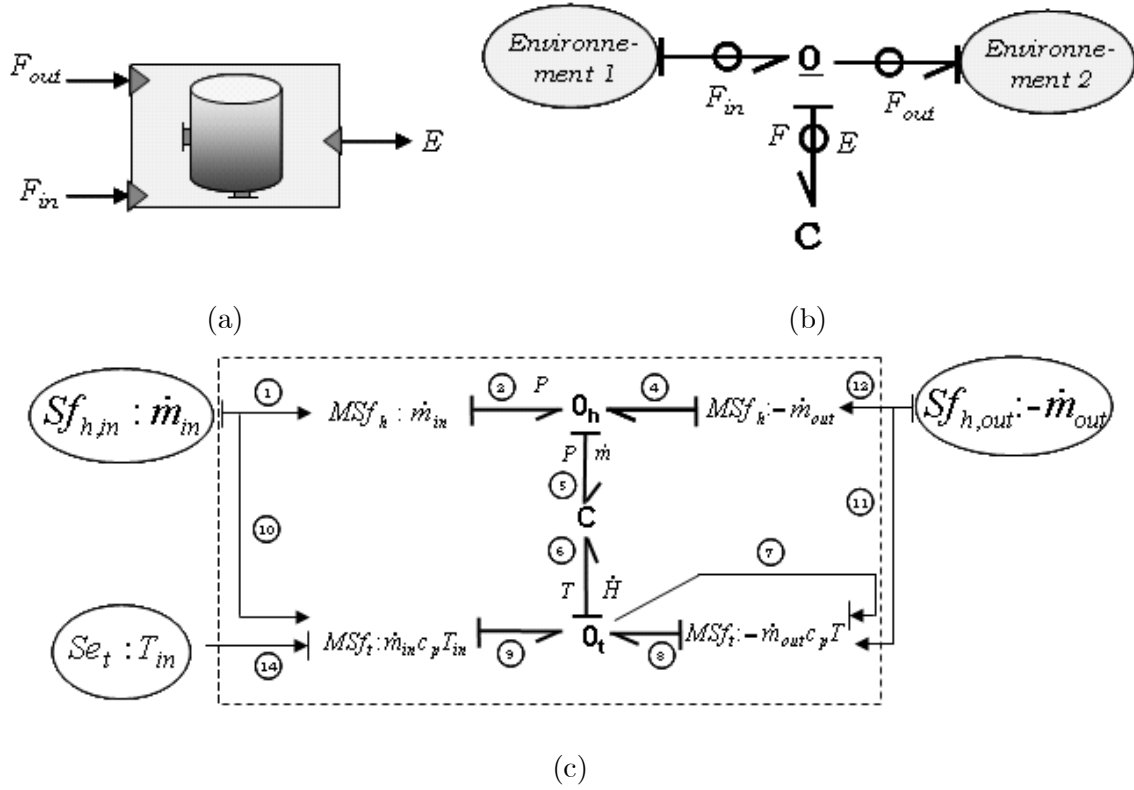


FIG. 1.12 – (a) Composant de stockage (b) modèle bond graph couplé et (c) découplé

(MSf_{tin}) et de sortie (MSf_{tout}) du réservoir de stockage sont exprimés en fonction de la température T_{in} , du débit massique \dot{m}_{in} d'entrée et de l'enthalpie spécifique.

$$MSf_{t,in} = \dot{m}_{in} \cdot c_p \cdot T_{in}, \quad MSf_{t,out} = \dot{m}_{out} \cdot \frac{H}{m}. \quad (1.25)$$

Les équations d'état sont alors déduites de façon systématique, en appliquant la procédure de génération de modèles dynamiques, à partir du modèle bond graph [4, 5, 6, 17].

$$\dot{x} = f(x,u) \Rightarrow \begin{cases} \dot{m} = \dot{m}_{in} - \dot{m}_{out} \\ \dot{H} = \dot{m}_{in} \cdot c_p \cdot T_{in} - \dot{m}_{out} \cdot \frac{H}{m} \end{cases} \quad (1.26)$$

$$y = C(x) \Rightarrow \begin{cases} P = \begin{cases} \frac{m \cdot g}{A} - \rho \cdot g \cdot z_p & \text{si } P < \frac{\rho \cdot A}{C_h} \cdot z_{\max} \\ \rho \cdot g \cdot z_{\max} & \text{si } P = \rho \cdot g \cdot z_{\max} \end{cases} \\ T = \frac{1}{c_v} \cdot \frac{H}{m} \end{cases}$$

La pression P à la position z_p ($z_p \in [0, z_{\max}]$) dans le réservoir ainsi que la température T du fluide stocké sont considérées comme les sorties de notre système. Pour un réservoir de forme

cylindrique, nous avons $C_h = \frac{A}{g}$ et z_{\max} , ρ et g sont respectivement le niveau maximum (en profondeur) du réservoir, la masse volumique du fluide et la constante universelle de gravité.

1.3.3.2 Accumulateur thermofluide saturé

Les accumulateurs de fluide saturé sont très utilisés dans l'industrie comme par exemple les chaudières ou les générateurs de vapeur. Dans la littérature on trouve plusieurs travaux qui ont été consacrés à la modélisation des générateurs de vapeur. On peut citer les travaux utilisant les équations différentielles ordinaires (EDO [33, 34, 35]) non linéaires, les équations différentielles partielles (EDP) [36, 37] ou encore la méthodologie bond graph [30, 16, 17]. Quoique les modèles à base d'EDP soient intéressants pour la conception et les simulations, ils sont toutefois difficiles à implémenter en commande et en surveillance à cause de leur complexité. Dans ce paragraphe, on considère l'accumulateur thermofluide saturé comme étant un composant connecté à d'autres composants du procédé de façon modulaire. Deux configurations de l'accumulateur sont alors possibles: un mélange liquide-vapeur mono-phasique homogène, et un mélange de deux phases séparées avec au milieu une interface. Dans ce qui suit, seule la première configuration est considérée puisqu'elle est très proche des phénomènes réels qui ont lieu dans les chaudières. Pour cela, on considère les hypothèses de modélisation suivantes:

- L'eau et la vapeur sont saturées et leurs propriétés thermodynamiques sont calculées au point d'équilibre (ceci est justifié par le fait que le mélange eau-vapeur est homogène),
- Le mélange eau-vapeur est à pression uniforme P , i.e. l'effet de la tension superficielle des bulles de vapeur est négligée,
- A cause des dimensions relativement petites de l'élément de stockage (dans ce cas la chaudière), toutes les variables ont des valeurs uniformément réparties,
- L'élément de stockage possède une capacité thermique mais subit des pertes de chaleur vers l'environnement externe,
- La vapeur produite est compressible.

Le modèle bond graph de l'élément accumulateur est donné par la figure 1.13(b). Le stockage de l'énergie hydraulique et thermique est modélisé par l'élément multiport $C : C_{ht}$. La chaleur emmagasinée par l'enveloppe métallique de la chaudière est représentée par un simple élément $C : C_m$. La chaleur transmise par conduction du fluide vers le métal et du métal vers l'environnement externe est modélisée respectivement par les éléments résistifs $R : R_{fm}$ et $R : R_{me}$. La température ambiante est représentée par une source d'effort $Se : T_{ex}$ et la puissance de la chauffe par le flux \dot{Q}_{in} .

La source de flux $Sf^* : V_b \dot{P}$ (lien 30) représente le travail mécanique de pressurisation dû aux variations de la pression P dans un volume constant, V_b , de l'accumulateur. Cette source

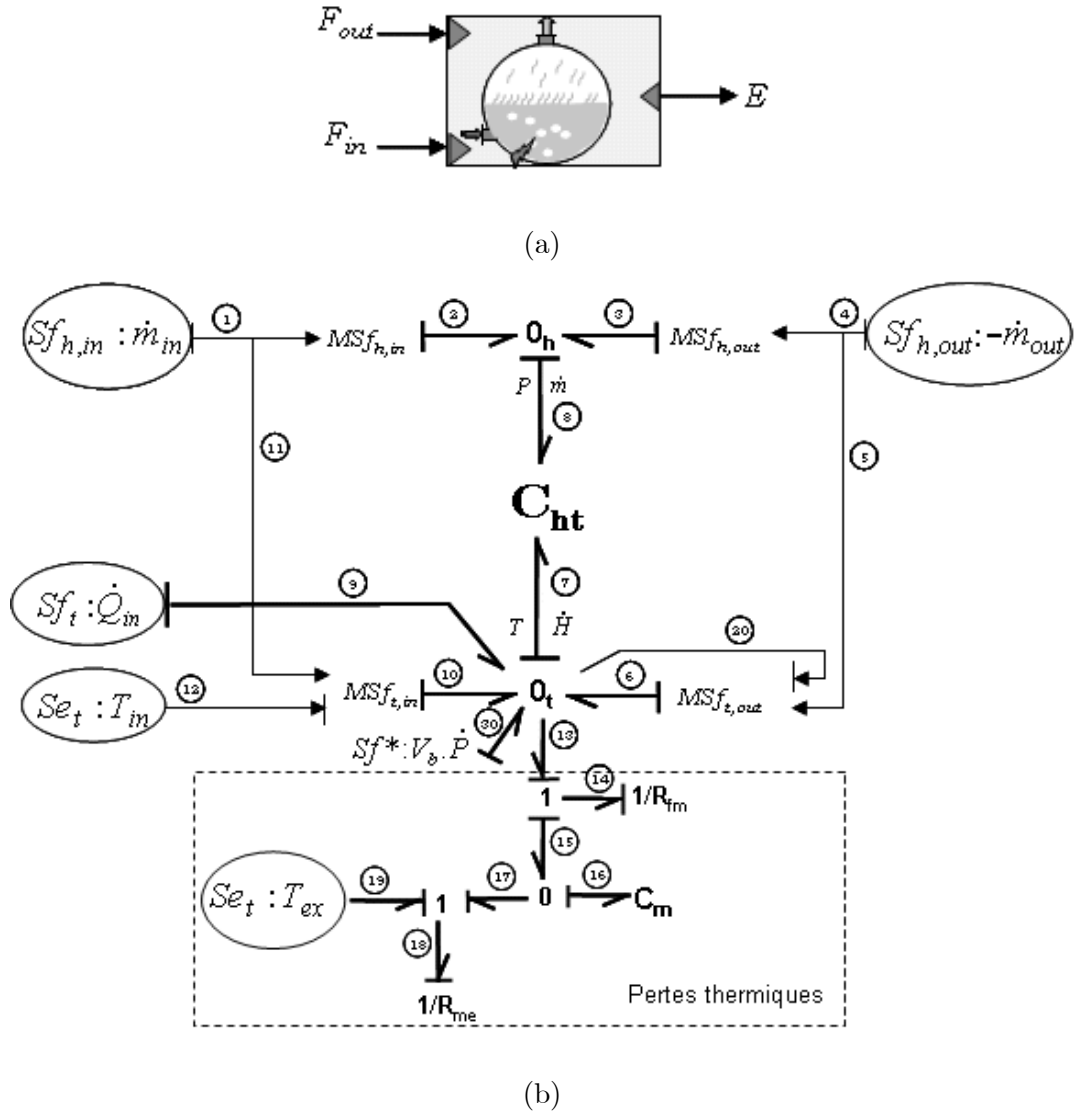


FIG. 1.13 – (a) Modèle iconique et (b) modèle bond graph d'un accumulateur d'un mélange monophasique homogène

apparaît parce que le flux d'enthalpie est utilisé à la place de la variation de l'énergie interne \dot{U} [17, 25].

Équations d'état Pour cet accumulateur, les variables d'état peuvent être choisies de différentes façons. Cependant, elles doivent avoir une interprétation physique et doivent décrire le stockage de la masse et de l'enthalpie. Deux cas sont alors possibles.

– Premier cas

Dans la modélisation bond graph, les variables d'état sont celles associées aux éléments de stockage (dans ce cas C), i.e. la masse m et l'enthalpie H stockées par le multiport $C : C_{ht}$

ainsi que la quantité de chaleur Q_m stockée par l'enveloppe métallique de l'accumulateur.

$$x = \begin{bmatrix} m & H & Q_m \end{bmatrix}^t. \quad (1.27)$$

Soit u le vecteur d'entrée de notre système:

$$u = \begin{bmatrix} \dot{m}_{in} & \dot{m}_{out} & \dot{Q}_e & T_{in} & T_{ex} \end{bmatrix}^t, \quad (1.28)$$

alors les équations d'état sous forme nonlinéaire $\dot{x} = f(x,u)$ (Eq. (1.29)) de l'accumulateur sont obtenues à partir du modèle bond graph de la figure 1.13(b).

$$\begin{cases} \dot{m} = \dot{m}_{in} - \dot{m}_{out} \\ \dot{H} = \dot{Q}_{in} + \dot{m}_{in}c_pT_{in} - \dot{m}_{out}h_v(P) + V_b\dot{P} - R_{fm} \left(T(P) - \frac{Q_m}{C_m} \right) \\ \dot{Q}_m = R_{fm} \left(T(P) - \frac{Q_m}{C_m} \right) - R_{me} \left(\frac{Q_m}{C_m} - T_{ex} \right) \end{cases} \quad (1.29)$$

La température de saturation T , du mélange peut être exprimée en fonction de la pression, $T(P)$, en utilisant des tables thermodynamiques. C_m est la capacité thermique totale de l'enveloppe métallique de l'accumulateur.

Dans le cas du mélange eau-vapeur, la température et la pression ne sont pas indépendantes. La pression P et la qualité de la vapeur X sont déterminées à partir du multipoint C après avoir résolu le système d'équations suivant (Eq. (1.30)):

$$\begin{cases} h_M = \frac{H}{m} = h_v(P)X + h_l(P)(1 - X) \\ \nu_M = \frac{V_b}{m} = \frac{X}{\rho_v(P)} + \frac{1 - X}{\rho_l(P)} \end{cases} \quad (1.30)$$

avec h_M et ν_M sont respectivement l'enthalpie spécifique et le volume spécifique du mélange. La quantité d'eau N dans les deux phases est déterminée par la relation suivante:

$$N = (1 - X).m.$$

– Deuxième cas

Les équations d'état obtenues dans le premier cas ne conviennent pas pour la commande et conduisent à des difficultés de simulation dues à la présence de la dérivée de la pression et la résolution de l'Eq. (1.30) du mélange eau-vapeur.

Puisque l'accumulateur est en équilibre thermodynamique, il convient de choisir comme variable d'état la pression à l'intérieur de l'accumulateur. En outre, cette variable est pratiquement facile à mesurer. La seconde variable d'état peut être choisie comme étant le volume total V_l de l'eau dans l'accumulateur.

$$x = \left[\begin{array}{ccc} P & V_l & Q_m \end{array} \right]^t. \quad (1.31)$$

Les bilans de masse et d'énergie exprimés respectivement à partir des jonctions 0_h et 0_t du modèle bond graph de la figure 1.13(b) sont:

$$\begin{aligned} 0_h : \frac{d}{dt} (\rho_l \cdot V_l + \rho_v \cdot V_v) &= \dot{m}_{in} - \dot{m}_{out} \\ 0_t : \frac{d}{dt} (\rho_l \cdot h_l \cdot V_l + \rho_v \cdot h_v \cdot V_v - \dot{P} \cdot V_b) &= \dot{Q}_{al} + \dot{m}_{in} \cdot c_p \cdot T_{in} - \dot{m}_{out} \cdot h_v(P) - \dot{Q}_m \end{aligned} \quad (1.32)$$

ρ_l , ρ_v , h_l et h_v sont respectivement les masses volumiques et les enthalpies spécifiques de l'eau et de la vapeur en fonction de la pression P . En outre, le volume de la vapeur, V_v , peut être exprimé en fonction du volume global de l'accumulateur, V_b , et du volume d'eau, V_l :

$$V_v = V_b - V_l. \quad (1.33)$$

En faisant les transformations suivantes:

$$\begin{aligned} A_{11} &= \rho_l - \rho_v, \quad A_{12} = V_v \cdot \frac{\partial \rho_v}{\partial P} + V_l \cdot \frac{\partial \rho_l}{\partial P}, \\ A_{21} &= \rho_l \cdot h_l - \rho_v \cdot h_v, \quad A_{22} = V_v \left(h_v \frac{\partial \rho_v}{\partial P} + \rho_v \cdot \frac{\partial h_v}{\partial P} \right) + V_l \left(h_l \frac{\partial \rho_l}{\partial P} + \rho_l \cdot \frac{\partial h_l}{\partial P} \right); \end{aligned}$$

on obtient les équations (1.34). Ce modèle est facilement utilisable et permet de calculer directement le volume d'eau ainsi que la pression à l'intérieur de l'accumulateur.

$$\left\{ \begin{array}{l} A_{11} \dot{V}_l + A_{12} \dot{P} = \dot{m}_{in} - \dot{m}_{out} \\ A_{21} \dot{V}_l + A_{22} \dot{P} = \dot{Q}_{al} + \dot{m}_{in} c_p T_{in} - \dot{m}_{out} h_v(P) - R_{fm} (T(P) - T_m) + \dot{Q}_e \\ \dot{Q}_m = R_{fm} \left(T(P) - \frac{Q_m}{C_m} \right) - R_{me} \left(\frac{Q_m}{C_m} - T_{ex} \right) \end{array} \right. \quad (1.34)$$

1.3.4 Classes détecteurs et régulateurs

La définition des classes décrites précédemment ne prend pas en compte l'instrumentation (capteurs et régulateurs) du système. Les classes capteurs et régulateurs traitent toutes les deux des signaux d'information et par conséquent, ne peuvent pas être considérées comme des composants passifs du procédé. La classe capteurs peut être divisée en deux sous-classes: détecteur d'effort (élément De sur un modèle bond graph: capteur de pression, de température, de niveau, etc.) et le détecteur de flux (élément Df sur un modèle bond graph: capteur de débit, de vitesse, de puissance, etc.). La classe régulateurs regroupe tous les types de

la régulation industrielle (PI, PD, PID, ToR, etc.) ainsi que les régulations et les lois de commande logicielles (implémentées dans des programmes informatiques).

1.3.5 Classe connecteurs

Une autre classe qu'on appellera classe connecteurs est ajoutée à l'ensemble des composants déjà définis. Les éléments de cette classe permettent la création des points de raccordement et d'interconnexion sur les conduites et entre les autres éléments du P&ID du procédé global. Les connecteurs ne stockent pas et ne transforment pas l'énergie, mais ils font simplement diverger ou converger les flux des différents éléments de transport.

1.4 Syntaxe d'interconnexion des classes de procédés

Les systèmes complexes résultent de l'interconnexion de différents composants élémentaires. L'architecture fonctionnelle de ces systèmes est construite autour de plusieurs niveaux hiérarchiques en commençant du niveau le plus bas (fonctions réalisées par les composants élémentaires) au niveau le plus haut (fonction finale du système). A chaque niveau, on associe une ou plusieurs classes fonctionnelles précédemment définies, ainsi que leurs modèles dynamiques générés en utilisant la représentation bond graph. Lorsqu'on veut interconnecter des composants appartenant à plusieurs classes différentes, des règles d'interconnexion doivent être satisfaites et vérifiées pour s'assurer que le niveau le plus haut du procédé obtenu est à la fois physiquement et fonctionnellement valide. Ces règles, sous forme d'une syntaxe, sont déduites directement du tétraèdre d'état de la figure 1.1, [2, 4], et elles ont été formulées pour la première fois par Feliot [23, 38] dans un langage formel appelé $L(G)$.

1.4.1 Règles d'interconnexion

A chaque composant du procédé on associe une classe fonctionnelle appartenant au vocabulaire $V = \{\mathbf{A}, \mathbf{T}, \mathbf{C}\}$. Une séquence de composants interconnectés (e.g. vanne, conduite, réservoir, etc.) sera représentée par une chaîne (ou un mot) w dont les caractères sont des éléments de V . En utilisant cette chaîne, Feliot [23, 38] a démontré que quatre règles suffisent pour définir l'ensemble des séquences de composants énergétiques interconnectés et dont la syntaxe est valide. Ces règles sont données par l'Eq. (1.35).

$$\begin{aligned}
R_0 &: w = C.w_C + A.w_A + T.w_T, \\
R_1 &: w_C = C.w_C + A.w_A + \varepsilon, \\
R_2 &: w_A = T.w_T + \varepsilon, \\
R_3 &: w_T = C.w_C + \varepsilon
\end{aligned}
\tag{1.35}$$

où w est une chaîne représentant un élément de V^* (un langage résultant de $V \cup \{\varepsilon\}$) dont ε est un mot de longueur zéro. w_C , w_A , et w_T représentent respectivement des chaînes de procédés de type **C**, **A** et **T**. L'interprétation des règles de l'Eq. (1.35) est la suivante:

- R_0 : une séquence de composants peut commencer par n'importe quel procédé associé aux classes déjà définies, à savoir: **A**, **T** ou **C**;
- R_1 : une succession de procédés de type **C** commence par un procédé de type **C** ou **A**;
- R_2 : une succession de procédés de type **A** commence par un procédé de type **T**;
- R_3 : une succession de procédés de type **T** commence par un procédé de type **C**.

Les règles R_0 à R_3 définissent un graphe à états finis (figure 1.14) dans lequel tout chemin, de longueur l et partant de l'état initial 0, engendre une séquence valide de l procédés.

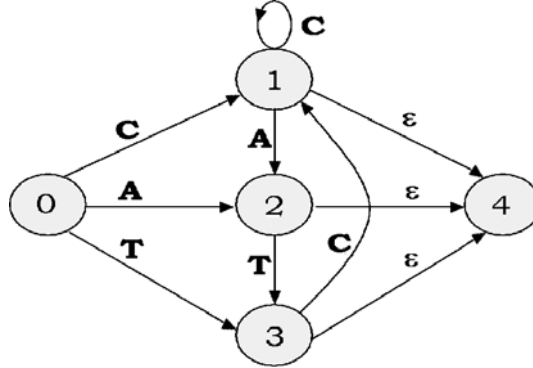


FIG. 1.14 – Un système de transition du langage $L(G)$

Pour vérifier la validité des éléments composant la chaîne d'instrumentation du système, on a ajouté d'autres règles à celles définies par Feliot. Ces règles additionnelles s'appliquent sur les classes capteurs, régulateurs et connecteurs. Elles sont définies comme suit:

- Les composants de la sous-classe détecteurs d'effort (De) peuvent être connectés seulement à des procédés de stockage de type **A**.
- Les composants de la sous-classe détecteurs de flux (Df) peuvent être connectés seulement à des procédés de transport de type **TC**.

- La sortie d’un élément de la classe détecteurs peut être connectée seulement à un composant appartenant à la classe régulateur ou à un détecteur de la même classe (des capteurs composés). Parfois, la sortie d’un élément de la classe détecteurs peut ne pas être utilisée dans une autre instrumentation.
- Dans notre travail, on suppose que les sorties des régulateurs sont soit connues soit mesurées. La sortie d’un régulateur doit être connectée à un port de modulation des composants de la classe sources (actionneurs physiques ou éléments MSe et MSf sur un modèle bond graph), de la classe transport (e.g. vanne commandée) ou de la classe transformation (e.g. moteurs ou éléments MTF et MGY sur un modèle bond graph).
- Les composants de la classe connecteurs doivent être connectés à des éléments appartenant à la classe transport (procédé de type **TC**).

1.4.2 Règles d’équivalence

Les règles d’équivalence s’appliquent seulement à des chaînes valides et permettent d’assimiler des successions de composants de classes différentes à des procédés **A**, **T**, **C** ou **TC**. En effet, une séquence (ou succession) valide de composants est équivalente à un procédé dont les entrées sont celles du premier composant de la chaîne et les sorties sont celles du dernier composant de la même chaîne. Ainsi, un modèle fonctionnel correspondant à un composant dont l’entrée est un flux et la sortie un effort est équivalent à un procédé de type **A**. A titre d’exemple, ce modèle peut être construit à partir de deux réservoirs reliés par une vanne:

$$A.TC.A \approx A.$$

En fonction des entrées et des sorties des chaînes de composants, trois règles d’équivalence ont été établies dans [23]. Elles appartiennent au langage $L(G)$ et concernent toutes les chaînes valides qu’on peut rencontrer dans un système physique donné. En outre, il a été démontré que:

1. un réseau de chaînes est d’une syntaxe valide si chacune de ses chaînes est valide,
2. un modèle fonctionnel est d’une syntaxe valide si tous ses niveaux hiérarchiques sont construits autour de réseaux valides.

A noter que les règles d’équivalence sont très utiles dans l’analyse fonctionnelle et structurelle d’un modèle en se basant sur les différents niveaux hiérarchiques (interconnexion de classes) qui le composent. Ainsi, du haut en bas, on a une forme générale des chaînes élémentaires du système et du bas vers le haut les chaînes déterminent le procédé équivalent à obtenir.

1.4.3 Exemple de chaînes

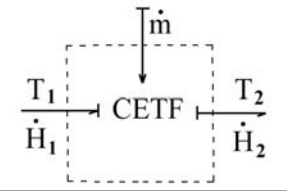
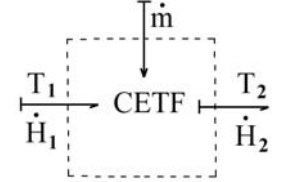
A titre d'exemple, supposons S_1 et S_2 deux sous-modèles utilisés dans un modèle global S décrit par la chaîne $S = S_1.S_2.\varepsilon$. Supposons aussi que S_2 soit un procédé de type **TC** (une vanne par exemple) et S_1 un procédé de stockage **A**. Puisque les deux éléments S_1 et S_2 sont tous les deux valides, alors le procédé global obtenu $A.TC.\varepsilon$, en connectant S_1 à S_2 , est lui aussi valide.

Par contre, si S_1 correspondait à deux éléments de stockage connectés l'un à l'autre sans un procédé de type **TC** au milieu (e.g. un réservoir connecté à un autre), il formerait alors une séquence invalide, $S_1 = A.A$. Par conséquent, le procédé final S obtenu (ainsi que le modèle fonctionnel correspondant) n'est plus valide puisqu'un de ces niveaux hiérarchiques n'a pas vérifié la syntaxe correcte définie dans les règles de validité. En effet, en l'absence de composant de transport réel entre les deux procédés de stockage, le sous-modèle S_1 doit être considéré comme un simple composant de stockage. Cependant, si on considère que $S_1 = A.TC.A + A \approx A$, (i.e. présence d'un composant de transport ou si les deux éléments de stockage sont considérés comme un seul composant), alors S_1 correspondra à une chaîne valide et par conséquent, le modèle fonctionnel global obtenu sera lui aussi valide.

1.4.4 Délimitation structurelle

La méthode formelle pour casser les liens sur un modèle bond graph global d'un système dans le but de créer des sous-modèles appelés capsules a été présentée dans [39]. Il a été montré que les composants polymorphes qui peuvent prendre plusieurs structures causales (e.g. vanne hydraulique) en fonction des variables connues ou des conditions aux limites peuvent être implémentés en utilisant des "templates" (modèles bond graphs avec différentes possibilités d'affecter la causalité) à la place de modèles dont la causalité est fixe. Dans ce cas, les templates prennent en compte les hypothèses et restrictions de modélisation (relations inversibles, singularités, boucles, etc.), les contraintes d'implémentation, et excluent les formes causales non plausibles pour élaborer le modèle du composant.

Par exemple, dans les systèmes thermofluides, on sait que lorsqu'un fluide chaud circule dans une conduite ou dans n'importe quel composant de transport, le flux d'enthalpie (\dot{H}) dépend seulement du débit massique (\dot{m}) et de la température en amont du fluide (T_{in}). Dans ce travail, le phénomène de convection thermique est représenté par l'élément multiport $CETF$ donné par la figure 1.7. Les deux formes causales possibles, ainsi que les équations correspondantes, sont montrées sur le tableau 1.1. En fait, l'information de la température de sortie du fluide n'est pas utilisée dans le modèle et ne peut pas être calculée en utilisant les valeurs connues du flux d'enthalpie et du débit massique du fluide. Par conséquent, le second port de l'élément $CETF$ est toujours en causalité conductance et il n'existe pas de chemin causal entre la température de sortie et le flux d'enthalpie.

Forme causale	Équations correspondantes
	$\dot{H}_1 = \dot{m}.c_p.T_1$ $\dot{H}_2 = \dot{m}.c_p.T_1$
	$T_1 = \frac{\dot{H}_1}{\dot{m}.c_p}$ $\dot{H}_2 = \dot{H}_1$

TAB. 1.1 – Définition de l'élément CETF

Pour créer le sous-modèle bond graph d'un composant à partir du modèle bond graph global du procédé, on détermine les limites structurelles qui doivent refléter l'appartenance physique des composants aux classes de procédés déjà définies. Comme le montre la figure 1.15, les jonctions (0, 1, TF, GY), la causalité, les liens et les différents éléments bond graphs contribuent à la détermination des composants physiques qui constituent le procédé final.

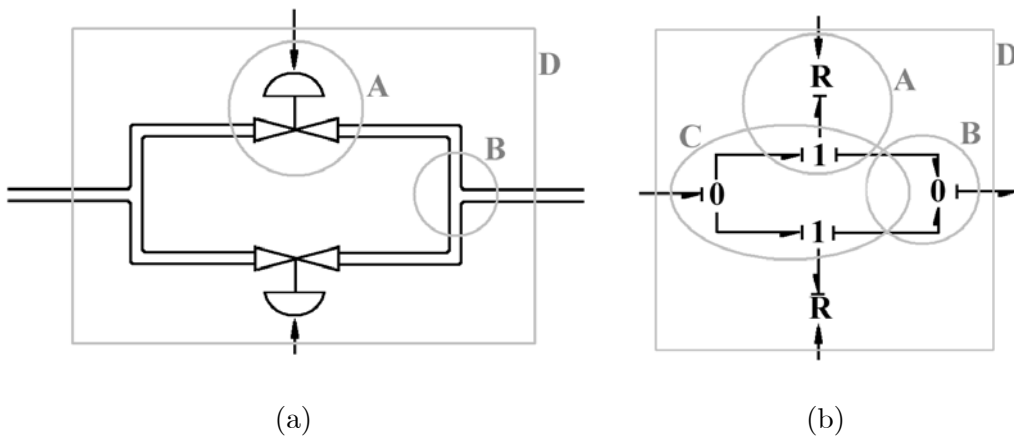


FIG. 1.15 – Conduite reliée à une vanne et le modèle bond graph correspondant

Sur le modèle bond graph de la figure 1.15(b), la limite marquée par le cercle **A** représente une vanne (avec deux entrées et une sortie) commandée par la sortie d'un régulateur. Une des entrées de la vanne est le signal de commande et la seconde entrée représente la différence de pression aux bornes de la vanne. La sortie de la vanne (causalement, on parle de la réaction) consiste en le débit qui la traverse. La relation entrée-sortie de l'élément R représentant la vanne est exprimée par la relation suivante:

$$f = \Phi(e),$$

avec Φ une fonction nonlinéaire, et e et f sont respectivement l'effort et le flux. De même, la limite marquée par le cercle **B** représente un composant de la classe connecteurs (une contrainte) utilisé pour la convergence des deux débits de fluide. Par contre, le contour structural **C** ne correspond à aucune des classes déjà définies et par conséquent ne peut pas être considéré comme un composant de notre bibliothèque. Le contour **D** représente le système entier vannes-conduites qui peut être construit en utilisant deux composants vannes et deux éléments de la classe connecteurs.

1.5 Génération du modèle global

Différents composants appartenant à différentes classes peuvent être connectés ensemble, en respectant les règles établies dans la section 1.4, pour créer un modèle fonctionnel et dynamique global valide. A noter que la validité du modèle global se vérifie en premier lieu par le langage étendu $L(G)$ décrit précédemment. Pour construire un modèle, on fusionne les liens de connexion (ou les limites de chaque composant) en vérifiant à chaque étape la compatibilité du modèle obtenu.

1.5.1 Exemple

L'exemple de la figure 1.16 illustre la procédure d'interconnexion de différents composants pour créer un modèle fonctionnel et dynamique valide. Il s'agit d'un réservoir connecté à une vanne au moyen d'une conduite.

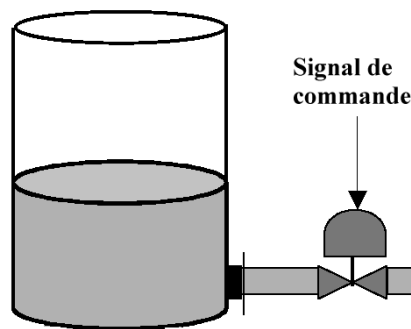


FIG. 1.16 – Un réservoir contenant un liquide sous-saturé connecté à une vanne

Le modèle pseudo-bond graph simplifié du réservoir contenant un liquide sous-saturé est donné par la figure 1.17. En réalité, un modèle complet qui tient compte de la position géométrique des ports d'entrée et de sortie, de l'évaporation, etc., est beaucoup plus complexe.

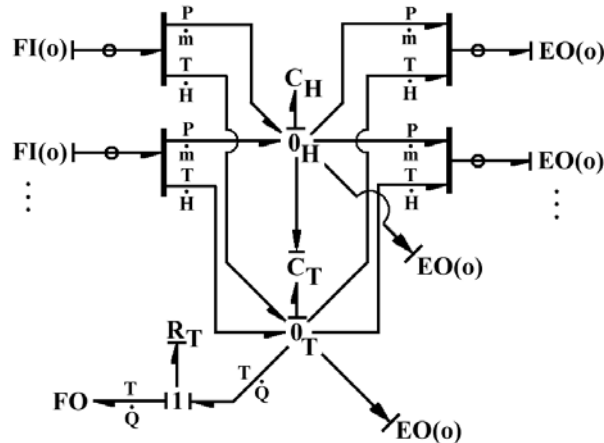


FIG. 1.17 – Modèle pseudo-bond graph d'un réservoir contenant un liquide sous-saturé

Sur le modèle pseudo-bond graph, les liens avec des cercles sont appelés des vecteurs de liens de dimension 2 qui véhiculent les deux énergies hydraulique et thermique. Les lignes épaisses verticales sont utilisées pour transformer les vecteurs de liens en liens scalaires et vice versa. Les éléments bond graphs C_H et C_T stockent respectivement l'énergie hydraulique (due à la masse du liquide) et thermique (enthalpie). Les variables d'effort pseudo-bond graphs sont la pression et la température, et les variables de flux sont le débit massique et le flux d'enthalpie.

Les expressions des capacités C_H et C_T sont données par les relations suivantes:

$$C_H = \frac{A}{g} \text{ et } C_T = m \cdot c_v,$$

avec A la section du réservoir, g la constante de gravité, m la masse totale du liquide calculée à partir de la variable d'état Q_H associée au stockage hydraulique et c_v la capacité thermique spécifique du liquide à volume constant. Les pertes de chaleur vers l'environnement extérieur sont modélisées par l'élément résistif R_T . Les éléments FI et EO sont des ports de connexion bilatéraux externes utilisés pour connecter les composants entre eux.

Contrairement aux blocs diagrammes, où les liens externes peuvent être spécifiés simplement comme des entrées ou des sorties unilatérales (ports I/O); en bond graphs il faut spécifier la nature de l'information (causalité effort ou flux). Le premier caractère du nom de port de connexion représente sa nature (E pour Effort ou F pour Flux, selon la causalité) et le deuxième caractère du nom représente le sens ou la direction de la puissance (I/O) échangée. La lettre entre parenthèses '(o)' indique que le port correspondant est optionnel, i.e. lorsque le composant est utilisé dans un modèle global, ce port peut ne pas être connecté ou utilisé. Il peut y avoir plusieurs ports d'entrée et de sortie dans un réservoir. Cette possibilité est prise en compte dans le modèle du composant. Il existe aussi deux ports d'effort en sortie

qui peuvent être optionnellement connectés pour observer la pression du liquide au fond du réservoir et sa température. Ils sont dédiés aux détecteurs d'efforts qui peuvent être connectés au réservoir afin d'observer son état. A signaler que puisque tous les ports du modèle correspondant au réservoir sont optionnels, ce modèle à lui seul peut être utilisé pour créer un procédé global valide sans aucune connexion avec d'autres composants.

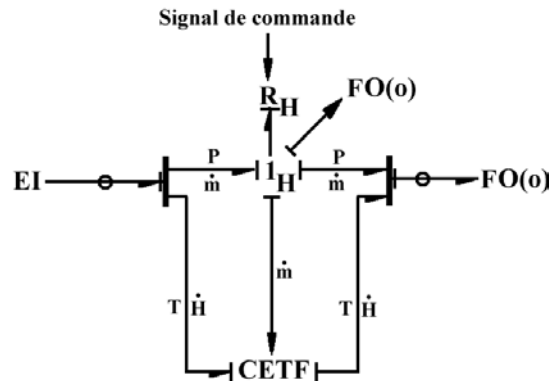


FIG. 1.18 – *Modèle pseudo-bond graph d'une conduite connectée à une vanne*

Le modèle pseudo-bond graph représentant la conduite connectée à la vanne de la figure 1.16 est montré sur la figure 1.18. Dans ce modèle, le port d'entrée n'est pas optionnel car physiquement, une conduite sans entrée n'a pas de sens. Le port de sortie, correspondant au flux, est quant à lui optionnel puisqu'il peut ne pas être utilisé si la sortie donne sur l'environnement (puisque l'on travaille avec des pressions relatives et que la température ambiante n'affecte pas les équations de l'élément *CETF*) ou encore si la conduite est totalement fermée. Le débit hydraulique à travers l'élément R_H est exprimé par une relation nonlinéaire donnée par l'Eq. (1.11). Un autre port de sortie, optionnel, permet de placer un détecteur de flux pour mesurer le débit dans la conduite et au travers de la vanne.

Lorsque les modèles bond graphs des différents composants sont disponibles, on procède à leur interconnexion pour former le modèle global du procédé. En outre, afin de garantir la validité du modèle final obtenu, les étapes suivantes sont exécutées l'une après l'autre:

- vérifier qu'à chaque composant correspond un nom unique comme identifiant.
- vérifier que tous les ports non-optionnels de chaque sous-modèle utilisé sont connectés.
- vérifier que le modèle respecte la syntaxe du langage $L(G)$. Dans cet exemple, c'est une connexion entre deux composants appartenant aux classes stockage et transport. Cela crée une chaîne $A.TC.\varepsilon$ (le chemin 0 – 2 – 3 – 1 – 4 de la figure 1.14) qui est valide par rapport à la syntaxe du langage $L(G)$.
- vérifier la compatibilité des ports de connexion. Dans ce cas, on connecte la sortie d'un réservoir (élément EO) à une entrée d'une vanne (élément EI). Pour cette connexion il

faut que la nature des ports soit la même (E pour effort) et la direction de la puissance doit être une des deux paires O-I (Output-Input) ou I-O (Input-Output).

Lorsque ces étapes sont exécutées, on peut considérer que le modèle obtenu est causalement valide et conforme à la syntaxe définie dans le langage $L(G)$. La dernière étape de la construction du modèle final consiste à ne pas considérer les ports optionnels et garder la trace de chaque composant (à travers la structure des jonctions) dans le modèle global en procédant de la façon suivante:

- créer le modèle global en y supprimant les paires de liaison EO et EI. Les ports optionnels non connectés sont soit mis à zéro soit remplacés par des sources de flux ou d'effort, selon la causalité. Puis, on supprime les ports qui peuvent l'être en utilisant les techniques standards de réduction en bond graph [5, 4, 6].
- la structure finale, regroupant les jonctions ainsi que les éléments bond graphs, constitue le modèle bond graph (ou pseudo-bond) global qui fait intervenir des liens simples (scalaires). Pour garder la trace des éléments bond graphs appartenant à un composant d'une classe donnée, une table de correspondance entre les éléments (jonctions, éléments passifs et actifs) et les identificateurs (noms) des composants est créée. Cette table est très importante et nous aidera dans l'analyse structurelle sur le modèle bond graph et dans la localisation des défaillances (là où on a besoin de remonter jusqu'au niveau composants).

En appliquant les deux dernières étapes précédentes, on obtient pour l'exemple de la figure 1.16 le modèle pseudo-bond graph global montré à la figure 1.19. Sur ce modèle, $Se : T_0$ représente la température de l'environnement extérieur.

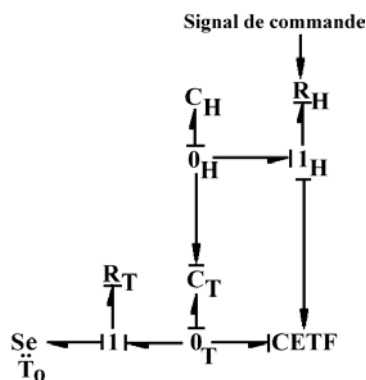


FIG. 1.19 – Le modèle pseudo-bond graph du système réservoir-vanne

1.5.2 Présence de boucles algébriques

On a vu que la syntaxe définie dans le langage $L(G)$ permet la connexion d'un procédé de type \mathbf{C} à un autre procédé du même type. Ce sont des cas qu'on rencontre souvent dans la pratique, comme par exemple des conduites connectées en série. En fait, la mise en série de plusieurs procédés de type \mathbf{C} donne lieu à des boucles algébriques entre des éléments R sur un modèle bond graph. Ces boucles apparaissent lorsque les caractéristiques individuelles des composants de type \mathbf{C} doivent être prises en compte (e.g. deux vannes en série commandées séparément), et donc doivent être représentées séparément par des éléments R sur le modèle bond graph correspondant.

Pour éviter (ou résoudre) ces boucles, différentes techniques ont été proposées dans [40]. Dans notre cas, pour respecter le langage $L(G)$, un élément de stockage (e.g. un réservoir cylindrique de section très petite) d'une capacité infiniment petite est inséré entre les composants de procédé de type \mathbf{C} pour créer une chaîne $C.A.C$ valide. Ce type d'élément est bien connu dans la communauté bond graph sous le nom d'une capacité de couplage [17] ou encore tampon (pad) [6].

En génie des procédés, la plupart des éléments résistifs sont nonlinéaires. Parfois, les boucles algébriques créées par ces éléments ne peuvent pas être résolues.

La figure 1.20 montre un exemple de système physique, une pompe reliée à une vanne, et dont le modèle bond graph correspondant (1.21) présente une boucle algébrique. Pour ce système, seul le phénomène hydraulique est modélisé puisque le phénomène thermique à travers l'élément $CETF$ ne forme pas de boucles.

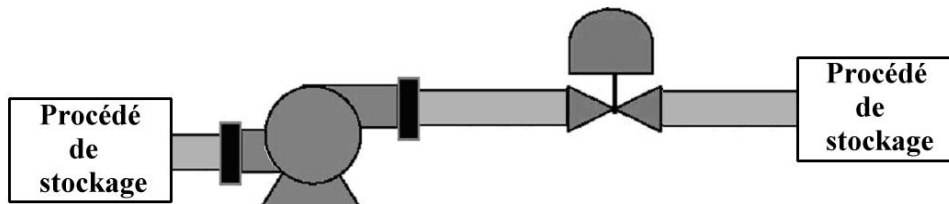


FIG. 1.20 – Interconnexion de procédés de type \mathbf{C} introduisant une boucle

Une pompe qui fait partie d'une chaîne de composants et qui n'est pas utilisée comme une source d'effort ou de flux (c'est le cas ici) est considérée comme composant appartenant à la classe de procédé transport \mathbf{C} . Si on suppose une pompe centrifuge, on peut alors la modéliser par un élément résistif R dont la valeur est négative puisque la pompe fournit dans ce cas de l'énergie. Sur la figure 1.20, la vanne aussi est modélisée par un élément R . Dans ce cas, une boucle algébrique est créée par la mise en série des deux éléments R comme le montre la figure 1.21.

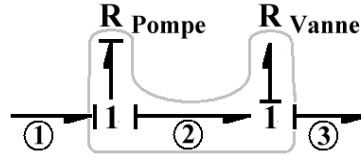


FIG. 1.21 – Boucle algébrique

La courbe caractéristique, pression-débit, des deux composants est donnée à la figure 1.8(b). Sur cette courbe, ΔP_0 est la différence de pression et f_0 le débit massique. f_p et f_c représentent respectivement le débit à travers la pompe et la vanne. La caractéristique de débit f_c change en fonction de l'ouverture de la vanne. Puisque les deux composants sont mis en série, le débit qui les traverse est le même et sa valeur est déterminée à partir du point d'intersection des deux caractéristiques. D'après l'Eq. (1.13), en considérant une caractéristique linéaire de la pompe et en prenant $b = 1$, le débit $f_p = \dot{m}_4$ peut être exprimé par la relation suivante:

$$f_p = K_1 \cdot (P_2 - P_1) + K_2, \quad (1.36)$$

avec K_1 et K_2 des paramètres connus de la pompe (K_1 étant négatif), et P_1 et P_2 sont respectivement les pressions en amont et en aval de la pompe.

Le débit f_c qui circule dans le composant conduite-vanne est:

$$f_c = C_d \cdot \sqrt{P_2 - P_3}, \quad (1.37)$$

avec C_d le coefficient de décharge de la vanne et P_3 la pression à sa sortie. Lorsque la pompe et la vanne sont connectées ensemble, on a $f_p = f_c$. En éliminant la variable P_2 des équations (1.36) et (1.37), on peut exprimer le débit de sortie f_p par l'équation suivante:

$$f_p = \frac{C_d^2 - C_d \cdot \sqrt{C_d^2 - 4 \cdot K_1 \cdot (K_2 + K_1 \cdot \Delta P)}}{2 \cdot K_1}, \quad (1.38)$$

avec $\Delta P = P_3 - P_1$.

L'équation (1.38) est une relation non linéaire qui exprime le débit en fonction de la différence de pression ΔP . Cette relation correspond donc à celle d'un élément bond graph R .

A noter que dans la plupart des cas l'obtention de la valeur de f_p sous la forme de l'Eq. (1.38) n'est pas toujours possible. Dans ce cas, deux éléments R doivent être utilisés séparément pour représenter les deux composants physiques. Parfois, il est préférable d'utiliser un élément multiport R , comme sur la figure 1.22, pour représenter la résistance équivalente des deux éléments et qui indique que le débit qui les traverse est le même. L'élément R est ici

en causalité conductance et le nouveau composant thermofluide obtenu par l'interconnexion de la pompe et de la vanne appartient à la classe des procédés de transport.

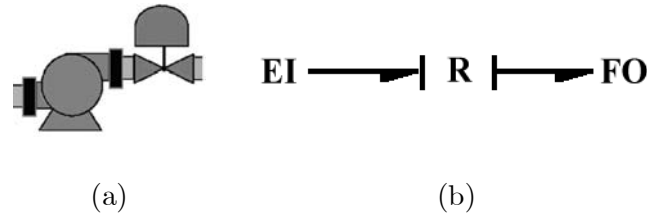


FIG. 1.22 – Pompe connectée à une conduite avec vanne (a) icône du composant (b) modèle bond graph correspondant

1.6 Contribution au placement de capteurs

On a vu précédemment qu'on peut toujours établir une table de correspondances entre les jonctions du modèle bond graph global du système et les composants physiques qui le constituent. Cette correspondance est très utile pour effectuer l'analyse structurelle sur le modèle bond graph, optimiser le placement de capteurs et améliorer la surveillabilité du procédé. En effet, la table de correspondance permet d'identifier les composants dans lesquels un placement ou non de capteurs ou d'actionneurs est nécessaire. Dans cette section, on verra comment cela est possible grâce au modèle bond graph, et en utilisant les résultats des travaux issus de l'analyse structurelle [41, 8, 15], sans faire aucun calcul formel. La procédure à suivre est illustrée à travers un exemple de système de deux réservoirs, donné à la figure 1.23.

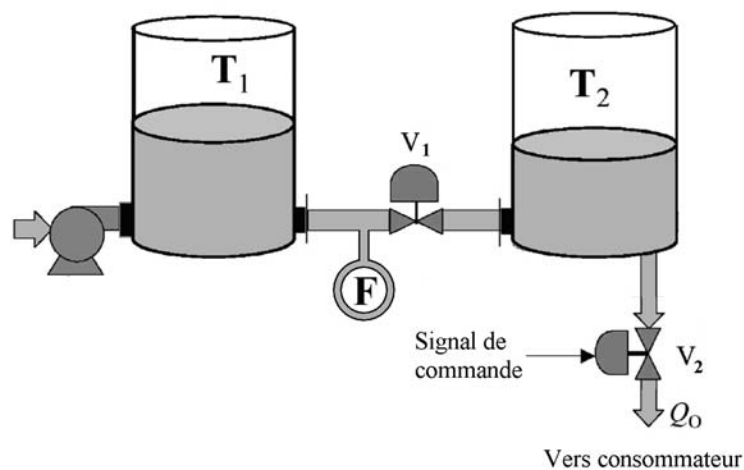
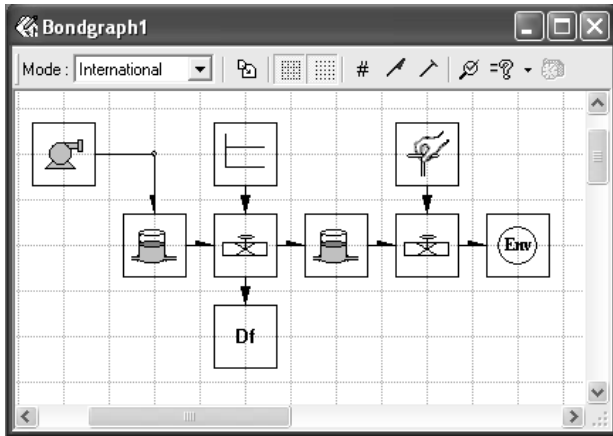
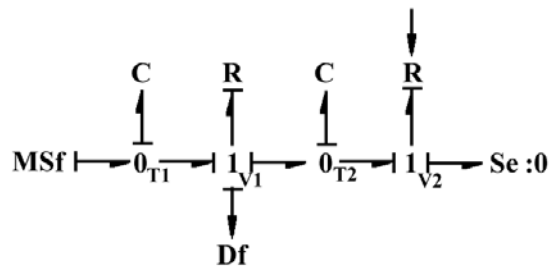


FIG. 1.23 – Système de deux réservoirs avec vanne et capteur de débit

Le système de la figure 1.23 est composé de deux réservoirs, T_1 et T_2 , et d'une vanne V_1 placée entre eux. La vanne V_2 sert à fournir de l'eau à l'utilisateur. Un capteur de débit F mesure la quantité d'eau qui circule de T_1 à T_2 . La position de la tige de la vanne V_2 est commandée par l'opérateur, tandis que la vanne V_1 est toujours ouverte à une position constante. La pompe alimentant T_1 est commandée en fonction du niveau d'eau du réservoir. Le modèle iconique du système en boucle ouverte (phénomène hydraulique seulement) est donné par la figure 1.24(a) et le modèle bond graph global correspondant en causalité intégrale préférentielle est donné par la figure 1.24(b).



(a)



(b)

FIG. 1.24 – Système de deux réservoirs: (a) modèle iconique (b) modèle bond graph

Sur le modèle bond graph de la figure 1.24(b), les suffixes (en indice) des jonctions indiquent les noms des composants physiques auxquelles elles appartiennent. Quoique la commandabilité et l'observabilité du système peuvent facilement être obtenues par l'analyse structurale sur le modèle bond graph, dans cet exemple seule l'observabilité est traitée. Selon Sueur et al.[41, 8], un système est observable si et seulement si les deux conditions suivantes sont satisfaites:

1. Tous les éléments de stockage doivent avoir un chemin causal vers au moins un détecteur (condition d'atteignabilité),
2. Tous les éléments de stockage peuvent être mis en causalité dérivée préférentielle (avec inversion de la causalité des détecteurs et/ou la dualisation des sources, si nécessaires).

Remarque 1.6.1 Dans la suite de ce mémoire, un signe “ \sim ” sera affecté à tout détecteur dont la causalité est inversée (par exemple un détecteur d'effort pour lequel la causalité est inversée sera noté \widetilde{De}).

Dans notre cas, ces deux conditions sont satisfaites, voir figure 1.25, et donc le système des deux réservoirs est structurellement observable avec l'instrumentation existant.

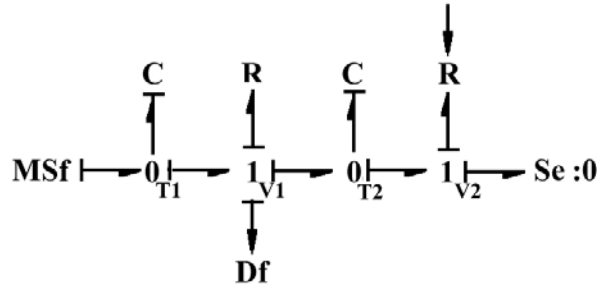


FIG. 1.25 – Le modèle bond graph global en causalité dérivée du système des deux réservoirs

Cependant, l'analyse menée n'est pas valide si l'on maintient la vanne V_2 complètement fermée. Dans ce cas, le modèle bond graph correspondant, en causalité dérivée préférentielle, est celui donné par la figure 1.26.

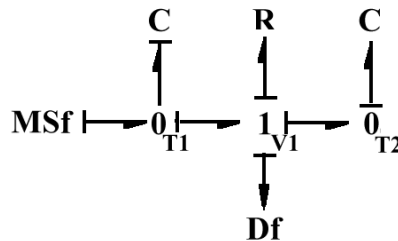


FIG. 1.26 – Modèle bond graph du système en causalité dérivée lorsque V_2 est fermée

Sur la figure 1.26, seul un des deux éléments de stockage peut être mis en causalité dérivée ce qui veut dire que le système n'est pas observable. en effet, le capteur de débit Df est utilisé pour calculer la différence de pression entre T_1 et T_2 , mais pas pour calculer les valeurs des deux pressions séparément.

De là, on voit bien que l'élément de stockage qui est resté en causalité intégrale est connecté à une jonction 0 correspondant au réservoir T_2 . On peut donc conclure qu'un détecteur supplémentaire est nécessaire dans le réservoir T_2 . Et puisque ce dernier appartient à la classe des procédés de stockage, et selon le langage $L(G)$, on a besoin d'un détecteur d'effort (De), qui peut être dans ce cas un capteur de pression ou de niveau.

En ajoutant un détecteur d'effort au système, on obtient le modèle bond graph de la figure 1.27(a) en causalité dérivée préférentielle. Sur ce modèle, il faut inverser la causalité du détecteur d'effort pour pouvoir mettre tous les éléments de stockage en causalité dérivée. En

outre, même si on supprime le capteur de débit, le modèle bond graph obtenu peut toujours être mis en causalité dérivée et satisfait la condition d'atteignabilité comme le montre la figure 1.27(b).

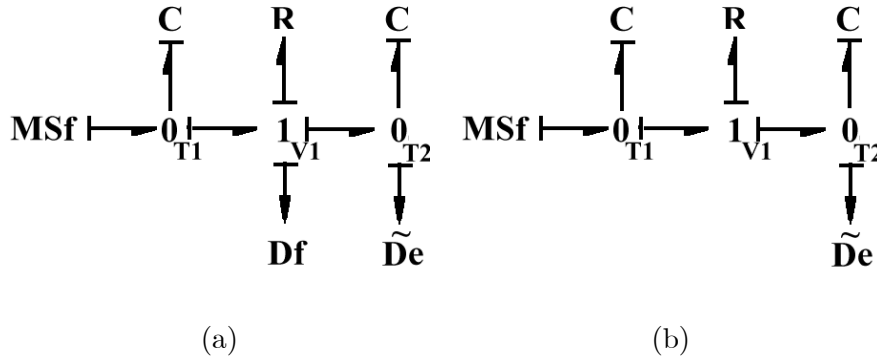


FIG. 1.27 – Modèle en causalité dérivée avec (a) deux détecteurs et (b) un détecteur

Sur la figure 1.27(a), la causalité du détecteur Df ne peut pas être inversée lorsque la causalité du détecteur d'effort De est déjà inversée (sinon on doit garder l'élément C de T_1 en causalité intégrale). On dit alors que l'un des deux détecteurs est redondant. En effet, lorsque la vanne V_2 est fermée et si les sections A_{T_1} et A_{T_2} des réservoirs sont constantes on a :

$$F = A_{T_2} \cdot \frac{dL}{dt},$$

avec L le niveau d'eau mesuré par le détecteur d'effort De . Dans ce cas, le capteur de niveau est structurellement et matériellement redondant.

Maintenant, on va essayer d'étudier le cas où seulement un détecteur de niveau De est présent sur le système. Le modèle bond graph correspondant est montré à la figure 1.28. Sur ce modèle on voit bien que le détecteur d'effort est nécessaire pour mettre tous les éléments de stockage du système en causalité dérivée et pour rendre le système observable.

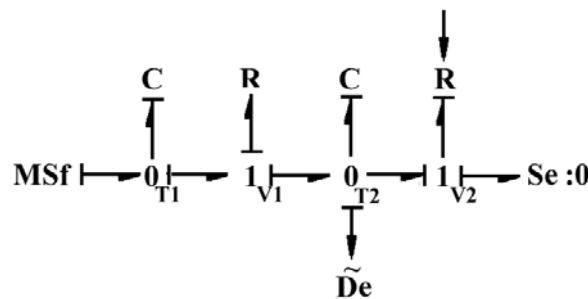


FIG. 1.28 – Le modèle du système en causalité dérivée en présence d'un seul capteur (De)

La procédure d'aide au placement de capteurs présentée dans cette section est utile dans l'analyse structurelle et pour la surveillance, et surtout simple à implémenter grâce à la modélisation hiérarchique et à la table des correspondances.

1.7 Conclusion

Une classification des composants thermofluides et une syntaxe pour leur interconnexion dans le but de créer des modèles fonctionnels et dynamiques valides a été présentée. Les modèles bond graphs des composants de chaque classe ont été générés. L'utilisateur pourra ainsi sélectionner, à partir d'une bibliothèque les composants pour construire le modèle dynamique bond graph de son système à partir d'un simple P&ID, faire de l'analyse structurelle pour la surveillance et proposer un placement de capteurs adéquat afin de satisfaire le cahier des charges. Le chapitre suivant sera consacré à la génération des indicateurs de défaillances pour la surveillance des processus thermofluides.

Chapitre 2

Génération d'algorithmes pour le diagnostic

2.1 Introduction

Ce chapitre traite de la génération d'algorithmes pour la surveillance des processus thermofluides à partir de leurs modèles bond graphs. Nous l'avons divisé en trois parties principales. Dans la première partie, nous présentons une procédure classique de génération de Relations de Redondance Analytique (RRAs) et de la matrice de signature des défaillances correspondante. Classique parce que les modèles bond graphs utilisés pour générer ces outils sont réguliers, i.e. ne présentant pas de problèmes particuliers (boucles, relations non linéaires non inversibles, etc.) pour l'élimination des variables inconnues et l'obtention de RRAs sous forme analytique symbolique. Dans le cas contraire, une nouvelle méthode est développée dans la seconde partie du chapitre pour la génération de résidus numériques. En effet, lorsque des boucles (algébriques ou différentielles) ou encore des relations non linéaires non inversibles sont présentes dans les modèles, il devient impossible de générer des RRAs sous leur forme symbolique. La méthode proposée permet, après certaines modifications adéquates apportées sur le modèle bond graph, de générer numériquement les résidus et la matrice des signatures correspondante. Enfin, la troisième partie est une extension des méthodes précédentes. Dans cette méthode les résidus sont obtenus à partir de modèles bond graphs dans lesquels les entrées sont des variables connues (mesures et sources) et les sorties sont les expressions des résidus. L'originalité de cette extension est que les résidus générés sont dépourvus de formes intégrales ou de singularités. Le travail présenté dans ce chapitre a été publié dans [42, 20, 43].

Première partie

**Génération de Relations de
Redondance Analytique**

2.2 Représentation graphique des systèmes

Avant de présenter l'algorithme de génération des RRAs nous avons voulu d'abord mettre en évidence certaines représentations graphiques des systèmes dynamiques. Pour cela, nous avons choisi de procéder par un simple exemple montré à la figure 2.1.

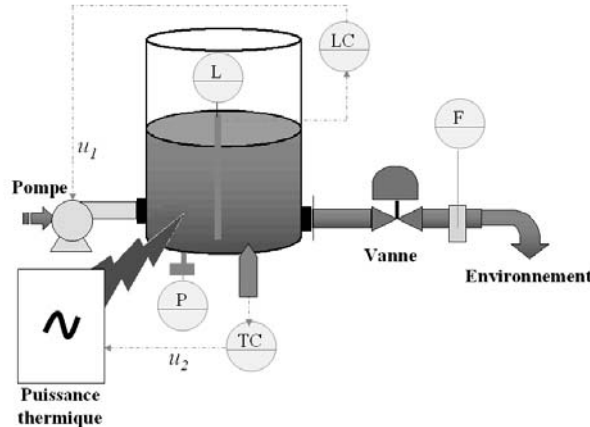


FIG. 2.1 – Réservoir alimenté en eau par une pompe

Le système consiste en un réservoir alimenté en eau par une pompe. Le régulateur de niveau (LC) agit sur la pompe pour maintenir le niveau d'eau constant dans le réservoir. L'eau dans ce dernier est chauffée par une résistance thermique et sa température est contrôlée par le régulateur (TC) en agissant sur la source thermique. L'instrumentation du processus est constituée des capteurs de niveau d'eau (L), de la température du réservoir (T) et du débit (F) au travers de la vanne manuelle.

Les capteurs de niveau et de température sont utilisés pour réguler respectivement le niveau et la température de l'eau dans le réservoir. Le capteur de débit mesure la quantité d'eau qui sort du réservoir tandis que le capteur de pression (P) représente une redondance matérielle par rapport au capteur de niveau ($L = \frac{P}{\rho \cdot g}$).

2.2.1 Représentation par un modèle structurel

Il est montré dans [44] que, du point de vue structurel, le modèle dynamique de tout système physique peut être représenté par la paire $[C, Z]$, avec $Z = \{z_1, z_2, \dots, z_n\}$ l'ensemble des variables et des paramètres du système, et $C = \{c_1, c_2, \dots, c_n\}$ l'ensemble des contraintes. La représentation structurelle exprime alors les relations qui existent entre les variables et les contraintes du processus. L'avantage de cette représentation est qu'elle permet de représenter n'importe quel système indépendamment de son modèle (qualitatif, quantitatif, statique,

dynamique, règles, tables, etc.). Ainsi, la forme analytique du modèle du système n'est pas forcément nécessaire.

En appliquant la description donnée dans [44], les contraintes correspondant à la partie hydraulique (la partie thermique sera traitée plus loin) du système de la figure 2.1 sont données au tableau 2.1.

Composant	Contrainte	Modèle
Réservoir	c_1	$\dot{M}(t) = q_i(t) - q_o(t)$
Pompe	c_2	$q_i(t) = \alpha \cdot u(t)$
Vanne	c_3	$q_o(t) = C_d \cdot \sqrt{M(t)}$
Capteur de niveau	c_4	$L(t) = \frac{M(t)}{\rho \cdot A}$
Capteur de pression	c_5	$P(t) = M(t) \cdot \frac{g}{A}$
Capteur de débit	c_6	$F(t) = q_o(t)$
Régulateur de niveau	c_7	$u(t) = \Phi_{PI}(M(t))$

TAB. 2.1 – *Modèle structurel de la partie hydraulique*

Dans le tableau 2.1, $M(t)$ représente la masse totale dans le réservoir, $q_i(t)$ le débit d'entrée, $q_o(t)$ le débit de sortie, $u(t)$ la commande du régulateur de niveau, C_d le coefficient de décharge de la vanne, A la section du réservoir et Φ_{PI} la fonction correspondant au régulateur de niveau.

En plus de ces relations, une huitième contrainte peut être ajoutée en introduisant la dérivée de la variable $M(t)$:

$$c_8 : \dot{M}(t) = \frac{dM(t)}{dt}.$$

L'ensemble des contraintes C et l'ensemble des variables Z du système physique sont donc déterminés comme suit:

$$\begin{aligned} C &= \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}, \\ Z &= \{M(t), \dot{M}(t), q_i(t), q_o(t), u(t), L(t), P(t), F(t)\}. \end{aligned} \tag{2.1}$$

A noter que le modèle structurel est utilisé pour déterminer les propriétés structurelles des systèmes (observabilité, commandabilité, surveillabilité et reconfigurabilité) indépendamment de valeurs numériques des paramètres.

2.2.2 Représentation par un digraphe

Considérons le modèle de l'exemple de la figure 2.1 sous la forme d'équations d'état suivantes:

$$\begin{aligned} \dot{x}(t) = f(x(t), u(t)) &= -C_d \cdot \sqrt{x(t)} + \alpha \cdot u(t), \\ y(t) = g(x(t), u(t)) &= \begin{pmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \end{pmatrix} = \begin{pmatrix} \frac{1}{\rho \cdot A} \cdot x(t) \\ \frac{g}{A} \cdot x(t) \\ C_d \cdot \sqrt{x(t)} \end{pmatrix}. \end{aligned} \quad (2.2)$$

Le digraphe correspondant au système d'équations (2.2) est une représentation graphique dans laquelle les sommets sont les entrées, les sorties et les états du système; et les liens entre les sommets sont définis par les règles suivantes:

- il existe un lien du sommet x_k (respectivement du sommet u_l) vers le sommet x_i si et seulement si la variable d'état x_k (respectivement la variable d'entrée u_l) apparaît dans la fonction f_i .
- il existe un lien du sommet x_k vers le sommet y_j si et seulement si la variable d'état x_k apparaît dans la fonction g_j .

Le digraphe associé aux équations (2.2) est donné à la figure 2.2.

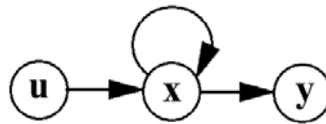


FIG. 2.2 – Le digraphe correspondant aux équations 2.2

La représentation par digraphe est une abstraction du modèle dynamique du moment que les sommets peuvent être interprétés comme étant des influences mutuelles entre les variables du système. L'inconvénient de cette forme de représentation est qu'elle ne représente pas les contraintes algébriques du modèle.

2.2.3 Représentation par un graphe biparti

A la différence du digraphe, le graphe biparti permet de représenter la structure du système en utilisant deux ensembles de sommets: les contraintes C et les variables Z et un ensemble

de liens définis par la règle suivante:

- il existe un lien entre le sommet $c_i \in C$ et le sommet $z_j \in Z$ si et seulement si la variable z_j apparaît dans la contrainte c_i (contrainte différentielle ou de mesure).

Dans ce cas, la structure, S , du modèle est une relation binaire définie comme suit:

$$S : C \times Z \rightarrow \{0, 1\}$$

$$(c_i, z_j) \rightarrow S(c_i, z_j) = \begin{cases} 1 & \text{ssi } z_j \text{ apparaît dans } c_i, \\ 0 & \text{sinon.} \end{cases} \quad (2.3)$$

La relation 2.3 conduit alors à ce qu'on appelle la matrice d'incidence du système. Celle qui correspond à notre système et qui est obtenue à partir des contraintes et des variables données par l'Eq. (2.1) est montrée au tableau 2.2.

	$M(t)$	$\dot{M}(t)$	$q_i(t)$	$q_o(t)$	$u(t)$	$L(t)$	$P(t)$	$F(t)$
c_1		1	1	1				
c_2			1		1			
c_3	1			1				
c_4	1					1		
c_5	1						1	
c_6				1				1
c_7	1				1			
c_8	×	1						

TAB. 2.2 – Matrices d'incidence du processus hydraulique

Dans le tableau 2.2 les cellules vides correspondent à la valeur 0 et le symbole \times signifie que dans la contrainte c_8 seule la forme différentielle est permise.

Les deux représentations graphiques précédentes (digraphe et graphe biparti) sont intéressantes dans le cas où les contraintes (et les variables) correspondant au processus physique sont disponibles. Dans le cas contraire, l'utilisateur doit les générer en utilisant des méthodes de modélisation. Pour le cas des systèmes simples, cette tâche est facile à réaliser. Cependant, pour les systèmes complexes (qui est souvent le cas) l'obtention de ces relations n'est pas une chose triviale. Par exemple, si on ajoute les phénomènes thermiques à la partie hydraulique

du système de la figure 2.1 on constate qu'il devient relativement difficile de générer les équations physiques du processus. C'est pourquoi la représentation bond graph en tant qu'outil de modélisation, d'analyse et de synthèse s'avère être utile pour la représentation des processus multi-énergies.

2.2.4 Représentation par bond graph

Le modèle bond graph, en causalité dérivée, correspondant à la partie hydraulique du système physique de la figure 2.1 est montré à la figure 2.3.

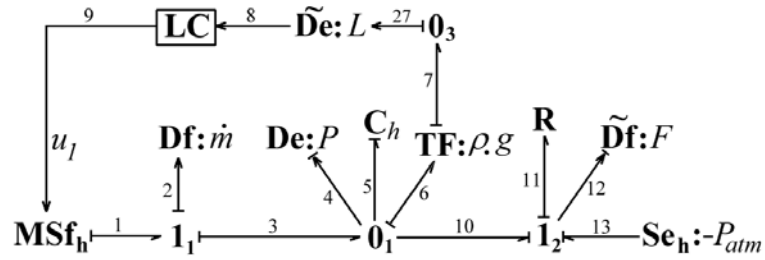


FIG. 2.3 – Modèle bond graph de l'exemple de la figure 2.1

En utilisant la représentation bond graph, l'ensemble des variables Z (connues K et inconnues X) et l'ensemble des contraintes C peuvent être déduits directement du modèle bond graph du processus.

2.2.4.1 Les contraintes

Les contraintes C peuvent être vues comme n'importe quelles relations qui relient les variables et les paramètres du système. Elles sont représentées par les contraintes structurelles (F_J), de comportement (F_B), de mesure (F_Y), du système de commande (F_C) et des sources contrôlées (F_A).

$$C = \left\{ F_J, F_B, F_Y, F_C, F_A \right\}$$

– Les équations structurelles F_J

Elles représentent l'ensemble des lois de conservation (de masse, d'énergie, etc.) et/ou des équations d'équilibre. Elles sont déduites des équations aux jonctions. Le nombre d'équations F_J est égal au nombre d'équations dans les “jonction 0” plus celles des “jonction 1” plus celles des éléments 2-ports (transformateur TF , et gyrateur GY).

$$F_J = \{F_{J0}\} \cup \{F_{J1}\} \cup \{F_{TF}\} \cup \{F_{GY}\},$$

$F_J \in R^{n_j+2*n_{2p}}$, n_j est le nombre de jonctions 0 et 1, et n_{2p} est le nombre d'éléments 2-ports (TF et GY).

Sur le modèle bond graph de la figure 2.3, les contraintes F_J sont:

$$\begin{aligned}
 F_{j_{01}} : f_6 &= f_3 - f_4 - f_5 - f_{10}, \\
 F_{j_{12}} : e_{12} &= e_{10} - e_{11} + e_{13}, \\
 F_{j_{03}} : f_{27} &= f_7, \\
 F_{TF} : e_6 &= \rho.g.e_7, \\
 F_{TF} : f_7 &= \rho.g.f_6, \\
 F_{j_{11}} : e_1 &= e_2 + e_3,
 \end{aligned} \tag{2.4}$$

– **Les équations de comportement F_B**

Les lois physiques exprimant la façon dont l'énergie est transformée représentent le modèle de comportement. Dans le modèle bond graph, elles décrivent les phénomènes physiques au niveau des éléments bond graphs (R , C , et I), et sont appelées "lois constitutives".

$$F_B = \{F_C\} \cup \{F_I\} \cup \{F_R\}, \tag{2.5}$$

$F_B \in R^{n_e}$, n_e est le nombre total des liens de puissance dans les éléments bond graph, R , C et I . Les équations de comportement du modèle bond graph de la figure 2.3 sont celles des éléments C_h et R .

$$\begin{aligned}
 f_5 &= C_h \cdot \frac{de_5}{dt}, \\
 e_{11} &= \Phi_R(f_{11}),
 \end{aligned} \tag{2.6}$$

où Φ_R est une fonction correspondant à la caractéristique de la vanne. Le paramètre C_h est donné par le relation suivante:

$$C_h = \frac{A}{g}, \tag{2.7}$$

avec A la section du réservoir et g la constante de gravité.

– **Le modèle des mesures F_Y**

Il décrit les mesures disponibles sur le modèle bond graph. Ce modèle exprime la façon dont les capteurs transforment les variables d'état du processus en signaux de sortie qui peuvent être utilisés dans l'élaboration des lois de commande et dans la détection et l'isolation des

défaillances. Sur un modèle bond graph les capteurs sont représentés par des détecteurs de flux (Df) et d'effort (De):

$$F_Y = \{F_{De}\} \cup \{F_{Df}\},$$

$F_Y \in R^{n_s}$, n_s est le nombre de détecteurs (ou capteurs). Dans notre cas, les équations de mesure sont:

$$\begin{aligned} e_5 = e_6 = \rho.g.e_7 = \rho.g.L, \\ e_4 = e_6 \Rightarrow P = \rho.g.L, \\ f_{11} = f_{12} = F, \\ f_2 = f_1. \end{aligned} \tag{2.8}$$

– **Le modèle des algorithmes de commande F_C**

Il décrit les algorithmes de commande dans lesquels les entrées des régulateurs sont les valeurs des consignes et de mesure des capteurs. Les sorties des régulateurs agissent sur les actionneurs représentés par des sources modulées d'effort (MSe) ou de flux (MSf).

$$F_C(u_ref, Y_m, \theta_{reg}) = 0, \tag{2.9}$$

où θ_{reg} est un paramètre du régulateur (gain, temps d'intégration, etc.), u_ref , et Y_m représentent respectivement la consigne et les sorties des capteurs. On a donc $F_C \in R^{n_r}$ avec n_r le nombre de régulateurs.

Contrairement aux équations structurelles et de comportement qui utilisent les variables de puissance effort-flux comme variables d'entrée-sortie, les loi de commande et de mesure utilisent les signaux d'information.

La sortie u_1 de l'algorithme de commande du régulateur de niveau d'eau utilisé dans l'exemple de la figure 2.3 est donnée par l'équation suivante:

$$u_1 = \Phi_{LC}(L_{ref}, L), \tag{2.10}$$

où L_{ref} est la consigne de niveau et Φ_{LC} la loi de commande du régulateur LC .

– **Modèle des sources contrôlées F_A**

Ces modèles décrivent les sources d'énergie contrôlées ou modulées par les signaux de commande (pompe régulée, source de tension commandée, etc.). Les signaux d'entrée u sont fournis par les régulateurs et les signaux de sortie sont les variables régulées MSe et MSf .

$$F_{A1}(MSf,u) = 0, F_{A2}(MSe,u) = 0,$$

$F_A \in R^{n_a}$ où n_a est le nombre total de sources (modulées ou non). La contrainte relative à la source modulée MSf_h de notre exemple correspond à celle donnée par l'équation suivante:

$$f_1 = \Phi_{Pompe}(u_1), \quad (2.11)$$

où Φ_{Pompe} représente la caractéristique de la pompe.

Le nombre total de contraintes sur un modèle bond graph est donc:

$$C \in R^{n_j + 2*n_{2p} + n_e + n_s + n_a + n_r}.$$

2.2.4.2 Les variables

L'ensemble des contraintes F s'appliquent à l'ensemble des variables Z : connues (K) et inconnues (X).

$$Z = X \cup K \quad (2.12)$$

Les variables inconnues X sont les variables de puissance (flux et effort) supportées par les liens de puissance du modèle bond graph. Le vecteur X contenant toutes les variables de puissance est:

$$X(t) = \{e_1(t), f_1(t)\} \cup \{e_2(t), f_2(t)\} \dots \cup \{e_{n_c}(t), f_{n_c}(t)\}, \quad (2.13)$$

avec n_c le nombre de liens de puissance. La dimension de X est donc $2 * n_c \Rightarrow X \in R^{2*n_c}$.

Les variables inconnues de notre exemple sont les variables effort et flux des liens de puissance sur le modèle bond graph de la figure 2.3.

Le sous-ensemble K des variables connues contient les valeurs des sources, les sorties des régulateurs (u), et les variables mesurées par les capteurs (Y_m).

$$K = MSe \cup MSf \cup Se \cup Sf \cup Y_m. \quad (2.14)$$

Finalement, $K \in R^l$ avec $l = n_s + n_a$. Donc $Z = l + 2 * n_c \Rightarrow Z \in R^{l+2*n_c}$.

Pour le modèle bond graph (2.3) de l'exemple d'application on a

$$K = \{MSf_h, Se : -P_{atm}, Df : \dot{m}, De : P, De : \rho.g.L, Df : F, u_1\}. \quad (2.15)$$

2.2.4.3 Les paramètres

$\theta \in R^p$ est le vecteur des paramètres. Dans le modèle bond graph les paramètres correspondent aux caractéristiques des éléments R , C et I , i.e. le coefficient de flux, la valeur de la capacité (qui peut être variable), les paramètres de supervision (tel que la consigne u_{ref}), etc..

2.3 Génération de RRAs

2.3.1 Génération de RRAs en utilisant le graphe biparti

Sur un graphe biparti, la génération de RRAs repose essentiellement sur l'élimination des variables inconnues X en utilisant les contraintes C [44]. Rappelons que l'ensemble des variables Z du système se compose de deux sous-ensembles: les variables connues K et les variables inconnues X . En outre, il a été montré dans [44] que tout système S peut être structurellement décomposé en trois sous-systèmes: un sous-système surdéterminé dont le nombre de variables Z est supérieur au nombre de contraintes C ($Z > C$), un sous-système juste-déterminé ($Z = C$) et un sous-système surdéterminé ($Z < C$).

Pour l'élimination des variables inconnues, la notion de couplage entre les variables et les contraintes est très importante. Le couplage est alors une causalité qui permet de construire la matrice d'incidence et de calculer les variables inconnues à partir des contraintes du système. Ainsi, les variables qui ne sont pas couplées ne peuvent pas être calculées et celles qui peuvent être couplées de différentes façons peuvent être calculées différemment donnant lieu à des relations de compatibilité utilisées pour la surveillance du processus. Ces relations sont alors appelées RRAs.

Une RRA est une contrainte calculée à partir d'un sous système surdéterminé et observable et exprimée en termes de variables connues du process. Elle a la forme symbolique suivante:

$$f(K) = 0. \quad (2.16)$$

L'évaluation numérique d'une RRA conduit à un résidu r :

$$r - f(K) = 0, \quad (2.17)$$

dont la valeur numérique en l'absence de défaillances doit être nulle.

Le modèle structurel de l'exemple de la figure 2.1 (tableau 2.1) présente huit contraintes et huit variables (dont cinq sont inconnues: les cinq premières colonnes du tableau 2.3). On a donc un système surdéterminé qui, en plus, ne présente ni de boucles différentielles ni de relations non inversibles. Par ailleurs, le couplage (représenté sur le tableau 2.3 par le symbole ♠) est maximal (complet) par rapport aux variables inconnues mais pas par

rapport aux contraintes. Les variables inconnues peuvent donc être éliminées (à travers les contraintes couplées) et remplacées dans celles qui ne le sont pas pour donner lieu à des relations comportant seulement des variables connues (RRAs). Pour cet exemple, trois RRAs correspondant aux trois contraintes non couplées peuvent donc être obtenues.

	$M(t)$	$\dot{M}(t)$	$q_i(t)$	$q_o(t)$	$u(t)$	$L(t)$	$P(t)$	$F(t)$
c_1		1	♠	1				
c_2			1		1			
c_3	1			♠				
c_4	♠					1		
c_5	1						1	
c_6				1				1
c_7	1				♠			
c_8	×	♠						

TAB. 2.3 – Matrice d'incidence et couplage des variables

La génération de RRAs est montrée sur le graphe biparti orienté de la figure 2.4. Sur ce graphe, les variables d'entrée $L(t)$, $P(t)$ et $F(t)$ sont des variables mesurées par les capteurs, et les cinq variables inconnues sont calculées à partir des contraintes c_1 , c_3 , c_4 , c_7 et c_8 . Les trois RRAs générées correspondent aux trois contraintes non couplées: c_2 , c_5 et c_6 (double lignes dans la figure 2.4).

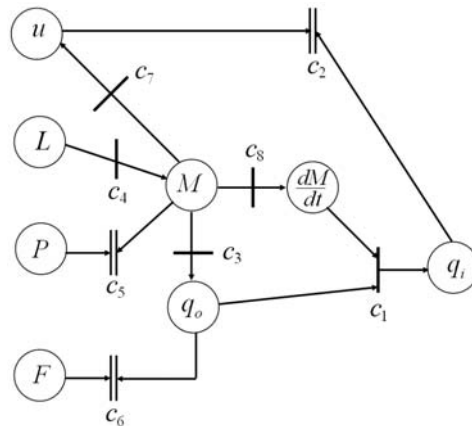


FIG. 2.4 – Génération de RRAs à partir d'un graphe biparti orienté

La génération des RRAs en utilisant le graphe biparti est relativement simple et facile pour cet exemple. Toutefois, si on analyse la façon avec laquelle les variables inconnues ont été éliminées, on s'aperçoit que pour des systèmes complexes cette méthode présente un inconvénient. En effet, en présence d'un grand nombre de contraintes, le choix du couplage permettant l'élimination des variables inconnues n'est pas évident.

2.3.2 Génération de RRAs en utilisant le bond graph

Le but derrière l'utilisation de la représentation bond graph est d'utiliser un seul outil pour la modélisation, la génération des RRAs, l'analyse structurelle et de surveillabilité et le placement de capteurs.

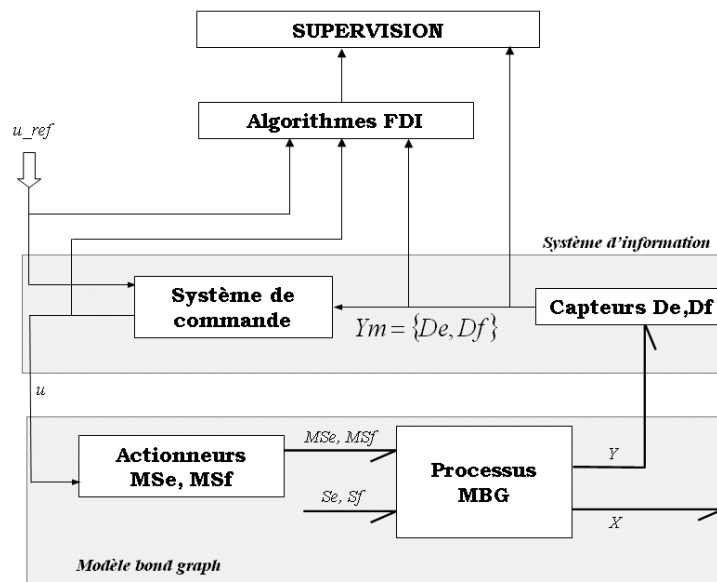


FIG. 2.5 – La représentation bond graph d'un système de surveillance

Un système de surveillance par bond graph peut être représenté par le schéma de la figure 2.5. On distingue essentiellement deux parties: le modèle bond graph et le système d'information. Le modèle bond graph représente la partie énergétique du processus physique. Il comporte le processus et l'ensemble des actionneurs. Ces derniers sont modélisés par des sources (d'effort ou de flux). Les sources peuvent être simples Se , Sf (pompe, chauffage, pression d'alimentation, etc.) ou modulées MSe , MSf (sources commandées par un signal externe fourni par un régulateur ou un utilisateur). Les capteurs et le système de commande (PID, ToR, etc.) forment le système d'information. Dans le premier système (énergétique), la puissance échangée est représentée par une demi-flèche (un lien) qui traduit les variables de puissance (l'effort et le flux). Dans le second système (système d'information) la puissance

échangée est négligeable, elle est représentée par un lien d'information.

La génération des RRAs par la méthodologie bond graph repose sur les mêmes principes que dans le graphe biparti, à savoir l'élimination des variables inconnues dans un sous-système surdéterminé et observable. Sur un modèle bond graph, l'observabilité peut être vérifiée en utilisant les propriétés structurelles développées dans [8, 41]. Les variables connues K sont celles des détecteurs et des sources et les variables inconnues X sont celles des liens de puissances dans les éléments C , I et R . L'élimination des variables inconnues est systématique sur un modèle bond graph grâce à ses propriétés causales et au parcours des chemins causaux.

Dans ce qui suit, la démarche développée consiste à générer de façon algorithmique et systématique les RRAs, les résidus correspondants et la matrice de signature des défaillances.

Dans une représentation par bond graph, la relation de l'Eq. (2.16) relative à la définition d'une RRA devient:

$$f(De, Df, Se, Sf, MSe, MSf, \theta_m) = 0, \quad (2.18)$$

avec θ_m l'ensemble des paramètres mesurés ou estimés du système.

L'algorithme que nous présentons ici permet de générer l'ensemble des RRAs possibles à partir d'un modèle bond graph d'un système donné. Le modèle est d'abord mis en causalité dérivée préférentielle (avec inversion des détecteurs si possible). Cela permet d'éviter les conditions initiales de l'état et de considérer les sources et les détecteurs comme des entrées du modèle. Les RRAs, sous forme différentielle, sont alors directement déduites des expressions des lois des jonctions 0 et 1 du modèle bond graph en causalité dérivée.

2.3.2.1 Algorithme de génération de RRAs par bond graph

L'algorithme suivant permet de générer des RRAs de façon systématique à partir d'un modèle bond graph. Voici, dans l'ordre, les étapes à suivre:

1. mettre le modèle bond graph en causalité dérivée préférentielle (en inversant la causalité des détecteurs si possible),
2. écrire les équations du modèle obtenu: de comportement F_B , des jonctions F_J , de mesure F_Y , des sources F_A , et de commande F_C ,
3. pour toute équation de jonction 0 et 1 contenant au moins un détecteur:
 - éliminer les variables inconnues en parcourant les chemins causaux sur le bond graph,
 - pour tout détecteur dont la causalité est inversée une RRA est déduite,
 - pour tout détecteur dont la causalité ne peut pas être inversée une RRA est déduite en mettant à égalité sa sortie avec la sortie d'un autre détecteur de même nature (redondance matérielle)

4. une RRA est obtenue à partir de chaque régulateur en comparant sa sortie mesurée avec la sortie prédite par son algorithme de commande,
5. refaire les étapes 3 et 4,
 - si les RRAs obtenues sont strictement différentes de celles déjà obtenues alors les garder, sinon continuer jusqu'à ce que toutes les équations des jonctions et celles des régulateurs soient explorées.

Appliquons maintenant cet algorithme sur notre exemple d'application donné à la figure 2.1. Le modèle bond graph en causalité dérivée correspondant à la partie hydraulique du système est donné à la figure 2.3. Les différentes contraintes analytiques du modèle sont données par les équations 2.4, 2.6, 2.8, 2.10 et 2.11. Par ailleurs, il est important de noter que dans cet algorithme, nous considérons que les sorties des sources et des régulateurs sont connues (mesurées).

L'étape 3 de l'algorithme consiste à éliminer les variables inconnues de chaque équation de jonction. Choisissons comme première équation celle correspondant à la jonction 0_1 . On a donc:

$$f_6 = f_3 - f_4 - f_5 - f_{10}. \quad (2.19)$$

Dans cette équation, le sous-ensemble des variables inconnues est $X = \{f_6, f_3, f_4, f_5, f_{10}\}$. Ces variables peuvent être éliminées en parcourant les chemins causaux sur le modèle bond graph de la figure 2.3. f_4 et f_6 correspondent respectivement aux variables flux des détecteurs de pression $De : P$ et de niveau $De : L$ et par conséquent, leurs valeurs sont nulles ($f_6 = f_4 = 0$). f_3 peut être déterminé à partir du chemin causal $MSf_h : \Phi_{Pump}(u_1) \rightarrow f_1 \rightarrow f_3$ (voir figure 2.3), puisque la sortie de l'actionneur est considérée dans notre cas comme étant connue ($MSf = \Phi_{Pump}(u_1)$). On peut donc écrire:

$$f_3 = \Phi_{Pump}(u_1).$$

Le chemin causal $Df : F \rightarrow f_{12} \rightarrow f_{10}$ permet de déterminer la variable f_{10} :

$$f_{10} = F,$$

tandis que la variable f_5 est calculée à partir de la relation constitutive de l'élément C_h :

$$f_5 = C_h \cdot \frac{de_5}{dt},$$

où C_h est donné par l'Eq. (2.7) et e_5 est déterminé en parcourant le chemin causal $De : L \rightarrow e_{27} \rightarrow e_7 \rightarrow e_6 \rightarrow e_5 \Rightarrow e_5 = \rho.g.L$. On a donc:

$$f_5 = \frac{A}{g} \cdot \rho \cdot g \cdot \frac{dL}{dt} = A \cdot \rho \cdot \dot{L}$$

Toutes les variables inconnues de l'Eq. (2.19) sont éliminées, la première relation de redondance analytique suivante est alors obtenue:

$$RRA_1 : \Phi_{Pump}(u_1) - \rho \cdot A \cdot \dot{L} - F = 0. \quad (2.20)$$

En procédant de la même façon que pour l'Eq. (2.19), une deuxième RRA peut être générée à partir de l'équation de la jonction 1₂. Dans cette équation, $e_{12} = 0$ puisqu'elle correspond à la variable effort d'un détecteur de flux ($Df : F$). On peut donc écrire:

$$e_{12} = e_{10} - e_{11} + e_{13} \Rightarrow e_{10} - e_{11} + e_{13} = 0, \quad (2.21)$$

le chemin causal $De : L \rightarrow e_{27} \rightarrow e_7 \rightarrow e_6 \rightarrow e_{10}$ permet de calculer e_{10} . La variable e_{13} représente la pression atmosphérique de l'environnement ambiant, tandis que e_{11} peut être obtenue à partir de la relation constitutive de l'élément R :

$$\begin{aligned} e_{10} &= \rho \cdot g \cdot L, \\ e_{13} &= -P_{atm}, \\ e_{11} &= \Phi_R(f_{11}) = \Phi_R(F). \end{aligned}$$

La deuxième RRA qui est structurellement indépendante de la première est donc

$$RRA_2 : \rho \cdot g \cdot L - \Phi_R(F) - P_{atm} = 0. \quad (2.22)$$

Les RRAs précédentes (Eq. (2.20) et Eq. (2.22)) sont toutes les deux obtenues à partir des jonctions correspondant au phénomène hydraulique du processus.

Complicons maintenant le système en prenant en compte sa partie thermique et appliquons l'algorithme précédent pour générer les RRAs correspondantes. Le modèle bond graph global, en causalité dérivée, du processus de la figure 2.1 est donné à la figure 2.6.

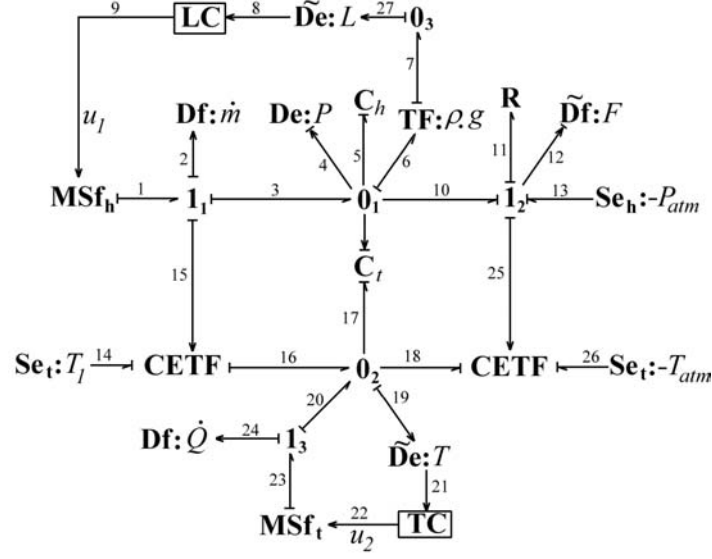


FIG. 2.6 – Modèle bond graph en causalité dérivée du système de la figure 2.1

En plus du phénomène hydraulique, le modèle bond graph représente la modélisation du phénomène thermique. Ce dernier est représenté par la source modulée MSf_t . Le couplage entre la partie hydraulique et la partie thermique est représenté par l'élément $CETF$ décrit au premier chapitre. Les contraintes supplémentaires introduites par le phénomène thermique sont systématiquement déduites du modèle bond graph. Elle sont données par les équations suivantes:

$$\begin{aligned}
 F_{j0_2} : f_{19} &= f_{16} + f_{20} - f_{17} - f_{18}, \\
 F_{j1_3} : e_{23} &= e_{20} + e_{24}. \\
 f_{17} &= \frac{d}{dt}(C_t \cdot e_{17}), \text{ avec } C_t = m \cdot c_p \text{ et } m = \int f_5 \cdot dt = C_h \cdot e_5 = \frac{A}{g} \cdot e_5, \\
 e_{17} &= e_{19} = T, \\
 f_{24} &= f_{23}. \\
 u_2 &= \Phi_{TC}(T_{ref}, T), \\
 f_{23} &= \Phi_{Chauffe}(u_2);
 \end{aligned} \tag{2.23}$$

dans lesquelles C_t est l'élément bond graph correspondant au stockage de l'énergie thermique, c_p est la capacité thermique spécifique du liquide, Φ_{TC} est la fonction relative au régulateur de la température et $\Phi_{Chauffe}$ la fonction caractéristique de la chauffe.

Le vecteur K des variables connues déduit du modèle bond graph global du système est donc:

$$K = \left\{ \begin{array}{l} MSf_h : \Phi_{Pompe}(u_1), MSf_t : \Phi_{Chauffe}(u_2), Se : -P_{atm}, Se_t : T_1, \\ Df : \dot{m}, De : P, De : \rho.g.L, Df : F, De : T, Df : \dot{Q}, u_1, u_2. \end{array} \right\}. \quad (2.24)$$

La RRA qui va suivre correspond à la partie thermique du système et est générée à partir de la jonction 0_2 . L'équation de conservation de cette jonction est donnée par la relation suivante:

$$f_{19} = f_{16} + f_{20} - f_{17} - f_{18}, \quad (2.25)$$

avec $f_{19} = 0$ puisqu'elle représente la variable flux d'un détecteur d'effort, $De : T$. Pour éliminer les variables inconnues $X = \{f_{16}, f_{20}, f_{17}, f_{18}\}$, de l'Eq. (2.25), on doit tenir compte du couplage physique entre les phénomènes hydraulique et thermique représentés par l'élément $CETF$ de la figure 1.7. Les équations de cet élément étant données dans le premier chapitre (tableau 1.1), on peut donc écrire:

$$f_{16} = f_1.c_p.T_1 = \Phi_{Pump}(u_1).c_p.T_1.$$

Dans le réservoir de stockage, la variation de la quantité de chaleur est représentée par la variable flux f_{17} . Cette variable peut être déterminée par la relation constitutive de l'élément C_t de l'Eq. (2.6) comme suit:

$$f_{17} = \frac{d}{dt}(C_t.e_{17}).$$

Le chemin causal $De : T \rightarrow e_{19} \rightarrow e_{17}$ permet de calculer la variable $e_{17} : e_{17} = T$. Le paramètre C_t étant donné par l'Eq. (2.23), on peut alors écrire:

$$f_{17} = \frac{d}{dt}(A.\rho.c_p.L.T) = A.\rho.c_p. [\dot{L}.T + L.\dot{T}].$$

D'après la loi de conservation de la jonction 1_3 , on a

$$f_{20} = f_{23} = \Phi_{Chauffe}(u_2).$$

Il reste maintenant à déterminer la variable f_{18} . Cette dernière correspond au flux d'enthalpie qui sort du réservoir et qui traverse la vanne. Ce flux peut être obtenu par la relation de l'élément $CETF$:

$$f_{18} = f_{11}.c_p.T = F.c_p.T.$$

La détermination des variables inconnues de l'Eq. (2.25) conduit à la troisième RRA suivante:

$$RRA_3 : \Phi_{Pompe}(u_1).c_p.T_1 + \Phi_{Chauffe}(u_2) - A.\rho.c_p. \left[\dot{L}.T + L.\dot{T} \right] - F.c_p.T = 0. \quad (2.26)$$

Les RRAs correspondant aux régulateurs sont directement déduites des algorithmes de commande (étape 4 de l'algorithme). Elles correspondent à la différence entre la sortie, supposée mesurée des régulateurs, et la sortie prédite fournie par l'algorithme de commande. Dans notre exemple, deux RRAs peuvent être générées:

$$RRA_4 : u_1 - \Phi_{LC}(L_{ref}, L) = 0, \quad (2.27)$$

$$RRA_5 : u_2 - \Phi_{TC}(T_{ref}, T) = 0.$$

Sur le modèle bond graph de la figure 2.1(b), la causalité du détecteur de flux $Df : \dot{m}$ (jonction 1₁) ne peut pas être inversée. En effet, ce détecteur est matériellement redondant avec la sortie du régulateur LC , comme le montre l'Eq. (2.10), et aussi avec les détecteurs $De : L$ et $Df : F$. De même, le détecteur $Df : \dot{Q}$ est matériellement redondant avec la sortie du régulateur LC et le détecteur $De : T$. Aussi, le détecteur d'effort $De : P$ est matériellement redondant avec le détecteur d'effort $De : L$. D'où les RRAs correspondantes suivantes:

$$RRA_6 : \dot{m} - \Phi_{Pompe}(u_1) = 0,$$

$$RRA_7 : \dot{Q} - \Phi_{Chauffe}(u_2) = 0, \quad (2.28)$$

$$RRA_8 : P - \rho.g.L = 0.$$

Comme résultat de l'algorithme présenté ici, pour un système disposant de n capteurs, il doit y avoir n résidus structurellement indépendants. De plus, en l'absence de défaillances la valeur de chaque résidu r_i ($i = 1 \dots n$) est égale à zéro. L'indépendance structurelle signifie que deux résidus ne doivent pas faire intervenir exactement les mêmes variables. Sur un modèle bond graph, cela se traduit par le fait que les résidus ont été obtenus par l'élimination des variables inconnues en parcourant des chemins causaux différents. Ainsi, pour l'exemple d'application, le nombre de détecteurs est égal à huit (avec les sorties des sources et des régulateurs considérées comme détecteurs) donnant lieu au même nombre de RRAs.

Proposition 1 *Dans un modèle bond graph en causalité dérivée préférentielle et observable, pour chaque jonction, 0 ou 1 avec un seul détecteur, correspond une RRA.*

Proposition 2 *Dans le cas où n détecteurs de même nature (effort ou flux) sont présents sur une seule jonction, $n - 1$ RRAs directes sont obtenues en mettant simplement à égalité les valeurs fournies par chaque détecteur.*

Proposition 3 *Dans un modèle bond graph totalement observable, le nombre de RRAs est égal au nombre de détecteurs. En outre, les RRAs obtenus sont structurellement indépendantes.*

2.3.2.2 Matrice de signature des défaillances

La structure des RRAs forme une matrice binaire S_{ji} qui nous renseigne sur la sensibilité des résidus aux défaillances des composants du processus physique (capteurs, actionneurs, régulateurs, éléments physiques). Les éléments de la matrice sont définis comme suit:

$$S_{ji} = \begin{cases} 1, & \text{si la } i^{\text{ème}} \text{ RRA contient des variables du } j^{\text{ème}} \text{ composant} \\ 0, & \text{sinon.} \end{cases} \quad (2.29)$$

La matrice S_{ji} est appelée la matrice de signature des défaillances qui fournit la logique pour la localisation des défaillances détectées durant le fonctionnement du système. L'objectif de la procédure de localisation (isolation) est de fournir à l'opérateur la liste des composants défaillants. Chaque composant a une signature représentée par un vecteur ligne de la matrice et qui ne prend pas en compte les valeurs affectées aux variables D_b (détectabilité) et I_b (isolabilité). Une défaillance d'un composant est détectable si la variable associée au composant est présente dans au moins une RRA (ou résidu r correspondant). Cette défaillance est localisable si et seulement si sa signature est unique, i.e. différente des signatures des autres composants.

A partir du cahier des charges demandé, on peut fixer la liste des composants à surveiller. Il est évident que les performances du système de surveillance (concernant l'isolabilité) dépend du nombre d'éléments à surveiller et des différences de leurs signatures de défaillance. Pour construire la matrice des signatures, certains composants sont supprimés de la matrice dès lors qu'ils sont considérés comme infaillibles. Dans cette application, on suppose par exemple que la température de l'environnement n'est jamais défaillante et que les sorties des régulateurs sont toujours correctement mesurées. La matrice de signature des défaillances correspondant à l'exemple de la figure 2.1 est donnée au tableau 2.4.

En analysant cette matrice, on voit bien que la variable associée à chaque composant est présente dans au moins un résidu. En plus, la signature de chaque composant est unique (différente des autres). Donc, toutes les défaillances du processus sont théoriquement détectables ($D_b = 1$) et isolables ($I_b = 1$). La non-détectabilité et la non-isolabilité, des défaillances d'un composant, correspondent respectivement à $D_b = 0$ et $I_b = 0$.

	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	D_b	I_b
$De : P$	0	0	0	0	0	0	0	1	1	1
$De : L$	1	1	1	0	0	0	0	1	1	1
$De : T$	0	0	1	0	0	0	0	0	1	1
$Df : \dot{m}$	0	0	0	0	0	1	0	0	1	1
$Df : \dot{Q}$	0	0	0	0	0	0	1	0	1	1
$Df : F$	1	1	1	0	0	0	0	0	1	1
<i>Réservoir</i>	1	0	1	0	0	0	0	0	1	1
<i>Vanne</i>	0	1	0	0	0	0	0	0	1	1
<i>Pompe</i>	1	0	1	0	0	1	0	0	1	1
<i>Chauffe</i>	0	0	1	0	0	0	1	0	1	1
<i>LC</i>	0	0	0	1	0	0	0	0	1	1
<i>TC</i>	0	0	0	0	1	0	0	0	1	1

TAB. 2.4 – Matrice de signature de défaillances du système

2.4 Procédure de détection et d'isolation des défaillances

Une fois les relations de redondance analytique obtenues et la matrice de signature de défaillances générée, la procédure de décision est la prochaine étape dans la conception d'un système de surveillance. Elle consiste à rendre la détection et la localisation des défaillances robuste, fiable et efficace. En général, la logique suivie pour conclure si oui ou non une défaillance est apparue, dépend du type de connaissance disponible concernant le processus, de l'approche utilisée et du type des défaillances traitées. Un large éventail d'algorithmes de décision est donné dans [45].

La procédure de décision est appliquée sur l'ensemble des résidus r (Eq. (2.17)) pour générer un vecteur de cohérence binaire $V = [v_1 \ v_2 \ \dots \ v_n]$, qui indique si oui ou non une défaillance est présente sur le processus. Chaque élément v_i de V est obtenu en appliquant la procédure de décision correspondante $\Phi_i(r_i)$, i.e., $v_i = \Phi_i(r_i)$. Ainsi, une alarme est générée lorsque $v_i = 1 \Rightarrow V \neq [0 \ 0 \ \dots \ 0]$.

Théoriquement, si le système est en bon fonctionnement, la valeur de chaque résidu r_i

($i = 1 \dots n$) doit être nulle. Mais en pratique, et dans le cas le plus simple, $|\Psi_{\Delta T}(r_i)|$ est borné par une petite quantité ε_i , avec $\Psi_{\Delta T}$ le pré-traitement sur chaque résidu (e.g. une moyenne glissante sur un intervalle de temps ΔT). Le paramètre ε_i est un seuil fixé (par expérience par exemple). En effet, en pratique les valeurs des résidus en fonctionnement normal ne sont jamais nulles. Ceci étant dû aux erreurs de modélisation ainsi qu'aux bruit des mesures.

Pour décider de la présence d'une défaillance, chaque résidu traité est comparé à une valeur de seuil ε_i , fixée *a priori*. Dans le cas le plus simple (ce qui est notre cas) chaque élément v_i de V est obtenu en utilisant la condition suivante:

$$v_i = \begin{cases} 1, & \text{si } |\Psi_{\Delta T}(r_i)| > \varepsilon_i; \\ 0, & \text{sinon.} \end{cases} \quad (2.30)$$

A noter que la valeur choisie de ε_i ne doit pas être trop grande pour éviter les non-détections et pas trop petite pour éviter les fausses alarmes.

2.5 Conclusion

La méthode proposée dans cette première partie permet de générer des RRAs de façon systématique et algorithmique à partir d'un modèle bond graph en causalité dérivée. L'avantage majeur de cette méthode est l'utilisation d'un seul outil: les bond graphs, pour la modélisation, l'élimination des variables inconnues, la génération des RRAs et de la matrice de signature des défaillances correspondante. Les RRAs sont générées sous forme de relations symboliques à partir d'un modèle bond graph ne présentant pas de relations nonlinéaires complexes non inversibles, et de boucles algèbro-différentielles. Dans le cas contraire, l'obtention des RRAs symboliques n'est donc pas possible. On peut alors contourner ce problème en utilisant des méthodes numériques de résolution et des logiciels appropriés. La partie suivante de ce chapitre traite du cas des contraintes causales non-inversibles et propose des solutions à ces problèmes.

Deuxième partie

**Cas de contraintes causales
non-inversibles**

2.6 Introduction

Dans cette deuxième partie du chapitre, est présentée une méthodologie permettant d'obtenir des résidus numériques à partir de modèles bond graphs présentant des nonlinéarités complexes non inversibles et/ou des boucles algébro-différentielles. Des modifications ont été alors apportées au modèle bond graph initial du processus physique. Ces modifications ont pour but final d'obtenir une forme de modèle dans lequel les variables connues (mesures, sources et paramètres) sont les entrées, et les résidus numériques sont des sorties. Cette forme de représentation est appelée modèle Bond Graph de Diagnostic (BGDg). Un tel modèle peut être ensuite utilisé pour aussi bien le calcul en temps réel de résidus que pour leur vérification en hors ligne en se servant des données du processus déjà disponibles.

2.7 Contraintes non-inversibles

La méthode classique, de génération des RRAs et des résidus correspondants, développée dans la première partie de ce chapitre n'est pas applicable aux modèles présentant des irrégularités. Les modifications proposées dans ce chapitre sont générées par un traitement systématique et ont pour but de contourner le problème d'élimination des variables inconnues et de générer des résidus sous forme numérique (par opposition à la forme symbolique) même en présence de contraintes non-inversibles ou de boucles.

Pour introduire la problématique et montrer les motivations derrière ce travail, prenons comme exemple le modèle bond graph en causalité dérivée préférentielle de la figure 2.7. Ce modèle présente deux boucles: une algébrique (A) entre les éléments R_{V1} et R_{V2} et l'autre différentielle (B) formée par la boucle causale entre les éléments C_{T1} et R_{V2} .

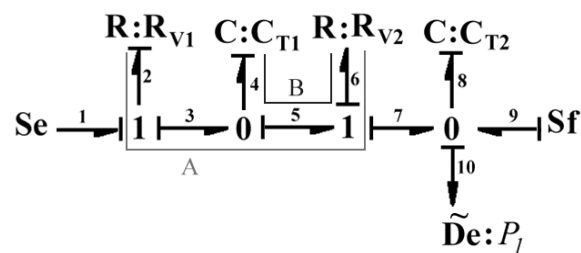


FIG. 2.7 – Modèle bond graph avec deux boucles: ‘A’ algébrique et ‘B’ différentielle

Physiquement, les éléments R (R_{V1} et R_{V2}) et C (C_{T1} et C_{T2}) peuvent représenter respectivement des vannes et des réservoirs de stockage. Si on admet que la causalité affectée à R_{V1} est fixe et correspond à celle de la figure 2.7, sa relation constitutive, exprimant la variable flux f_2 en fonction de la variables effort e_2 , peut être alors exprimée par la relation non linéaire suivante:

$$f_2 = \Phi_{V1}(e_2),$$

avec Φ_{V1} une relation non-inversible. De même, la relation constitutive de R_{V2} est donnée par

$$e_6 = \Phi_{V2}(f_6).$$

La boucle algébrique qui lie les deux éléments R est exprimée analytiquement par la relation suivante:

$$f_2 = \Phi_{V1}(Se - \Phi_{V2}(f_2 - f_4) - P_1). \quad (2.31)$$

La relation 2.31 exprime la variable f_2 comme une fonction non linéaire d'elle même qui, en plus, ne permet pas de calculer f_2 puisqu'on a supposé que la causalités des éléments R est fixe et que les relations Φ_{V1} et Φ_{V2} sont non-inversibles. l'équation 2.31 peut alors être réécrite comme:

$$f_2 = \Psi_1(f_2, f_4, P_1),$$

qui exprime bien une boucle algébrique non solvable.

La deuxième boucle, différentielle, est due à l'existence d'un chemin causal entre un élément $R : R_{V1}$ et un élément $C : C_{T1}$. En supposant que les relations constitutives des éléments de stockage sont linéaires, l'équation correspondant à cette boucle est donnée par

$$e_4 = \Phi_{V2}\left(f_3 - C_{T1} \cdot \frac{de_4}{dt}\right) + P_1,$$

qui peut être écrite sous la forme suivante:

$$e_4 = \Psi_2\left(f_3, \frac{de_4}{dt}, P_1\right). \quad (2.32)$$

Le problème avec la présence des boucles est qu'il est impossible, lorsque la variable inconnue ne peut pas être calculée, de générer des RRAs symboliques. En effet, d'après la proposition 3 (voir la première partie de ce chapitre), à l'exemple de la figure 2.7 doit correspondre une RRA puisque le système représenté, par son modèle bond graph, est observable et qu'il contient un détecteur ($De : P_1$). En appliquant l'algorithme de génération des RRAs on peut écrire

$$f_7 + f_9 - f_8 = 0,$$

avec $f_9 = Sf$ et $f_8 = C_{T2} \cdot \frac{dP_1}{dt}$. Ces deux variables inconnues étant calculées en fonction de variables connues, il reste à déterminer f_7 . On a:

$$f_7 = f_5 - f_4 = \Phi_{V1}(e_2) - C_{T1} \cdot \frac{de_4}{dt},$$

ce qui ne permet pas d'obtenir la valeur de f_7 puisque la variable e_4 est présente dans la boucle différentielle correspondant à l'Eq. (2.32). Donc la génération de la relation de redondance analytique n'est pas possible dans ce cas. La méthode classique de génération de RRAs, décrite dans la première partie de ce chapitre, présente donc un inconvénient. En effet, la méthode ne peut pas être appliquée lorsque des boucles algébriques et différentielles non solvables sont présentes dans le modèle bond graph en causalité dérivée préférentielle.

Toutefois, il est possible, au moyen de logiciels appropriés (Matlab-Simulink, 20-Sim [46, 47], Symbols [18, 39], etc.), de résoudre la plupart des boucles. Le but de cette partie est donc de proposer un ensemble de modifications sur le modèle bond graph initial du processus afin de résoudre numériquement les boucles et générer ainsi des résidus et la matrice de signature des défaillances correspondante.

Des modifications sur le modèle bond graph en causalité dérivée préférentielle sont réalisées. Elles permettent d'obtenir un modèle final dans lequel les entrées sont des données connues ou mesurées (sources, détecteurs, paramètres) et les sorties sont des résidus numériques.

2.8 Détecteurs de résidus numériques

Avant d'introduire les modifications sur le modèle bond graph, on a besoin de définir quelques éléments nouveaux utilisés dans la suite de cette partie.

Un élément détecteur Ds , est un élément qui mesure la valeur d'un signal quelconque à l'exception du flux et de l'effort. Par exemple, un capteur mesurant la sortie d'un régulateur est représenté par un élément Ds . Lorsque la causalité de cet élément est inversée, on représente l'élément correspondant obtenu par un détecteur de source, (source de signal), appelé Ss . A signaler que l'élément Ss a été déjà présenté dans [9] pour une utilisation plus générale dans l'inversion des modèles pour le calcul des lois de commande. Ces deux éléments, Ds et Ss , permettent en surveillance de représenter les parties commande des processus.

En affectant la causalité dérivée préférentielle au modèle bond graph, avec si possible inversion de la causalité des détecteurs, cinq configurations possibles peuvent apparaître:

1. causalité inversée dans un détecteur d'effort (De),
2. causalité inversée dans un détecteur de flux (Df),
3. causalité non inversée d'un détecteur d'effort (De),
4. causalité non inversée d'un détecteur de flux (Df),
5. inversion d'un détecteur de signal, Ds , pour donner lieu à une source de signal, Ss (toujours inversée).

En outre, on suppose dans ce travail que la causalité des sorties des détecteurs liés aux régulateurs n'est jamais inversée. En effet, les sorties des régulateurs correspondent aux valeurs de retour des lois de commandes programmées. Par conséquent, elles ne sont pas de nature flux ou effort et donc la notion de causalité ne leur est pas appliquée.

2.8.1 Modifications sur le modèle bond graph en causalité dérivée

En s'inspirant de la méthode développée dans la première partie, on peut à partir d'un modèle bond graph en causalité dérivée générer des résidus même en présence de contraintes causales non-inversibles. L'idée est d'utiliser le bloc diagramme correspondant au modèle bond graph, pour résoudre numériquement par des logiciels appropriés ces contraintes et générer ainsi des relations numériques de compatibilité.

Considérons un détecteur d'effort De . En inversant sa causalité, on sait que le flux renvoyé par ce détecteur est nul. En outre, on sait, d'après la syntaxe d'interconnexion des composants développée dans le premier chapitre, qu'un détecteur d'effort ne peut être connecté qu'à une jonction 0 (sans considérer les régulateurs). Puisque la génération de résidus se fait sur un modèle bond graph en causalité dérivée, un détecteur d'effort dont la causalité est inversée peut être considéré comme une source d'effort modulée par la valeur mesurée comme le montre la figure 2.8 (c). Si on applique l'algorithme de génération de RRAs, sur une jonction 0 comportant un détecteur d'effort De en causalité inversée on peut écrire:

$$f_{De} = \sum_{i_in=1}^n f_{i_in} - \sum_{i_out=1}^n f_{i_out}, \quad (2.33)$$

avec f_{i_in} et f_{i_out} sont respectivement les flux d'entrée et les flux de sortie de la jonction 0, et f_{De} la variable flux correspondant au détecteur d'effort De . Puisque $f_{De} = 0$, l'Eq. (2.33) devient:

$$\sum_{i_in=1}^n f_{i_in} - \sum_{i_out=1}^n f_{i_out} = 0, \quad (2.34)$$

qui, une fois les variables f_{i_in} et f_{i_out} calculées, conduit à une relation de redondance analytique. Cependant, dans le cas de contraintes causales non-inversibles sur le modèle, l'élimination de ces variables devient impossible. Il faut donc résoudre numériquement l'Eq. (2.34) pour obtenir le résidu. Dans la suite de ce travail, la variable réactive (ici le flux) d'un détecteur d'effort dont la causalité est inversée correspond à un détecteur de résidu numérique représenté par l'élément Df^* comme le montre la figure 2.8(c).

Le même raisonnement s'applique à un détecteur de flux dont la causalité est inversée pour générer un résidu numérique représenté par l'élément De^* . Les modifications apportées

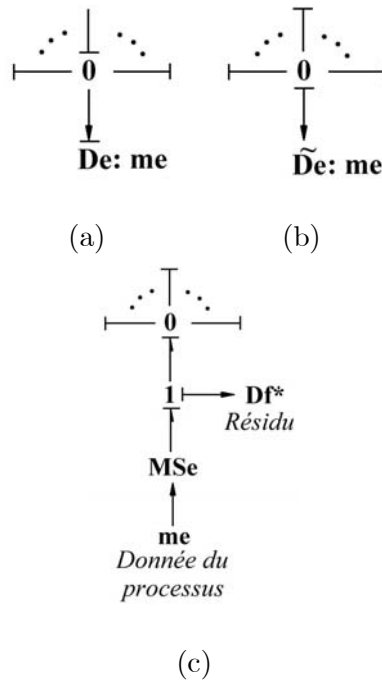


FIG. 2.8 – (a) De en causalité intégrale (b) De en causalité inversée (c) représentation proposée pour la causalité inversée de De donnant lieu au détecteur de résidu Df^*

à ce type de détecteur sont données en figure 2.9. Cette représentation est la forme duale à celle du détecteur d'effort.

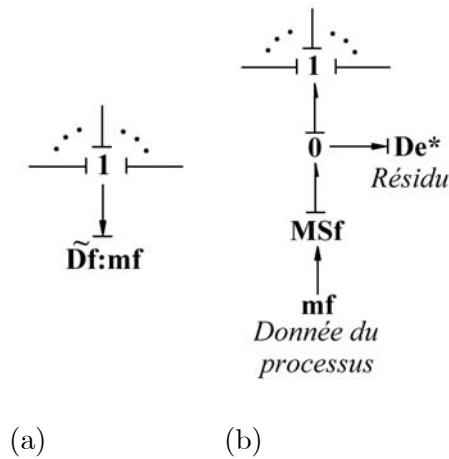


FIG. 2.9 – (a) Df en causalité inversée (b) substitution correspondante et détecteur de résidu

Concernant un détecteur d'effort dont la causalité ne peut pas être inversée, les modifications proposées sont données par la figure 2.10(a). La valeur mesurée de ce détecteur peut être déterminée à travers les mesures des autres détecteurs de même nature (effort ou flux). La différence entre les deux quantités est nulle et le résultat de l'équation n'est rien d'autre qu'un

résidu, qui est représenté sur le modèle bond graph par un détecteur de résidu numérique comme le montre la figure 2.10(b).

Il a été montré dans [8] qu'un détecteur dont la causalité ne peut pas être inversée lorsqu'on met le modèle bond graph en causalité dérivée préférentielle est un détecteur redondant du point de vue structurel. Le résidu obtenu dans ce cas est un résidu direct (i.e. sans passer par l'élimination de variables inconnues).

Les modifications apportées à un détecteur de flux dont la causalité ne peut pas être inversée donnent lieu à une forme duale à celle donnée par la figure 2.10(b).

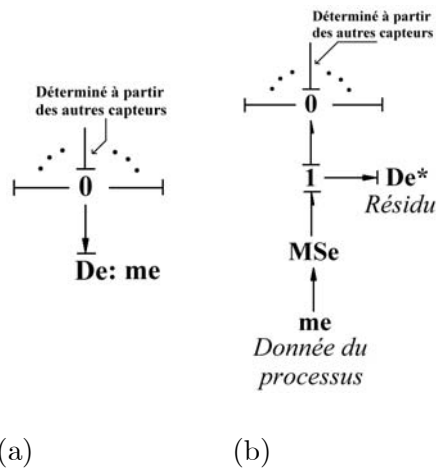


FIG. 2.10 – (a) causalité non inversée d'un détecteur d'effort, De , (b) substitution correspondante et détecteur de résidu obtenu

Une fois que ces modifications sont faites sur le modèle bond graph en causalité dérivée préférentielle, on obtient un modèle dans lequel les entrées sont les mesures et les valeurs des sources, et les sorties sont les détecteurs de résidus numériques. Comme on le verra plus loin dans cette partie, on peut directement à partir du modèle bond graph final obtenu construire le bloc diagramme correspondant et le mettre dans un logiciel (Matlab-Simulink, 20-Sim ou Symbols) pour enfin résoudre numériquement les différentes boucles et les relations non-inversibles et générer des résidus numériques.

A noter que le nombre de résidu obtenu dépend fortement de l'observabilité du système. En effet, on ne peut pas générer des résidus à partir d'une partie de graphe non observable et ce même avec la résolution numérique des contraintes.

D'après la description précédente, pour les détecteurs dont la causalité peut être inversée, on peut conclure que le type de résidu obtenu renseigne sur la nature de la loi de conservation de la jonction (0 ou 1) utilisée. Un détecteur de résidu numérique fourni par un élément Df^* correspond à un détecteur d'effort qui doit être connecté à une jonction 0. De même, un résidu obtenu à partir d'un élément De^* doit venir d'un détecteur de flux connecté à une

jonction 1. Physiquement, et d'après le premier chapitre, l'identification du type de jonction nous renseigne sur la classe du composant utilisé.

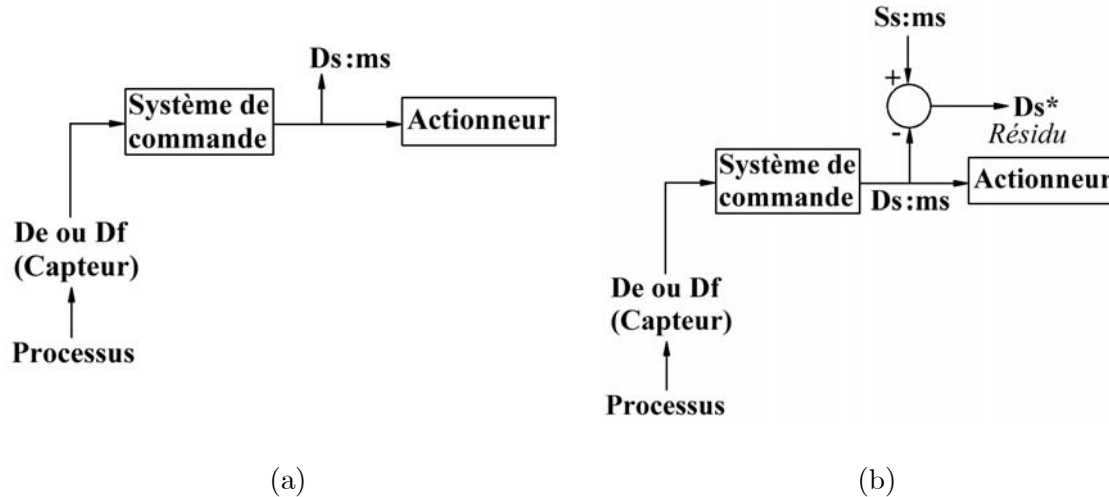


FIG. 2.11 – (a) détecteur de signal, D_s , (b) substitution correspondante et capteur de résidu

Concernant le système d'information constitué du système de commande (voir figure 2.5), la génération de résidus est relativement simple. Les sorties des régulateurs sont des signaux à puissance nulle et dont les valeurs sont supposées connues (mesurées). En outre, dans la plupart des applications, les algorithmes de commande sont implémentés dans des logiciels de supervision. La sortie du régulateur peut donc être considérée comme connue puisqu'elle est fournie par un capteur logiciel.

Les résidus générés à partir des régulateurs sont en fait de simples comparaisons entre les mesures des sorties des régulateurs et celles prédites par les algorithmes de commande. Le résidu numérique obtenu est représenté par un élément détecteur D_s^* , comme le montre la figure 2.11 (b).

2.8.2 Génération des résidus

D'après les modifications proposées dans la section précédente, il paraît évident qu'à chaque capteur correspond un détecteur de résidu numérique (pour un système totalement observable). Le nombre de résidus obtenu est donc égal au nombre de capteurs présents sur le processus physique. Comme dans la méthode classique de génération de RRAs sous forme symbolique, on distingue deux sortes de résidus: des résidus déduits correspondants aux détecteurs dont la causalité a été inversée, et des résidus directs obtenus à partir des détecteurs dont la causalité ne peut pas être inversée. En outre, les résidus obtenus sont structurellement indépendants. Cette propriété est justifiée par le fait que parmi les résidus générés aucun ne contient les mêmes éléments que les autres ce qui permet d'avoir des signatures différentes.

En effet, pour les résidus directs il est facile de vérifier qu'ils sont tous différents. Les résidus déduits correspondent à des détecteurs qui ne sont pas structurellement redondants.

La procédure de génération de résidus numériques à partir d'un P&ID d'un processus physique comporte les étapes suivantes:

1. construire le modèle bond graph en causalité intégrale préférentielle,
2. mettre le modèle bond graph obtenu en causalité dérivée préférentielle,
3. introduire les modifications sur tous les détecteurs pour générer des résidus numériques,
4. construire à partir du modèle bond graph en causalité intégrale le bloc diagramme correspondant,
5. faire la même chose que dans l'étape 4 sur le modèle bond graph en causalité dérivée (avec les modifications),
6. en l'absence d'installation réelle, coupler les deux modèles de telle sorte que les sorties du modèle en causalité intégrale soient les entrées de celui en causalité dérivée. Sinon, injecter les sorties des capteurs dans le modèle en causalité dérivée,
7. les résidus sont obtenus à partir des détecteurs de résidus numériques du modèle bond graph en causalité dérivée.

2.8.2.1 Exemple d'application

Pour appliquer cette méthode on a choisi un système de trois réservoirs, commandé par deux régulateurs: un *PI* et un *ToR (LC)* comme le montre la figure 2.12. L'eau est pompée vers le premier réservoir par une pompe. On suppose que la source d'eau est située à un niveau h suffisamment très bas. Le régulateur de niveau *LC* a pour but de maintenir à une valeur constante le niveau d'eau, mesuré par le capteur de pression P_1 , dans le réservoir T_1 en agissant sur la pompe. Le débit massique de la pompe est mesuré par le capteur F_1 et la sortie du régulateur *LC* par le capteur u_1 . Le débit de sortie vers l'environnement externe qui est mesuré par le capteur F_2 , est régulé à une certaine valeur par le *PI* en agissant sur la vanne V_3 . Les vannes, V_1 et V_2 , sont ouvertes à des positions fixes (leurs coefficients de décharge sont donc constants).

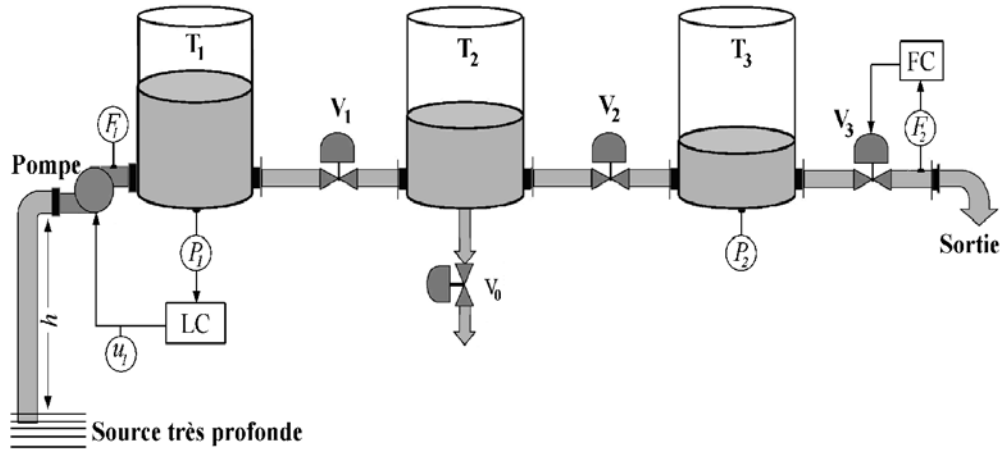


FIG. 2.12 – Système commandé à trois réservoirs.

Le modèle pseudo-bond graph de ce système en causalité intégrale préférentielle est donné à la figure 2.13. Dans ce modèle, la pompe est modélisée par un élément résistance R . Comme on l'a vu dans le premier chapitre, la courbe caractéristique de la pompe exprime le débit comme une fonction non linéaire de la différence des pressions à ses bornes. Si en plus on suppose que cette fonction ne peut pas être inversée, alors l'élément résistance R représentant la pompe reste toujours en causalité conductance sur le modèle bond graph.

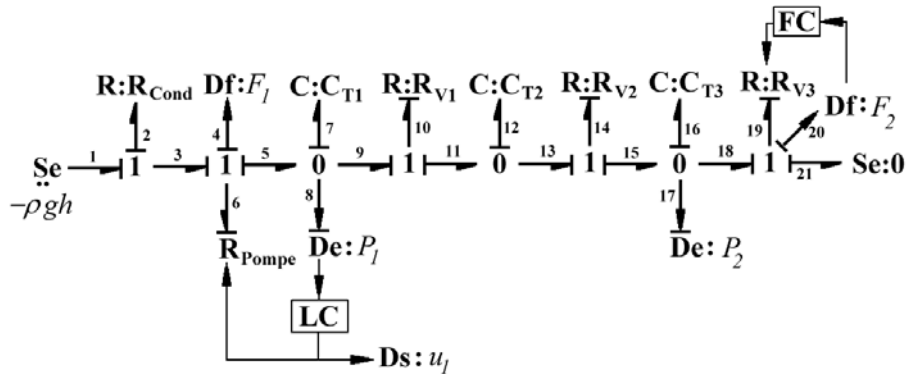


FIG. 2.13 – Modèle BG du système à trois réservoirs.

Le modèle bond graph en causalité dérivée préférentielle obtenu à partir du modèle en causalité intégrale est donné à la figure 2.14. A cause de l'hypothèse formulée sur la causalité fixe (conductance) de la pompe, quatre boucles sont générées. La première boucle (qui est algébrique et indiquée par la lettre 'A' sur la figure 2.14) est due au couplage entre deux éléments R correspondant à la pompe et à la conduite. Cette boucle est présente aussi dans le modèle bond graph du système en causalité intégrale de la figure 2.13. Sur ce modèle, cette

boucle peut être évitée en ajoutant un élément de stockage C , entre les éléments $R : R_{Cond}$ et $R : R_{Pompe}$. Cette méthode développée dans [3, 6], permet de supprimer la boucle algébrique mais augmente le temps de simulation. En effet, l'élément C introduit une faible constante de temps dans le modèle et par conséquent ajoute une nouvelle variable d'état au modèle. La deuxième boucle algébrique, marquée par 'B' sur la figure 2.14, apparaît lorsqu'on met le modèle bond graph en causalité dérivée préférentielle. Les boucles algébriques sont aussi appelées ZCPs (pour Zero-order Causal Path) [48, 49, 40].

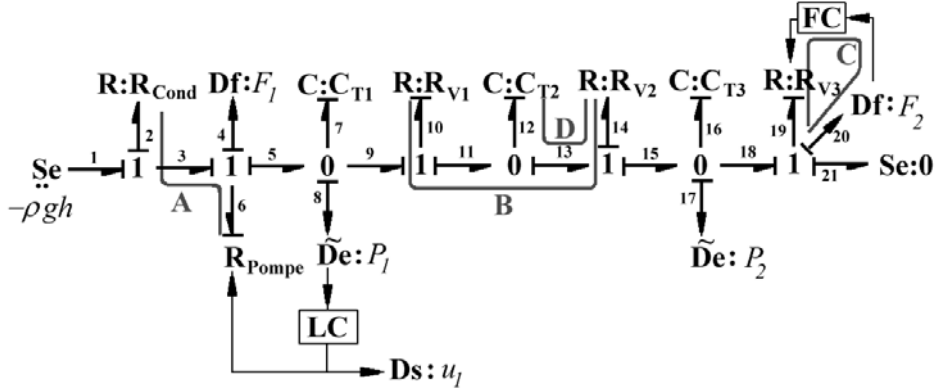


FIG. 2.14 – Modèle BG en causalité dérivée préférentielle du système à trois réservoirs.

Une troisième boucle, différentielle cette fois et marquée par 'D', est présente sur le modèle bond graph de la figure 2.14. Cette boucle est due à l'existence d'un chemin causal bouclé entre l'élément $R : R_{V2}$ et l'élément de stockage $C : C_{T2}$. Ce type de boucle est plus critique qu'une simple ZCP car elle ne peut pas être résolue symboliquement sans faire intervenir une forme intégrale, et donc des conditions initiales, dans la solution. Cependant, des méthodes numériques ([40, 50, 51]) existent pour la résolution de ces boucles permettant ainsi de générer des résidus numériques.

Soit P_b la pression juste à l'entrée de la pompe et dont la valeur est considérée inconnue. La pression P_1 à la sortie de la pompe est quant à elle mesurée par le détecteur d'effort $De : P_1$. Le débit f est le même à travers la pompe et la conduite, et sa valeur est donnée par l'équation suivante:

$$f = \Phi_p(P_b - P_1).OnOff(P_1) = C_d \cdot \sqrt{-\rho \cdot g \cdot h - P_b}, \quad (2.35)$$

où P_b est une valeur négative correspondant à la pression du vide créée par la pompe ($P_b < -\rho \cdot g \cdot h$), C_d est le coefficient de décharge de la conduite et Φ_p est la fonction caractéristique de la pompe. La boucle 'A' sur le modèle bond graph en causalité dérivée peut être résolue à cause de la présence du détecteur de flux $Df : F$. On a alors deux équations et une inconnue

P_b dans l'Eq. (2.35); ce qui permet d'obtenir le résidu suivant:

$$F_1 - \Phi_p \left(-\frac{F_1^2}{C_d^2} - \rho.g.h - P_1 \right) . OnOff(P_1) = 0. \quad (2.36)$$

Considérant maintenant la boucle algébro-différentielle formée par les chemins marqués par 'B' et 'D' sur la figure 2.14. Cette boucle n'existait pas sur le modèle bond graph en causalité intégrale mais elle est apparue à cause de la causalité dérivée affectée à l'élément C_{T2} . Le débit f_9 dans le lien $n^\circ 9$ peut être exprimé en fonction de P_1 et e_{13} (Eq. (2.37)). Cependant, dans cette équation si la variable f_9 ne peut pas être calculée à partir de variables connues et sans faire intervenir les valeurs initiales des états; alors le calcul de toute autre variable en fonction de f_9 devient impossible.

$$f_9 = R_{V1}(P_1 - e_{13}), \quad (2.37)$$

e_{13} peut être exprimée par l'équation suivante:

$$e_{13} = e_{14} + P_2 = R_{V2}^{-1}(f_{13}) + P_2 = R_{V2}^{-1} \left(f_9 - C_{T2} \cdot \frac{de_{13}}{dt} \right) + P_2 \quad (2.38)$$

où R_{V1} et R_{V2} représentent respectivement, les caractéristiques des vannes V_1 et V_2 . Les équations 2.37 et 2.38 forment la boucle algébro-différentielle entre les variables f_9 et e_{13} et leur solution ne peut pas être obtenue sous forme symbolique. En admettant que dans les lois constitutives des deux éléments R_{V1} et R_{V2} le flux f est une fonction linéaire de l'effort e :

$$e = R.f \Rightarrow f = \frac{e}{R},$$

la valeur de e_{13} est alors la solution d'une équation différentielle du premier ordre:

$$e_{13} = \alpha.e^{-\frac{1}{C_{T2}} \cdot \left(\frac{R_{V1} + R_{V2}}{R_{V1} \cdot R_{V2}} \right) \cdot t} + \frac{P_1 R_{V2} + P_2 R_{V1}}{R_{V1} + R_{V2}} \quad (2.39)$$

où α est une constante qui dépend des valeurs initiales de l'état du réservoir C_{T2} . Ainsi, tout résidu généré sous forme symbolique à partir des boucles 'B-D' sera dépendant des conditions initiales. Dans le cas où les éléments R_{V1} et R_{V2} sont nonlinéaires, seule une solution numérique peut déterminer e_{13} .

Remarque 2.8.1 *Pour casser la boucle algébro-différentielle formée par les éléments R_{V1} , R_{V2} et C_{T2} ; un détecteur d'effort De (dans ce cas un capteur de pression) au niveau du réservoir T_2 suffit. Cependant, pour des contraintes financières ou physiques (milieu agressif) d'un côté et, parfois, l'impossibilité de mesurer des variables de puissance (e.g. flux d'enthalpie, entropie) de l'autre côté rendent inévitable la présence de ces boucles.*

Sur la figure 2.14, le chemin marqué par ‘C’ représente une boucle algébrique qui existe dans le modèle bond graph en causalité intégrale préférentielle de la figure 2.13. Cette boucle algébrique est due à la contre-réaction du régulateur *PI*. L’équation du débit f_{19} est donnée par la relation suivante:

$$f_{19} = C_{dV3}(x) \cdot \sqrt{P_2}, \quad (2.40)$$

$$x = PI(f_{19}),$$

où x représente la sortie du régulateur. La forme implicite de l’Eq. (2.40) correspond à une boucle algébrique sur le modèle en causalité intégrale. Cependant, sur le modèle en causalité dérivée préférentielle utilisé pour la surveillance; cette boucle disparaît puisque l’entrée du régulateur est mesurée ($f_{19} = F_2$).

A partir des discussions précédentes, on voit bien que les boucles présentes sur le modèle bond graph en causalité dérivée peuvent conduire à un ensemble d’équations algèbro-différentielles implicites sans solution. La génération, sous forme symbolique, des RRAs et des résidus correspondants ainsi que de la matrice de signature des défaillances est alors impossible.

L’étapes 3 de la procédure donnée à la section 2.8.2 nous permet de générer numériquement tous les résidus possibles (5 résidus pour 5 détecteurs). Pour cela, il faut d’abord apporter les modifications nécessaires sur le modèle bond graph en causalité dérivée pour obtenir le modèle bond graph de diagnostic (BGDg) de la figure 2.15.

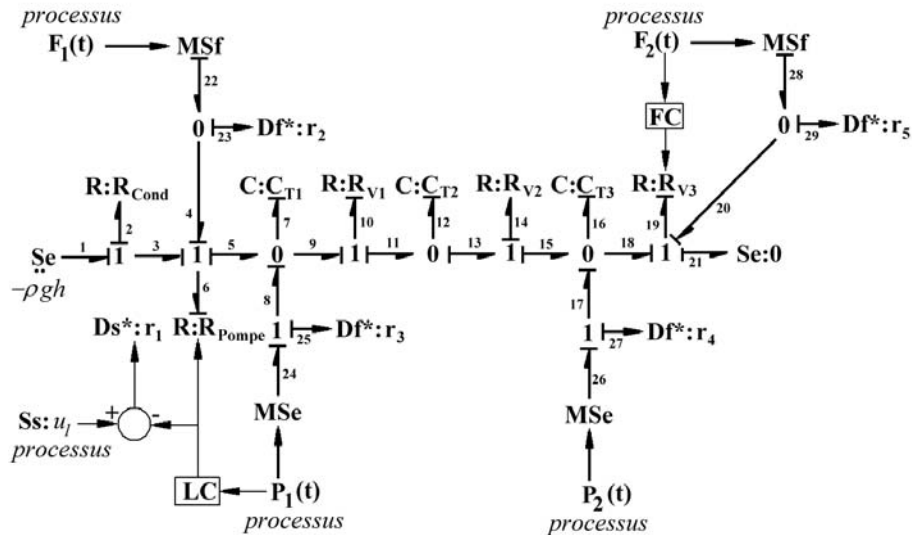


FIG. 2.15 – BGDg du système des trois réservoirs.

Le BGDg de la figure 2.15 possède cinq entrées (correspondant aux cinq capteurs du système) et cinq sorties (correspondant aux cinq résidus numériques à générer) comme l'illustre le bloc diagramme de la figure 2.16.

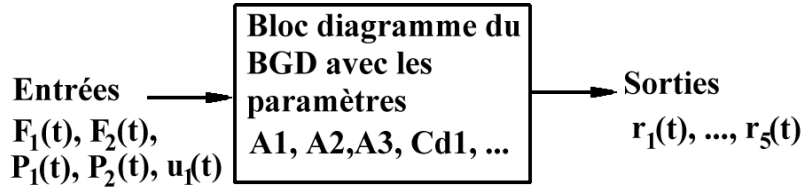


FIG. 2.16 – Entrée-sortie du BGDg sous forme de bloc diagramme.

2.8.2.2 Matrice de signature des défaillances

Le modèle BGDg permet d'obtenir les valeurs numériques des résidus mais aussi la matrice de signature des défaillances S_{ji} . En effet, cette dernière est indispensable pour la localisation des défaillances pouvant survenir durant le fonctionnement du système. Dans la première partie de ce chapitre, on a vu que le parcours des chemins causaux permettait d'éliminer les variables inconnues pour générer des RRAs. Dans cette partie, le parcours des chemins causaux de la variable inconnue vers le détecteur est utilisé pour construire la matrice de signature des défaillances. Ainsi, un résidu r_j est sensible à une défaillance dans le composant $Comp_i$ si et seulement si la variable associée à ce dernier est présente dans le chemin causal permettant la génération du résidu r_j . Par exemple, les composants intervenant dans le détecteur de résidu r_2 sont déterminés par les chemins causaux suivants:

1. $F_1(t) \rightarrow f_{22} \rightarrow f_{23}$
2. $-\rho gh \rightarrow e_1 \rightarrow e_3 \rightarrow e_6 \rightarrow R_{Pompe} \rightarrow f_6 \rightarrow f_4 \rightarrow f_{23}$
3. $f_2 \rightarrow R_{Cond} \rightarrow e_2 \rightarrow e_3 \rightarrow e_6 \rightarrow R_{Pompe} \rightarrow f_6 \rightarrow f_4 \rightarrow f_{23}$
4. $P_1(t) \rightarrow e_{24} \rightarrow e_8 \rightarrow e_5 \rightarrow e_6 \rightarrow R_{Pompe} \rightarrow f_6 \rightarrow f_4 \rightarrow f_{23}$
5. $P_1(t) \rightarrow LC \rightarrow R_{Pompe} \rightarrow f_6 \rightarrow f_4 \rightarrow f_{23}$

Les composants impliqués dans le résidu r_2 sont donc donnés par le vecteur $K_2 = [F_1, R_{Pompe}, R_{Cond}, P_1, LC]$. L'application de cette procédure sur tous les détecteurs de résidus numériques conduit à la matrice de signature des défaillances S_{ji} . La matrice des signatures, du système des trois réservoirs de la figure 2.12, est donnée par le tableau 2.5.

Remarque 2.8.2 *On peut montrer, en utilisant l'algorithme donné à la première partie de ce chapitre, que le résidu r_2 peut être généré sous forme symbolique et que sa signature est identique à celle obtenue par l'analyse des chemins causaux. L'avantage de la méthode numérique, est que les résidus et la matrice des signatures peuvent être obtenus même dans le cas où les contraintes ne sont pas solvables symboliquement.*

	r_1	r_2	r_3	r_4	r_5	D_b	I_b
u_1	1	0	0	0	0	1	1
$Df : F_1$	0	1	0	0	0	1	1
$De : P_1$	1	1	1	1	0	1	1
$De : P_2$	0	0	1	1	1	1	1
$Df : F_2$	0	0	0	1	1	1	0
<i>Conduite</i>	0	1	1	0	0	1	0
<i>Pompe</i>	0	1	1	0	0	1	0
T_1	0	0	1	0	0	1	1
V_1	0	0	1	1	0	1	0
T_2	0	0	1	1	0	1	0
V_2	0	0	1	1	0	1	0
T_3	0	0	0	1	0	1	1
V_3	0	0	0	1	1	1	0
<i>LC</i>	1	1	1	0	0	1	1
<i>PI (FC)</i>	0	0	0	1	1	1	0

TAB. 2.5 – Matrice de signature des défaillances du système des trois réservoirs

Les signatures de la conduite et de la pompe sont identiques ce qui signifie que les défaillances affectant ces composants ne peuvent pas être isolées. En revanche, si on considère que ces deux éléments forment un seul composant physique, leur défaillance peut être localisée. De même, les signatures des composants T_2 , V_1 et V_2 sont identiques et par conséquent leurs défaillances ne sont pas isolables.

Il est important de souligner que la matrice de signature des défaillances construite à partir des chemins causaux correspond à une configuration (ou mode de fonctionnement) bien déterminée et donc au modèle associé. La forme des équations pour chaque élément bond graph est alors la même durant toute la période de fonctionnement dans une configuration donnée. Par exemple, le blocage total de la vanne V_2 représentée par l'élément R_{V_2} provoque la rupture de tous les chemins causaux de (et vers) cet élément. De tels cas sont traités

comme des modèles à part conduisant à des matrices de signature de défaillances différentes et la gestion des différents modes de fonctionnement peut se faire en utilisant les modèles externes [42].

2.8.2.3 Génération de résidus numériques

Après avoir construit et introduit les modifications nécessaires sur le modèle bond graph en causalité dérivée du système, on procède à la génération du bloc diagramme correspondant. Ce dernier sera ensuite implémenté dans un logiciel approprié pour obtenir automatiquement et numériquement les résidus. La procédure de génération du bloc diagramme à partir d'un modèle bond graph est bien développée dans [5]. Il existe plusieurs outils logiciels pour la résolution des équations implicites et des boucles algèbro-différentielles et aussi pour des applications en temps réel. Matlab-Simulink [52, 53] est l'un de ces logiciels. Il fournit une boîte à outils appelée 'Real Time Work Shop' pour coupler les modèles Simulink aux processus réels et accepte les modèles sous la forme de bloc diagramme. En outre, des relations mathématiques peuvent être écrites sous forme de fonctions Matlab ou autres blocs Simulink. Le bloc diagramme représentant le BGDg du système des trois réservoirs de la figure 2.15 est donné à la figure 2.17.

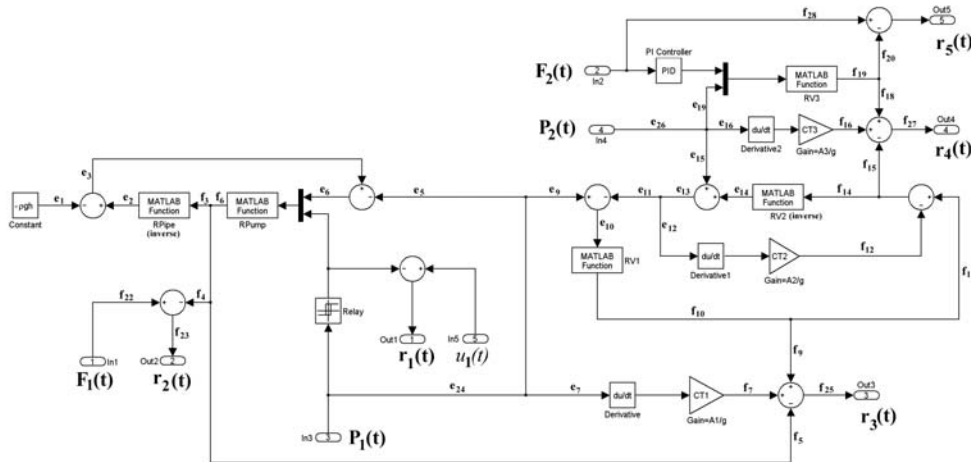


FIG. 2.17 – Bloc diagramme du BGDg du système des trois réservoirs

Le modèle Simulink correspondant au bloc diagramme peut ensuite être utilisé pour générer une '*S – fonction*' soit sous Matlab-Simulink soit sur d'autres logiciels à base des bond graphs (Symbols, 20-Sim, etc.).

Pratiquement, il existe deux moyens pour générer des résidus numériques et surveiller le processus physique: le premier consiste à construire d'abord le bloc diagramme du modèle

bond graph de diagnostic, l'implémenter sous Simulink et ensuite lui injecter les données acquises directement des capteurs du processus. Cette façon de procéder nécessite donc la disponibilité physique du processus avec toute son instrumentation (capteurs, cartes d'acquisition, interface avec Matlab-Simulink, etc.). Dans le cas contraire, c'est les sorties du modèle bond graph en causalité intégrale du processus qui serviront de données capteurs du modèle BGDg.

Le besoin de simulation Dans la plupart des applications industrielles, l'introduction de défaillances pour la génération des modes de fonctionnement correspondants peut être très risquée et dangereuse, e.g. fuite de gaz toxique d'une cuve de stockage, blocage d'une conduite transportant un liquide de refroidissement, etc. Et plus important encore, le processus doit être suffisamment équipé pour réaliser de telles défaillances. Sur le système des trois réservoirs de la figure 2.12, une fuite à partir du réservoir T_2 peut être réalisée en ouvrant la vanne V_0 . Cependant, avec l'instrumentation existant, il n'y a aucune possibilité de produire des fuites à partir de T_1 et T_3 . Parallèlement, il est très difficile d'introduire des fautes sur des régulateurs et actionneurs physiques (matériels). En outre, l'acquisition des données à partir des processus ayant des dynamiques lentes peut prendre beaucoup de temps.

Pour contourner toutes ces contraintes, la simulation du comportement du processus peut être exploitée dans le but de produire une base de données riche pour des tests hors lignes. Ainsi, toutes les défaillances peuvent être introduites sur le modèle de simulation sans aucun risque sur l'opérateur et sur le processus.

2.8.2.4 Construction du modèle de simulation

Le modèle global de simulation se compose de deux modèles distincts: le modèle bond graph en causalité intégrale avec les différentes défaillances possibles et le modèle bond graph en causalité dérivée qu'on appelle aussi BGDg. Les sorties du premier sont injectées comme entrées du second afin de générer les résidus et la matrice de signature des défaillances correspondante.

Connexion des deux modèles Le couplage des deux modèles bond graphs, en causalité dérivée et intégrale, a pour but la construction de modèles en mode normal et défaillant. En l'appliquant à l'exemple du système des trois réservoirs, on obtient le modèle couplé de la figure 2.18. Sur ce modèle, on distingue quatre zones distinctes: les événements (e.g. intervention de l'utilisateur), le modèle bond graph en causalité intégrale incluant les défaillances définies dans le cahier des charges, la zone de couplage dans laquelle on a ajouté du bruit aux sorties du modèle en causalité intégrale, et enfin le modèle bond graph de diagnostic qui permet de générer des résidus sous forme numérique.

Sur la figure 2.18, les numéros des liens et la nomenclature de certains éléments sont les mêmes dans les deux modèles. Cela ne pose aucun problème puisque ces deux modèles sont traités séparément, chacun avec sa propre nomenclature. Le modèle dynamique en causalité intégrale préférentielle comporte des sources nécessaires à l'introduction des défaillances déclenchées par l'utilisateur. Les événements peuvent être des échelons unitaires, des actions temporelles, des rampes, etc.. Les défaillances quant à elles, peuvent être de différents types (abruptes, progressives, etc.). Dans ce travail, seul le cas des défaillances simples (non multiples) est considéré. Cela veut dire qu'il ne peut pas y avoir apparition simultanée de plusieurs défaillances.

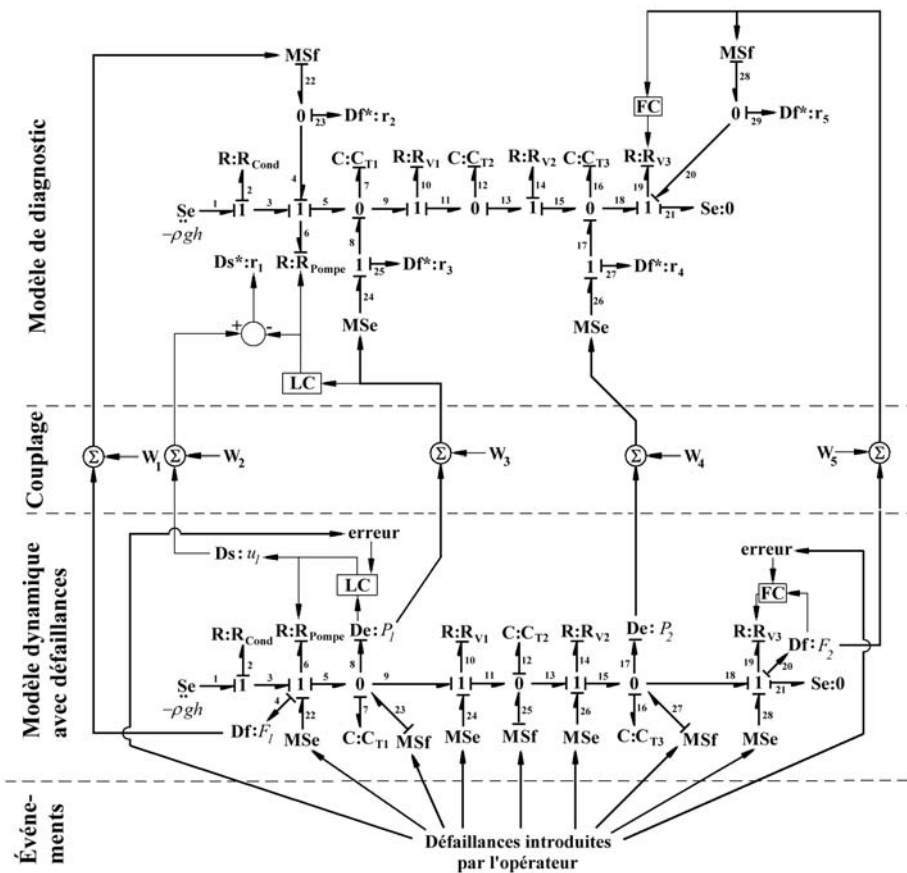


FIG. 2.18 – Couplage des modèles dynamique et de diagnostic.

La partie 'Couplage' montre l'interface entre les deux modèles, celui en causalité intégrale et celui de diagnostic. Pour mieux simuler les dynamiques réelles des capteurs et afin d'observer l'effet du bruit, toutes les sorties du modèle bond graph en causalité intégrale sont noyées dans du bruit (W_1 à W_5 sur la figure 2.18).

En fonction de l'outil logiciel utilisé, différents types de sources de bruits sont disponibles.

La source de bruit blanc fournie par Matlab-Simulink convient mieux pour notre simulation.

Les dérivées dans le modèle de diagnostic sont calculées en temps discret, tandis que dans le modèle en causalité intégrale les sorties sont évaluées en temps continu. Pour traiter le modèle global couplé, un outil de simulation hybride est alors indispensable.

Pour des besoins de simulation, il est nécessaire d'introduire quelques modifications sur les éléments de stockage en causalité dérivée afin de les représenter par des composants en causalité intégrale.

Modèle bond graph d'une différentiation Le circuit électronique d'un dérivateur analogique peut être représenté par celui de la figure 2.19(a) et son modèle bond graph correspondant est donné à la figure 2.19(b).

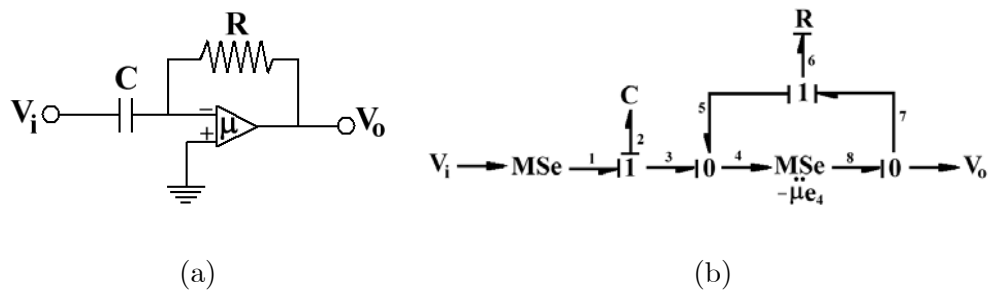


FIG. 2.19 – (a) *Dérivateur analogique*, (b) *modèle bond graph*.

En appliquant la règle de Mason sur le modèle de la figure 2.19(b), le gain $G(s)$ entre l'entrée et la sortie du dérivateur est calculée par la relation suivante:

$$G(s) = \frac{-\mu}{1 + \frac{1}{R.C.s} + \frac{\mu}{R.C.s}}, \quad (2.41)$$

dans laquelle lorsque $\mu \rightarrow \infty$ on a $G(s) \rightarrow -R.C.s$. En prenant $R = \frac{1}{C}$, et $\mu \rightarrow \infty$ on a $G(s) \rightarrow -s$ qui représente bien un dérivateur idéal.

Le but de ce circuit est donc de remplacer, dans le modèle bond graph, tout élément de stockage en causalité dérivée par le modèle du dérivateur. Ainsi, pour tout élément C en causalité dérivée est substitué un modèle bond graph en causalité intégrale permettant de faire des simulations comme le montre la figure 2.20.

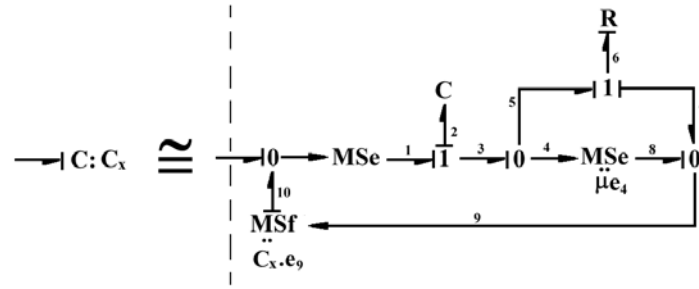


FIG. 2.20 – Représentation d'un élément C en causalité dérivée par un dérivateur analogique.

La contre-réaction à travers la source modulée de la figure 2.20 peut être aussi représentée par un élément gratateur (GY) avec activation des liens [6].

2.8.2.5 Résultats de simulation

La simulation des scénarios de défaillances du système des trois réservoirs a été réalisée en utilisant Matlab-Simulink. Ce dernier permet de résoudre des équations algèbro-différentielles, d'effectuer des traitements sur les signaux résidus, de gérer différents modes de fonctionnement en utilisant les organigrammes d'état (state flow charts), de migrer vers des implémentations en temps réel, etc..

De la même façon que pour le modèle bond graph de diagnostic, le bloc diagramme correspondant au modèle bond graph en causalité intégrale du système a été généré et implémenté sous Simulink (figure 2.21).

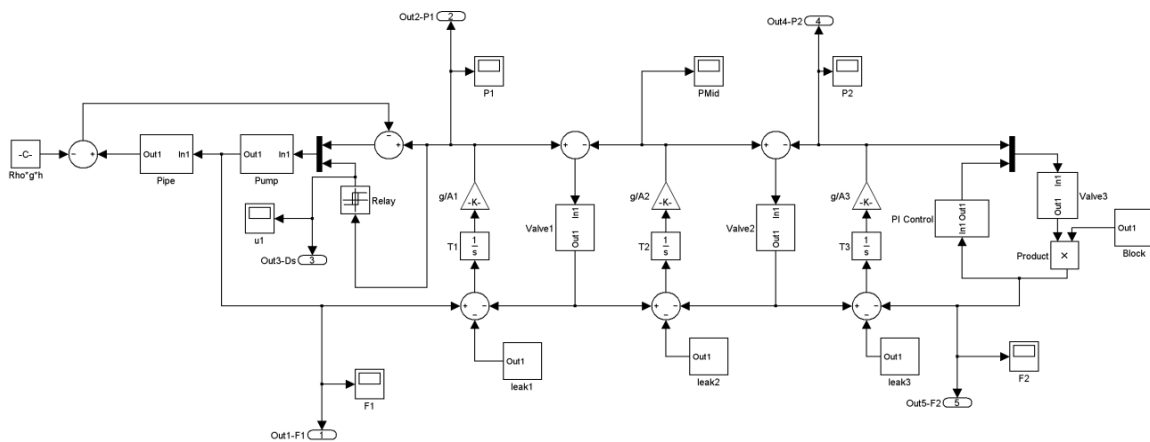


FIG. 2.21 – Bloc diagramme du modèle bond graph en causalité intégrale.

Pour tenir compte des bruits et des incertitudes des capteurs, du bruit est ajouté aux sorties du modèle en causalité intégrale. L'amplitude du bruit est de 20% de la valeur de

sortie de chaque capteur. Les défaillances pouvant affecter les capteurs sont représentées par des perturbations sur les sorties du modèle en causalité intégrale. Ces perturbations peuvent être des valeurs non nulles ajoutées aux sorties pour les faire dévier de leurs valeurs de fonctionnement normal.

Sur le modèle de simulation, on considère que les vannes ont une caractéristique linéaire; i.e. $C_d(x) = C_d \cdot x$. Pour la pompe, le débit Q est donné par une fonction linéaire de la différence de pression ΔP :

$$Q = \lambda_1 \cdot \Delta P + \lambda_2,$$

où λ_1 et λ_2 sont deux paramètres connus avec $\lambda_1 < 0$. Les valeurs des paramètres utilisées dans la simulation sont données au tableau 2.6

Les conditions initiales des états du modèle bond graph en causalité intégrale sont considérées nulles. Lors de la détection de la présence de boucles algébriques et différentielles, le programme Simulink génère des avertissements mais pas des erreurs. Mais avant de procéder à la simulation des différents scénarios de défaillance, il est important de valider le modèle du système car comme on l'a vu dans la première partie de ce chapitre la qualité de la surveillance dépend fortement de celle du modèle. Pour cet exemple, la réponse du modèle (bond graph en causalité intégrale) en fonctionnement nominal est donnée par la figure 2.22. Sur cette figure, on peut constater les courbes correspondant aux pressions dans les trois réservoirs: P_1 et P_2 correspondent respectivement aux pressions au fond des réservoirs T_1 et T_3 ; et P_{mid} est la pression estimée au fond du réservoir T_2 .

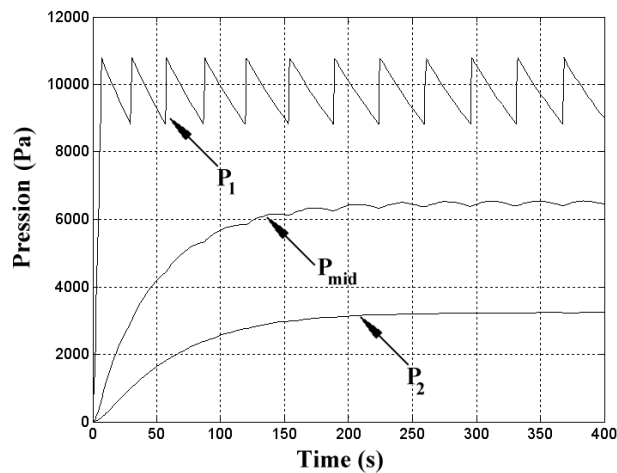


FIG. 2.22 – La réponse du système des trois réservoirs.

Un autre point important dans un système de surveillance est le choix des seuils sur les

Paramètre	Description	Valeur	Unité
A_1	Section du réservoir T1	0.1	m^2
A_2	Section du réservoir T2	0.05	m^2
A_3	Section du réservoir T3	0.04	m^2
C_d	Coefficient de décharge de la conduite	0.05	$\sqrt{kg.m}$
C_{d1}	Coefficient de décharge de la vanne V1	0.01	$\sqrt{kg.m}$
C_{d2}	Coefficient de décharge de la vanne V2	0.01	$\sqrt{kg.m}$
C_{d3}	Coefficient de décharge de la vanne V3, lorsqu'elle est totalement ouverte	0.01	$\sqrt{kg.m}$
λ_1	Paramètre de la pompe	-0.001	$m.s$
λ_2	Paramètre de la pompe	110	$Kg.s^{-1}$
h	Longueur de la conduite	10	m
L_s	Consigne de niveau pour le régulateur LC	1	m
F_s	Consigne de débit pour le PI	0.1	$Kg.s^{-1}$
K_p	Gain de la proportionnelle du PI	1	–
K_i	Gain de l'action intégrale du PI	0.01	–
ρ	Masse volumique du fluide	1000	$Kg.m^3$
g	Constante de gravité	9.81	$m.s^2$

TAB. 2.6 – Valeurs des paramètres du système des trois réservoirs

résidus et au delà desquels une alarme est déclenchée. Pratiquement, ces seuils (tableau 2.7) sont déterminés sur la base d'observations durant le fonctionnement normal du système.

ε_1	ε_2	ε_3	ε_4	ε_5
0.02	0.1	0.1	0.1	0.02

TAB. 2.7 – Seuils des résidus

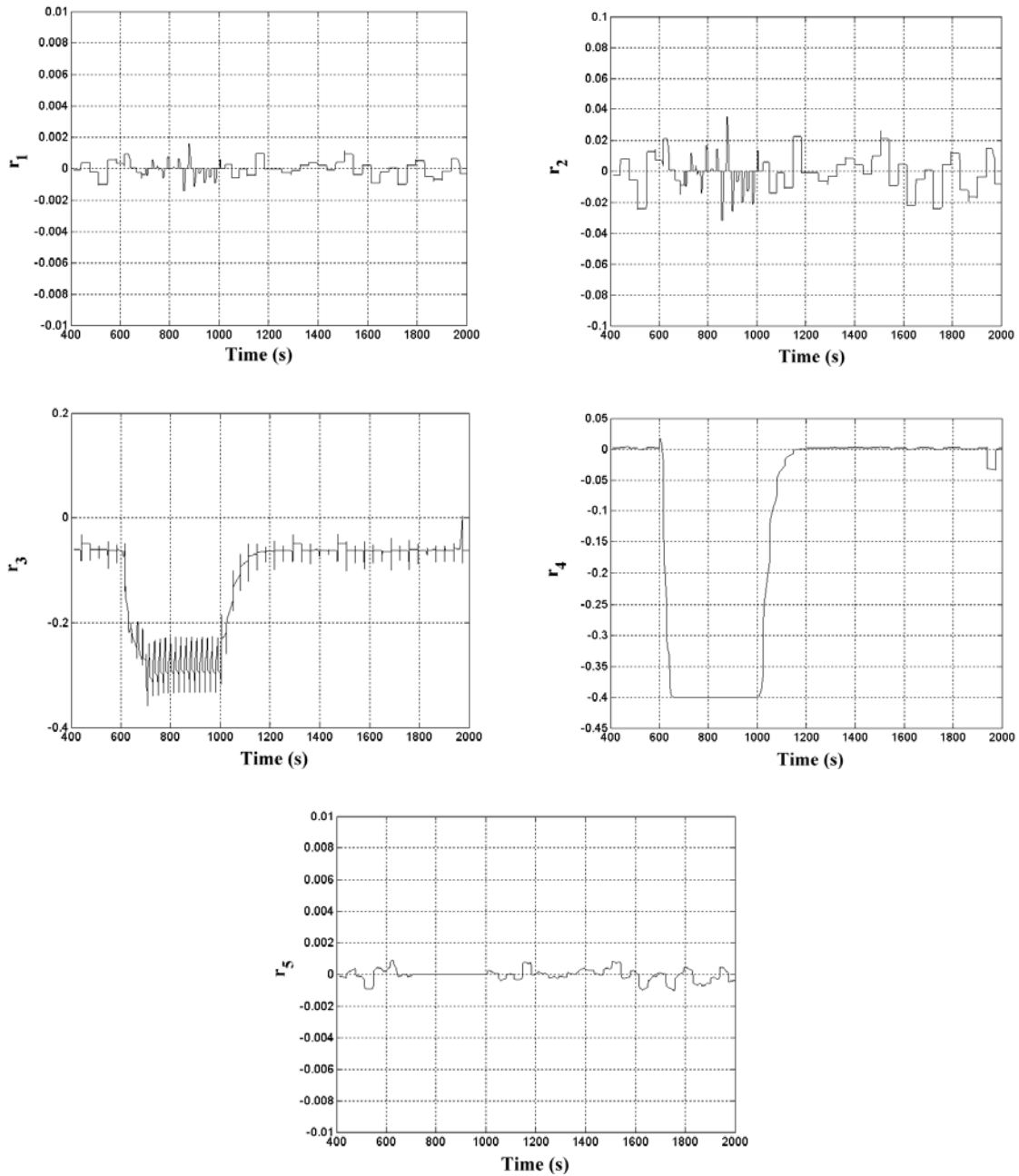


FIG. 2.23 – Réponse des résidus à la fuite dans T_2

Deux scénarios de défaillance ont été simulés. Le premier consiste en une fuite d'eau du réservoir T_2 durant un intervalle de temps allant de 600 à 1000 s. La figure 2.23 montre la réponse des résidus r_3 et r_4 et leur sensibilité à cette fuite. Si on se réfère à la signature du composant T_2 donnée au tableau 2.5 on constate que ce résultat est conforme à ce qui est prévu; c'est à dire qu'en cas de défaillance du composant T_2 seuls les résidus r_3 et r_4

excéderont leurs seuils respectifs. Le vecteur de cohérence résultant est donc $V = [0 \ 0 \ 1 \ 1 \ 0]$. Ce dernier se confond avec plusieurs vecteurs de signature dans la matrice du tableau 2.5 et par conséquent, la fuite dans T_2 est seulement détectable mais pas localisable.

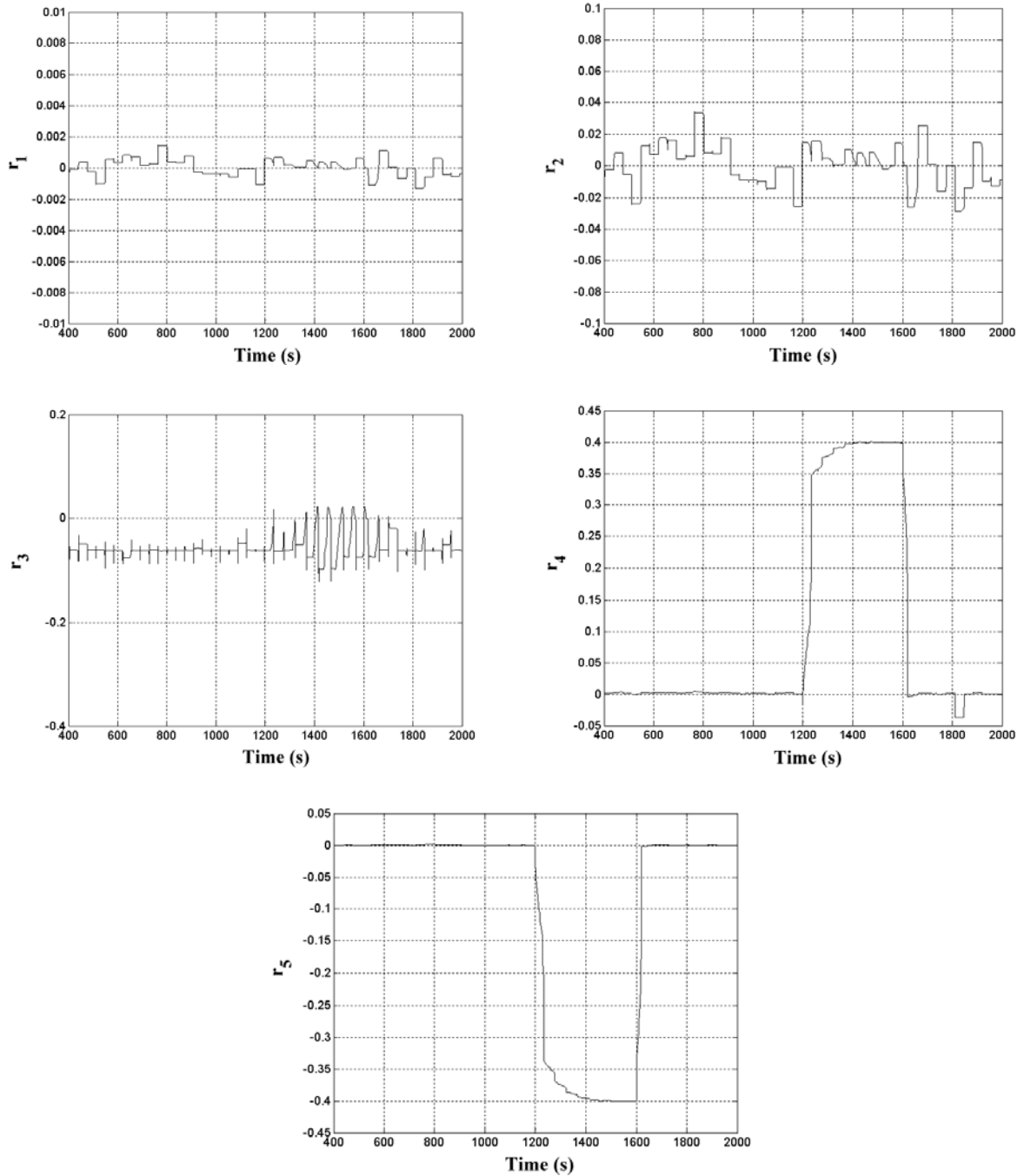


FIG. 2.24 – Réponse des résidus au blocage de V_3

Le deuxième scénario concerne un blocage partiel (à 50%) de la vanne V_3 pendant l'in-

tervalle de temps allant de 1200 à 1600 *s*. Les réponses des résidus à cette défaillance sont données à la figure 2.24. Le vecteur de cohérence résultant, $V = [0,0,0,1,1]$, possède plusieurs correspondances sur la matrice des signatures, et donc, cette défaillance aussi ne peut pas être localisable mais seulement détectable.

Dans les résultats de simulation présentés ici, une moyenne glissante d'une durée de 10 *s* a été appliquée sur tous les résidus pour éliminer l'effet du bruit.

2.9 Conclusion

La méthode proposée dans cette deuxième partie du chapitre améliore la disponibilité de l'information pour un système d'aide à la décision en générant davantage de résidus structurellement indépendants sous forme numérique. Cela est primordial dans les cas où la forme symbolique des résidus est impossible à avoir à cause de la présence de différents types de boucles dans les équations du modèle analytique du processus. L'algorithme pour la génération des résidus numériques est directement obtenu à partir du modèle bond graph du système en utilisant un ensemble de modifications. Ces dernières conduisent à une structure de graphe appelée modèle Bond Graph de Diagnostic (BGDg) dans lequel les données issues des capteurs sont les entrées et les résidus sont des sorties. L'analyse des chemins causaux vers les résidus dans le modèle BGDg est ensuite utilisée pour construire la matrice de signature des défaillances. Le modèle de diagnostic peut être utile pour tester les performances de la procédure de surveillance à travers les données de simulation et celles récoltées en hors ligne durant le fonctionnement du processus. La méthode développée a été appliquée sur un système commandé de trois réservoirs et des simulations ont été réalisées.

Enfin, il est important de souligner que la méthode proposée dans cette partie du chapitre peut être appliquée sur des modèles comportant différentes formes de boucles non solvables mais aussi sur des modèles réguliers sans boucles.

Troisième partie

Bond Graph de Diagnostic Étendu

2.10 Bond Graph de Diagnostic Étendu

En inversant la causalité des détecteurs dans un bond graph de diagnostic, il arrive parfois que certains éléments de stockage restent en causalité intégrale préférentielle. Dans une méthode classique de génération de RRAs, on peut introduire des dérivées pour éviter le problème des conditions initiales des éléments en causalité intégrale; ce qui est possible et valide puisque la dérivée d'une RRA est aussi une RRA.

Prenons l'exemple d'un système masse-ressort donné à la figure 2.25(a) et dont le modèle bond graph correspondant est montré à la figure 2.25(b). Le modèle bond graph de diagnostic de ce système est celui donné par la figure 2.25(c).

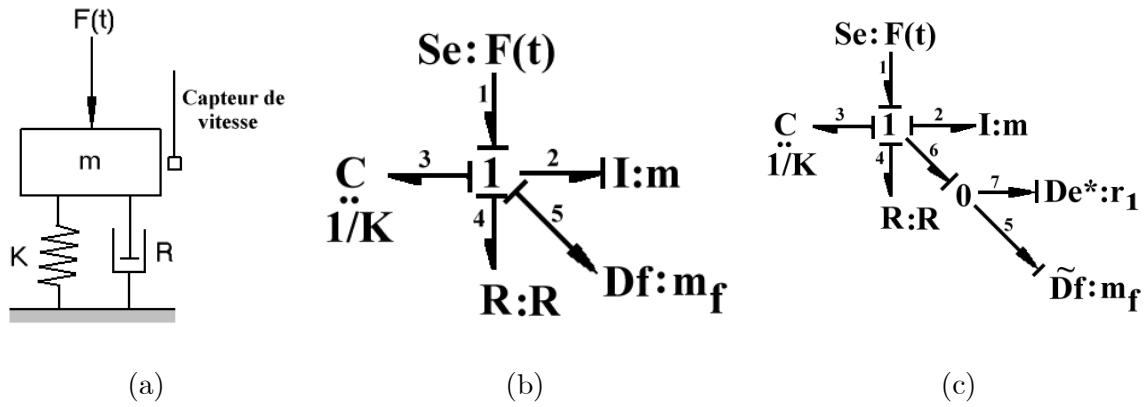


FIG. 2.25 – *Système mécanique et son modèle bond graph*

A partir du modèle BGDg de la figure 2.25(c), l'unique RRA possible à générer, correspondant au résidu r_1 , peut être donnée par la relation suivante:

$$RRA_1 : F(t) - m.\dot{m}_f - R.m_f - K. \int m_f .dt = 0.$$

Pour éviter la forme intégrale sur la variable m_f , on peut dériver la RRA pour obtenir:

$$RRA_1 : \dot{F}(t) - m.\ddot{m}_f - R.\dot{m}_f - K.m_f = 0.$$

La dérivation de la RRA était alors nécessaire (pour éviter les conditions initiales) à cause de la présence d'élément de stockage $C : \frac{1}{K}$ en causalité intégrale. Ceci peut être directement représenté sur le modèle bond graph en respectant la règle suivante:

- Lorsqu'on affecte la causalité aux détecteurs, le modèle bond graph global doit correspondre à un système surdéterminé de telle sorte que les chemins causaux à partir de tous les éléments de stockage doivent donner sur des sommets externes connus (sources ou mesures). Une forme de causalité dérivée telle que celle de la figure 2.26(a) est donc

considérée comme non-valide à cause de l'existence d'une boucle différentielle entre deux éléments de stockage I et C .

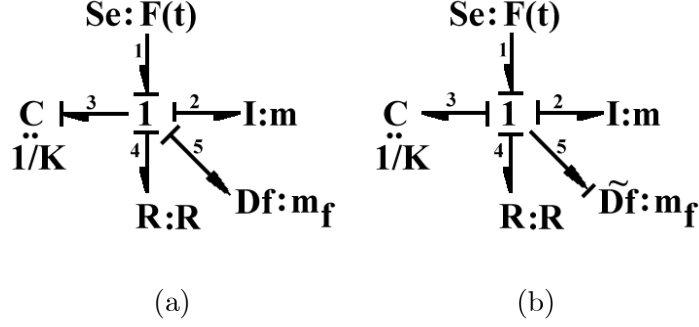


FIG. 2.26 – (a) Bond graph présentant une boucle (b) élimination de la boucle par inversion de la causalité du détecteur

Sur le modèle bond graph de la figure 2.26(b), tous les chemins causaux à partir des éléments I et C arrivent sur des détecteurs (mesures). Ainsi, à partir du modèle de la figure 2.26(a), pour la jonction 1 on peut écrire:

$$\begin{aligned}
 e_1 &= F(t), \\
 e_2 &= m \cdot \dot{f}_3, \\
 e_4 &= R \cdot f_3, \\
 e_5 &= 0, \\
 e_3 &= e_1 - e_2 - e_4 - e_5, \\
 f_1 &= f_3, \\
 f_2 &= f_3, \quad f_3 = \frac{1}{K} \cdot \dot{e}_3, \\
 f_4 &= f_3, \\
 f_5 &= f_3 \text{ ou } m_f = f_3.
 \end{aligned} \tag{2.42}$$

On a donc un système de neuf équations à huit inconnues qui permet de générer une RRA.

Mais, comment élimine-t-on les variables inconnues, autrement dit: comment pose-t-on la causalité (i.e. déterminer les variables inconnues à partir des variables connues) pour générer les RRAs?

Premièrement, on localise toutes les boucles présentes sur le système d'équation. Ensuite,

à chaque fois qu'une variable connue est dans le côté gauche d'une équation:

- on réécrit l'équation si et seulement si cette dernière contient, dans son côté droit, une seule variable inconnue calculable.

Cela veut dire que si on a $x_k = \Phi(x_i, x_j)$ avec x_i et x_k connus, peut-on alors déterminer x_j ? Par exemple, pour la dernière équation du système 2.42, on peut dans une première étape (1) écrire:

$$f_3 = m_f,$$

qui conduit aux réductions successives pour finalement, dans une seconde étape (2), obtenir:

$$m_f = \frac{1}{K} \cdot \frac{d}{dt} (F(t) - R \cdot m_f - m \cdot \dot{m}_f),$$

ou bien

$$\dot{F}(t) - R \cdot \dot{m}_f - m \cdot \ddot{m}_f - K \cdot m_f = 0.$$

Cette relation est la même que celle générée à partir d'un modèle BGDg, sauf qu'ici on n'a pas à apporter de dérivée. A noter que l'étape (1) est relative à un problème d'affectation de la causalité et que l'étape (2) n'est possible que si l'ensemble des équations est régulier.

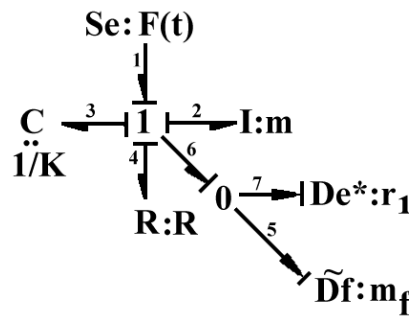


FIG. 2.27 – Modèle BGDg et le résidu correspondant

La méthode basée sur le modèle BGDg et qui est applicable sur la plupart des systèmes doit donc être étendue. A partir de la figure 2.27, on peut écrire tous les chemins causaux vers le détecteur de résidu r_1 à partir des sommets externes (sources et mesures). L'ordre d'un chemin causal est incrémenté lorsqu'il traverse un élément de stockage en causalité intégrale. Dans le cas contraire (élément de stockage en causalité dérivée), l'ordre du chemin causal est décrémenté. Ici, l'ordre est montré en exposant.

$$\begin{aligned}
e_1^0 &\rightarrow e_6^0 \rightarrow e_7^0, \\
f_5^0 &\rightarrow f_6^0 \rightarrow f_2^0 \rightarrow e_2^{-1} \rightarrow e_6^{-1} \rightarrow e_7^{-1}, \\
f_5^0 &\rightarrow f_6^0 \rightarrow f_3^0 \rightarrow e_3^1 \rightarrow e_6^1 \rightarrow e_7^1, \\
f_5^0 &\rightarrow f_6^0 \rightarrow f_4^0 \rightarrow e_4^0 \rightarrow e_6^0 \rightarrow e_7^0.
\end{aligned}$$

On a donc:

$$r_i = \sum_{-1}^1 \Phi_i(m_f, F(t)),$$

où Φ_i est la fonction correspondant à l'ordre dérivée.

Naturellement, dans une RRA l'ordre le plus élevé doit être égal à zéro. Dans le cas contraire, on doit intervenir par des dérivées successives sur la RRA, pour ramener l'ordre à zéro. Ces résultats peuvent alors être formulés comme suit:

1. Soit p_{ij} le nombre de chemins causaux du $i^{\text{ième}}$ sommet externe (source ou détecteur) vers le $j^{\text{ième}}$ détecteur de résidu. Soit m_{ij} l'ordre du chemin causal.
2. Soit $n_j = \max(m_{ij})$, pour le $j^{\text{ième}}$ détecteur de résidu,
3. Si $n_j > 0$, alors au moins n_j dérivées sont nécessaires sur le détecteur de résidu pour que la relation obtenue soit indépendante des conditions initiales.
4. Si $n_j = 0$, alors le résidu obtenu est prêt pour l'implémentation.

Ces propositions peuvent être représentées par un modèle bond graph de diagnostic étendu comme celui de la figure 2.28.

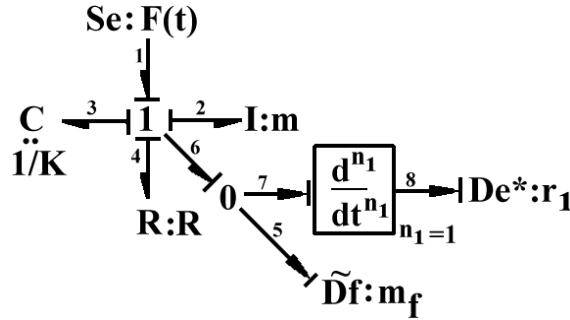


FIG. 2.28 – *Modèle Bond Graph de Diagnostic Étendu.*

Il est important de noter que le bond graph de diagnostic étendu n'est nécessaire que si on est en présence d'un système structurellement non-observable.

2.10.1 Choix de la causalité

Considérons maintenant le système de deux réservoirs donné à la figure 2.29(a) et ses deux modèles bond graphs correspondant à deux affectations de causalité dérivée (figure 2.29(b) et

(c).

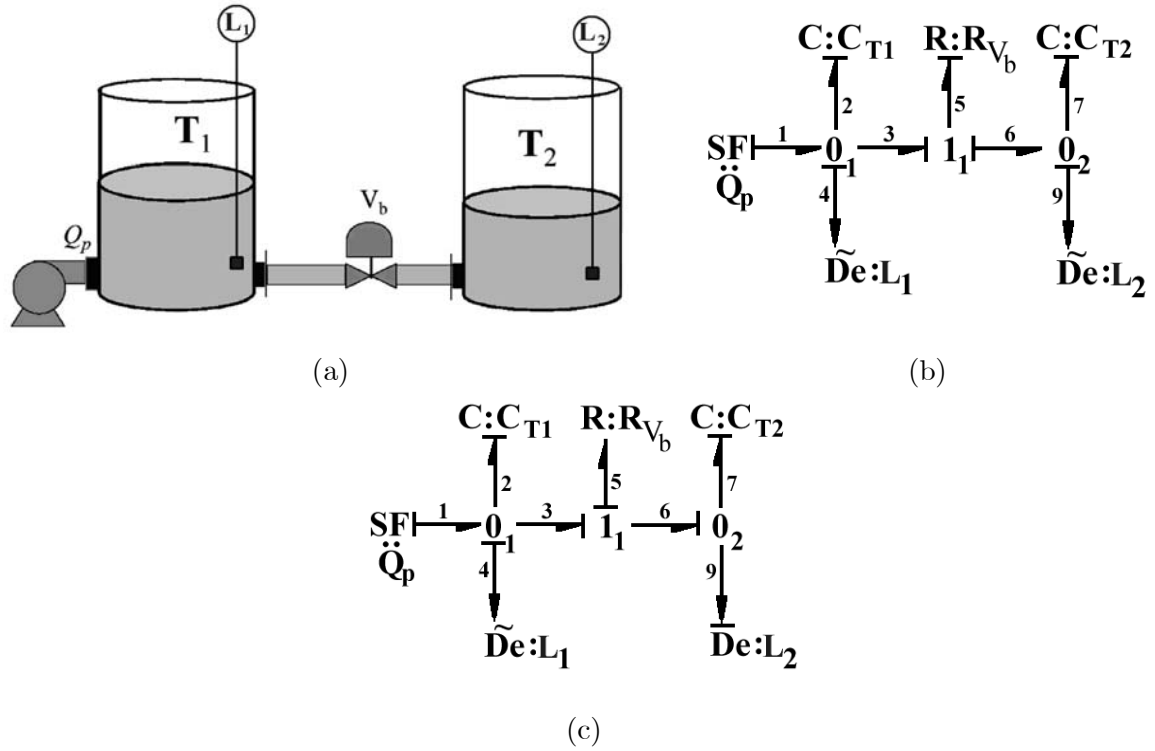


FIG. 2.29 – Affection de la causalité sur un BGDg étendu

En supposant que la vanne V_b est à tout moment ouverte, on peut déduire que le système est observable avec seulement un capteur de niveau. Mais dans le cas où la vanne V_b reste fermée, on aura besoin de deux capteurs de niveau pour assurer l'observabilité du système. Dans ce dernier cas, les RRAs générées à partir de la figure 2.29(b) sont:

$$\begin{aligned} RRA_1 : Q_p - \rho \cdot A_1 \cdot \frac{dL_1}{dt} - C_d \cdot \sqrt{L_1 - L_2} &= 0, \\ RRA_2 : C_d \cdot \sqrt{L_1 - L_2} - \rho \cdot A_2 \cdot \frac{dL_2}{dt} &= 0. \end{aligned} \quad (2.43)$$

Une fois que la boucle algébrique existant entre les éléments R et C du modèle de la figure 2.29(c) est résolue, on peut générer les deux RRAs suivantes:

$$\begin{aligned} RRA_1 : Q_p + \rho \cdot A_1 \cdot \frac{dL_1}{dt} - \rho \cdot A_2 \cdot \frac{d}{dt} \left(L_1 - \frac{\left(Q_p - \rho \cdot A_1 \cdot \frac{dL_1}{dt} \right)^2}{C_d^2} \right) &= 0, \\ RRA_2 : L_2 - L_1 + \frac{\left(Q_p - \rho \cdot A_1 \cdot \frac{dL_1}{dt} \right)^2}{C_d^2} &= 0. \end{aligned} \quad (2.44)$$

On peut facilement constater que les RRAs de l'Eq. (2.44) présentent des singularités dans le cas où $C_d = 0$ (la vanne est complètement fermée). Pour éviter cela, on peut multiplier les deux côtés de chaque RRA par C_d^2 . Lorsque $C_d = 0$, les deux RRAs de l'Eq. (2.44) deviennent alors:

$$\begin{aligned} RRA_1 : Q_p - \rho \cdot A_1 \cdot \frac{dL_1}{dt} &= 0, \\ RRA_2 : Q_p - \rho \cdot A_1 \cdot \frac{dL_1}{dt} &= 0; \end{aligned} \quad (2.45)$$

qui sont structurellement dépendantes pour cet état hybride. Cependant, pour $C_d = 0$, les RRAs générées à partir du modèle bond graph de la figure 2.29(a) (Eq.(2.43)) deviennent:

$$\begin{aligned} RRA_1 : Q_p - \rho \cdot A_1 \cdot \frac{dL_1}{dt} &= 0, \\ RRA_2 : \rho \cdot A_2 \cdot \frac{dL_2}{dt} &= 0; \end{aligned} \quad (2.46)$$

qui sont structurellement indépendantes et valides pour tous les états hybrides du système. On peut donc formuler la proposition suivante:

Proposition 4 *Le choix de l'affectation de la causalité doit être fait de telle sorte que le système reste observable sur l'ensemble de ses états hybrides.*

2.10.2 Problèmes de couplage entre variables en bond graph

Supposons que le système des deux réservoirs possède seulement un détecteur de flux comme le montre la figure 2.30(a). Le modèle bond graph en causalité dérivée du système est donné à la figure 2.30(b). Le système est structurellement non-observable. En effet, avec un capteur de débit, on peut déterminer la différence entre les deux niveaux mais on ne peut pas calculer séparément la valeur de chacun d'eux.

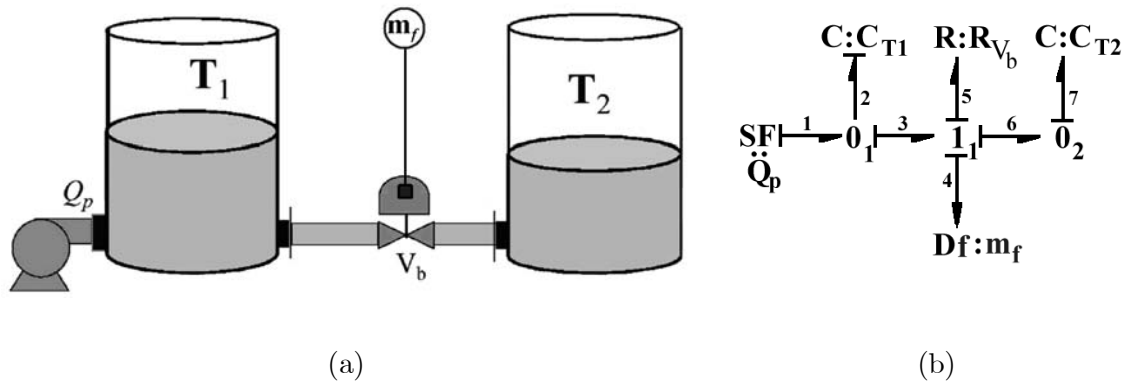


FIG. 2.30 – Système de deux réservoirs avec une seul détecteur de flux

Sur le modèle bond graph de la figure 2.30(b) il n'y a pas d'autre moyen d'affecter la causalité de telle sorte que tous les chemins causaux à partir des éléments de stockage donnent sur des sommets externes (sources ou détecteurs). Toutefois, comme on l'a déjà démontré, même si le système n'est pas totalement observable on peut générer des RRA (une RRA dans ce cas).

Pour que tous les chemins causaux donnent sur des sommets externes, il faut que les éléments de stockage restent en causalité intégrale comme le montrent les figures 2.31(a) et 2.31(b).

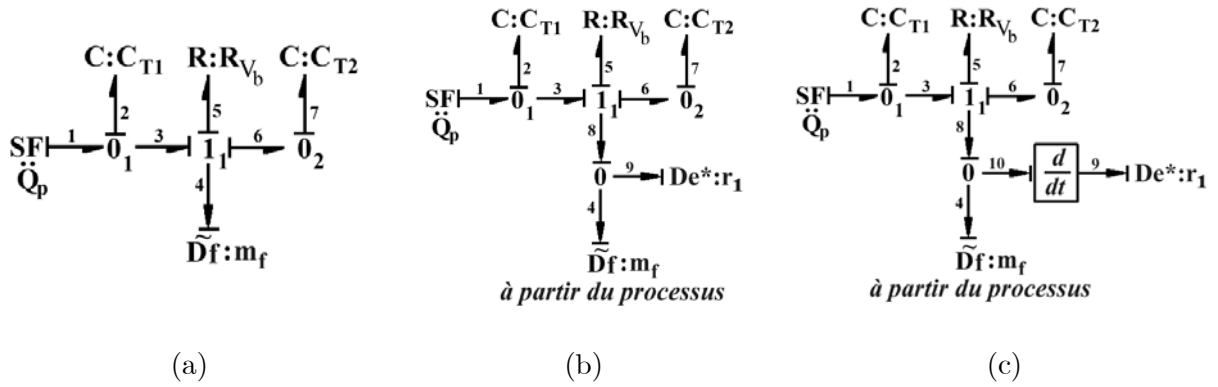


FIG. 2.31 – Génération de résidus à partir d'un modèle bond graph dont les éléments de stockage sont en causalité intégrale

De la figure 2.31(b), la seule RRA qu'on peut obtenir est la suivante:

$$\frac{1}{C_{T1}} \cdot \int (Q_p - m_f) \cdot dt - \frac{m_f^2}{C_d^2} - \frac{1}{C_{T2}} \cdot \int m_f \cdot dt = 0.$$

Puisque l'ordre maximal des chemins causaux pour arriver au détecteur de résidu $De^* : r_1$ à partir des variables connues est égal à 1, on doit procéder à une dérivée pour obtenir une RRA valide (figure 2.31(c)). La nouvelle RRA obtenue est donc:

$$\frac{1}{C_{T1}} \cdot (Q_p - m_f) - \frac{2}{C_d^2} \cdot m_f \cdot \frac{dm_f}{dt} - \frac{1}{C_{T2}} \cdot m_f = 0.$$

En éliminant la singularité causée par $C_d = 0$, on obtient

$$\frac{C_d^2}{C_{T1}} \cdot (Q_p - m_f) - 2 \cdot m_f \cdot \frac{dm_f}{dt} - \frac{C_d^2}{C_{T2}} \cdot m_f = 0.$$

Lorsque $C_d = 0$, l'expression précédente devient $m_f \cdot \frac{dm_f}{dt} = 0$. Cette relation est satisfaite non seulement pour $m_f = 0$ (débit nul) mais aussi lorsque le débit est constant ($m_f = c^{ste}$) qui peut être du à une défaillance du capteur de débit. Cette situation nous induit donc à une erreur.

On peut alors se poser la question suivante: comment doit-on affecter la causalité de telle sorte que les singularités soient automatiquement évitées? Malheureusement, la réponse est que cela n'est pas possible sur un modèle bond graph. Par exemple, si on affecte la causalité conductance à l'élément R_{V_6} de la figure 2.31(a) les chemins causaux ne vont pas donner sur des sommets externes connues.

Sur un modèle bond graph, lorsque la causalité d'un lien est choisie pour un chemin donné, e.g. la variable $x \in [e,f]$, alors la causalité de la variable complémentaire, i.e. $y \in [e,f] - x$, est automatiquement affectée. Dans un modèle bond graph correct, la causalité sur les variables ne peut pas être affectée indépendamment. La détermination automatique d'une information (e.g. flux) dans un chemin causal à partir de l'information complémentaire (dans ce cas l'effort) peut ne pas être souhaitable pour la génération des résidus. Mais, ceci est une propriété des modèles bond graphs. Ainsi, lorsque l'utilisateur construit son modèle bond graph pour une application donnée; s'il choisit de commencer par exprimer les équations du bilan des variables effort il aura automatiquement les équations du bilan des variables flux. C'est pourquoi l'outil bond graph est souhaitable pour modéliser les systèmes physiques en général et plus particulièrement les systèmes où différents domaines d'énergie sont présents.

Pour conclure, un travail de recherche plus approfondi est nécessaire pour rendre flexible la manipulation de la causalité sur les modèles bond graphs; plus particulièrement pour des applications de surveillance (élimination des variables inconnues).

Chapitre 3

Informatisation des procédures développées

3.1 Introduction

Dans les chapitres précédents, nous avons développé une approche systématique pour: construire les modèles bond graphs des processus thermofluides, classifier les différents composants utilisés, vérifier leur interconnexion et aussi des algorithmes pour générer les RRAs et la matrice de signature des défaillances à partir d'un modèle bond graph. Réaliser toutes ces tâches manuellement est un travail fastidieux. De plus, lorsqu'on a besoin d'apporter des modifications à l'architecture du processus, comme par exemple supprimer ou ajouter des composants ou modifier le placement des capteurs, la structure du modèle subit de profonds changements et toutes les manipulations doivent être refaites à nouveau. Pour une manipulation optimale de la modélisation des processus thermofluides, que ce soit pour obtenir un modèle de simulation ou pour le diagnostic, l'utilisateur a donc besoin d'un outil (de préférence graphique) performant. Cet outil doit être rapide dans la génération du modèle, capable de faire abstraction sur les équations mathématiques souvent compliquées, et simple d'utilisation. L'outil proposé doit aussi produire une sortie qui peut être facilement interfaçable avec d'autres plateformes auxiliaires.

Ce chapitre présente la boîte à outils ModelBuilder développée dans le cadre du projet CHEM et qui répond aux exigences citées ci-dessus. Ce travail a été publié dans [19].

3.2 État de l'art des logiciels existants

Il existe plusieurs logiciels commercialisés qui traitent de la modélisation des processus en génie des procédés. Dans ce qui suit, nous présenterons les plus connus et les plus utilisés

d'entre eux.

- **ProSimPlus** est un outil d'ingénierie de procédés qui effectue des bilans de matière et d'énergie en régime permanent pour un large éventail de procédés industriels. Il est utilisé aussi bien en conception qu'en exploitation pour l'optimisation de procédés, le dégoulotage d'unités, le revamping ou encore les études de faisabilité et permet de représenter fidèlement le comportement des procédés de fabrication. ProSimPlus a de nombreuses applications dans les industries chimiques, pharmaceutiques, de la chimie fine, pétrochimiques, du traitement du gaz, du raffinage ainsi que dans les sociétés d'ingénierie. Sa richesse thermodynamique et son architecture permettent la représentation de la quasi-totalité des systèmes rencontrés. Plusieurs versions du logiciel Prosim sont disponible: Prosim, ProsimPlus, ProsimCMS et WebProsim.
- **Quick Hydraulics** est un logiciel utilisé dans la simulation des procédés chimiques (flowsheeting), et pour établir les bilans de masse et d'énergie. Quick Hydraulics offre une multitude d'applications pour la conception des processus. Pour les besoins des calculs, Quick Hydraulics utilise une base de données intégrée comportant un peu plus de 800 composants. L'écoulement dans les conduites peut être considéré comme un flux mono-phasique ou biphasique, avec ou sans pertes de charges, etc.. Le dimensionnement des composants et les calculs d'isolation peuvent être pris en compte avec cet outil. Le module de simulation propose plusieurs composants: pompes, conduites, coudes et tés, réducteurs, vannes, chauffage, compresseurs, échangeurs de chaleurs, etc..
- **VisiMix2000** permet aux ingénieurs de simuler toutes sortes de procédés tel que l'échange de chaleur et de masse et les réactions chimiques. Le logiciel est facile d'utilisation et accessible à tout ingénieur en chimie ou en génie des procédés. Il lui suffit alors de sélectionner les composants à utiliser avec les paramètres adéquats. En plus du calcul des paramètres du processus, VisiMix2000 analyse les données reçues et génère des avertissements dans le cas où une ou plusieurs de ces données n'est pas acceptable ou non valide du point de vue fonctionnement ou sûreté. VisiMix2000 informe aussi l'opérateur des zones de stagnations, des surcharges des actionneurs, etc..
- **Aspen Plus** est un logiciel développé par AspenTech. C'est un logiciel intégré composé de différents outils pour la simulation des processus en génie des procédés. Aspen HYSYS est un outil de modélisation des procédés pour des simulations en régime permanent, la conception, la surveillance, l'optimisation et la planification pour la production pétrolière. Aspen COMThermo est un composant important dans le logiciel, qui permet d'introduire les propriétés thermodynamiques des matériaux utilisés dans le modèle.
- **gPROMS** (pour general PROcess Modelling System) est un logiciel de modélisation, d'optimisation et de simulation des processus. Les modèles de gPROMS peuvent être utilisés pour les simulations, l'optimisation et pour l'estimation en utilisant l'interface

graphique de gPROMS ou l'un des outils intégrés à gPROMS tel que FLUENT, Aspen Plus, Hysys ou Matlab-Simulink. Le noyau de gPROMS est un système à base d'équations, i.e. les processus sont décrits par les relations physico-chimiques qui les caractérisent. gPROMS analyse les équations et propose un choix (régime établi ou dynamique) à l'utilisateur pour la simulation, l'optimisation ou l'estimation des paramètres.

- **INDISS** est une plateforme de simulation de RSI, une filiale du groupe IFP (Institut Français du Pétrole). INDISS est un outil de simulation des processus dynamiques qui met à disposition de l'utilisateur une riche bibliothèque de composants en génie des procédés pour construire les modèles dynamiques. INDISS est capable de traiter des systèmes complexes en utilisant des outils de calcul matriciel. Il utilise une résolution globale pour les pressions et les débits et une approche modulaire et séquentielle pour les simulations dynamiques. Plusieurs étapes de calcul sont réalisées en une unité de temps. Une étape se focalise sur le calcul relatif au phénomène hydraulique et le bilan de masse tandis que d'autres étapes de calcul traitent du bilan d'énergie et des calculs thermodynamiques. INDISS a été conçu sur une architecture de composants multicouches qui permet une intégration facile de composants externes tel que les modèles thermodynamiques ou chimiques.

En plus des logiciels cités, plusieurs autres sont disponibles dans le commerce, e.g. Design II sous Windows, Super Pro, HySIM, Super Target, etc..

Les logiciels à base de l'outil bond graph tel que Symbols 2000 ou 20-Sim sont capables de créer des modèles en génie des procédés par une simple interconnexion de sous-modèles. Ils intègrent tous les deux des bibliothèques de composants hydrauliques et thermiques. AmeSim d'Imagine est un autre outil de modélisation par bond graph et qui peut être utilisé pour des systèmes en génie des procédés. Un autre outil de modélisation par bond graph, HybrSim, s'avère très efficace pour la modélisation et la simulation des systèmes hybrides. Une particularité des logiciels à base des bond graphs est qu'ils peuvent être utilisés pour la commande des systèmes.

Dans le domaine de la surveillance, le graph causal temporel et les "Tree Graphs" par exemple sont utilisés pour la détection des défaillances des systèmes physiques en régime établi. Toutefois, il n'existe pas d'outil logiciel pour construire les "Tree Graphs". Ces derniers doivent être générés manuellement à partir du modèle bond graph et ensuite injectés dans un logiciel spécifique pour décider de la présence ou non de défaillances. Toujours dans la catégorie des outils logiciels par bond graphs, "DecisionTools" offre plusieurs boîtes à outils pour l'analyse des systèmes physiques. L'une d'elle, @RISK, est utilisée avec Excel pour l'analyse des risques et pour la simulation. "PrecisionTree" est une autre boîte à outils de décision qu'on peut intégrer dans Excel et qui permet de créer des diagrammes d'influence pour exprimer les relations entre les composants du système. Enfin, FaultrEASE est un programme

qui permet de créer, éditer et manipuler des équations relatives aux arbres de défaillances pour la surveillance des processus.

Un des inconvénients des logiciels (traitant de l'outil bond graphs) cités ci-dessus est qu'ils sont conçus pour répondre aux besoins de simulation. Les expressions mathématiques du modèle global sont alors obtenues en liant les équations des sous-modèles correspondants des composants physiques. Du point de vue bond graph, les variables associées aux ports d'entrée et de sortie au point d'interconnexion de deux composants sont mises à égalité pour construire le modèle dynamique global. On parle d'une modélisation à causalité fixe qui convient mieux à la simulation.

Cependant, pour la surveillance des processus, on préfère une causalité flexible. Cette flexibilité permet d'avoir plusieurs couplages possibles entre les variables et facilite l'élimination des variables inconnues. Dans ce cas les sous-modèles doivent être représentés sous forme d'équations acausales. Toutefois, il est difficile de lier des modèles acausaux dans une interface graphique à cause des difficultés pour formuler les règles d'interconnexion entre objets.

Dans ce chapitre est présentée la boîte à outils ModelBuilder qui est le résultat informatique des procédures développées dans les deux premiers chapitres. Elle nous aide à obtenir le modèle dynamique du processus et permet de générer les RRAs et la matrice de signature des défaillances à partir du P&ID en utilisant une interconnexion graphique et acausale.

3.3 Représentation du P&ID par bond graph

L'une des caractéristiques importante des systèmes physiques est l'interaction passive (par opposition à l'interaction active) entre les éléments qui les composent. Les systèmes en génie des procédés sont des processus multi-énergies (plusieurs couplages: hydrauliques, thermiques, électriques et mécaniques) qui mettent en jeu les deux types d'interaction. L'outil bond graph s'avère alors être utile pour la modélisation de ces systèmes.

Pour montrer comment passer d'un P&ID d'un système donné à la représentation bond graph correspondante, nous avons choisi un exemple de processus thermofluide donné par la figure 3.1(a). Le système consiste en un réservoir alimenté en eau débitée par une pompe. L'eau est chauffée par une résistance thermique et est fournie à un utilisateur en agissant sur la vanne. Les capteurs P et T mesurent respectivement la pression et la température dans le réservoir. Le capteur F est quant à lui utilisé pour mesurer la valeur du débit d'eau fournie au consommateur.

Le modèle bond graph correspondant au processus est donné à la figure 3.1(b). Sur cette figure, les lignes en pointillés sont utilisées pour séparer les différents composants et les liens avec des petits cercles représentent des liens bond graphs vectoriels. La représentation iconique du modèle bond graph de la figure 3.1(b) est donnée par la figure 3.2(a).

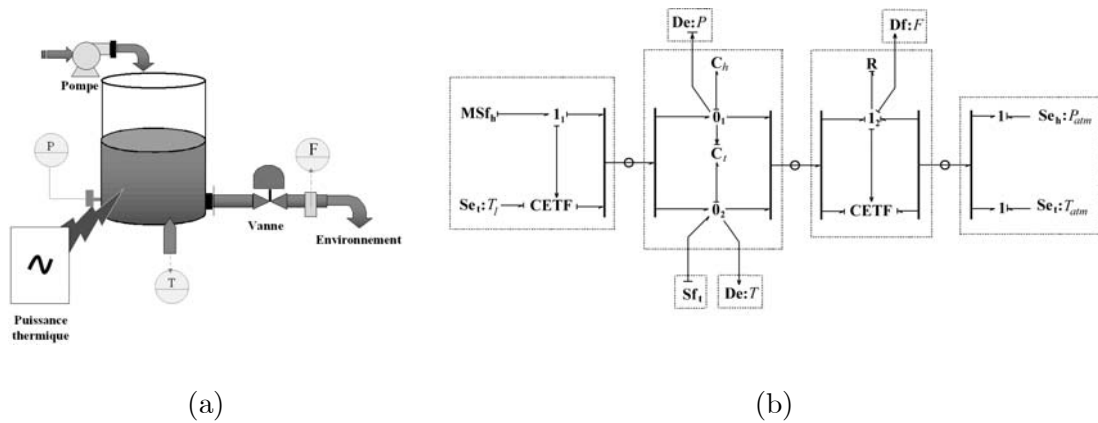


FIG. 3.1 – (a) Un simple système thermofluide et (b) son modèle bond graph

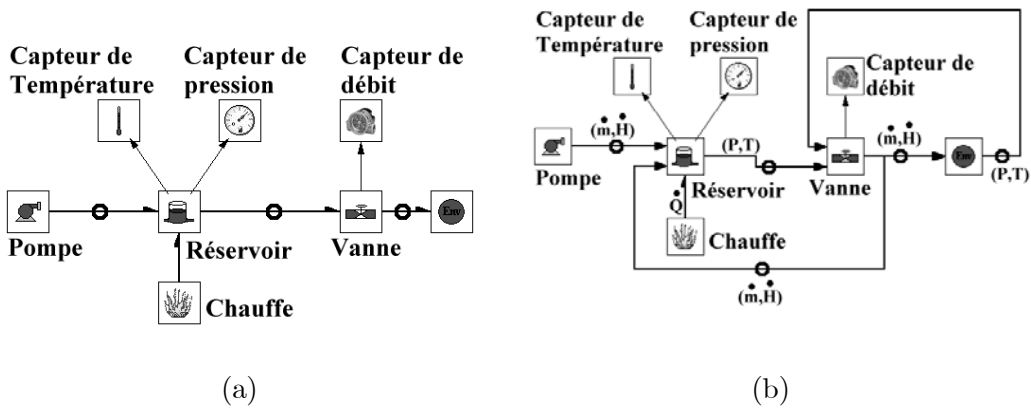


FIG. 3.2 – (a) représentation icônique (b) bloc diagramme

Les connexions passives sont montrées par des liens en demie-flèche et les connexions actives par des flèches complètes. Dans le cas des systèmes mono-énergie, à chaque connexion passive correspondent deux informations (effort, flux). Toutefois, dans les liens vectoriels (figure 3.2(a)) de notre exemple, à chaque point de connexion deux informations sont données et deux autres sont reçues (de dimension 2). Le modèle iconique de la figure 3.2(a) est très proche du P&ID du processus physique et facile à comprendre par l'opérateur.

Le système de la figure 3.1(a) peut aussi être représenté par un bloc diagramme comme le montre la figure 3.2(b). Dans ce cas, chaque connexion passive est dissociée pour être représentée par des actions et des réactions sur le bloc diagramme. Pour les systèmes complexes, cette forme de représentation devient alors moins efficace. Ainsi, la plupart des logiciels en génie des procédés tel que Prosim, gPROMS, etc. utilisent des connexions simples pour interconnecter les composants. Ils utilisent donc une syntaxe de haut niveau (tel que le langage décrit dans le premier chapitre) pour vérifier la validité des interconnexions et du modèle global obtenu, et la causalité n'est pas montrée explicitement à l'opérateur.

Considérons deux composants A et B interconnectés dans un processus donné comme sur la figure 3.3.

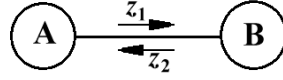


FIG. 3.3 – Deux composants A et B interconnectés

Le composant A fournit l'information z_1 à B qui à son tour lui renvoie l'information z_2 . A noter que si la connexion entre les composants A et B était vectorielle, z_1 et z_2 seraient des vecteurs. La forme générale des équations dynamiques pour l'objet A peut être exprimée par les relations suivantes:

$$\begin{aligned} \dot{x}_A &= \Phi_{x_A}(x_A, u_A, z_2), \\ \begin{pmatrix} y_A \\ \dots \\ z_1 \end{pmatrix} &= \begin{pmatrix} \Phi_{y_A}(x_A, u_A, z_2) \\ \dots \\ \Phi_{z_A}(x_A, u_A, z_2) \end{pmatrix}; \end{aligned} \quad (3.1)$$

où x_A est le vecteur d'état, u_A est le vecteur d'entrée, y_A est le vecteur de sortie, et Φ_{x_A} , Φ_{y_A} et Φ_{z_A} sont des fonctions non linéaires. Pour l'objet A, la variable z_2 est considérée comme une entrée inconnue alors que z_1 est considérée comme une sortie. De même, les équations dynamiques de B peuvent être écrites comme suit:

$$\begin{aligned} \dot{x}_B &= \Phi_{x_B}(x_B, u_B, z_1), \\ \begin{pmatrix} y_B \\ \dots \\ z_2 \end{pmatrix} &= \begin{pmatrix} \Phi_{y_B}(x_B, u_B, z_1) \\ \dots \\ \Phi_{z_B}(x_B, u_B, z_1) \end{pmatrix}; \end{aligned} \quad (3.2)$$

Le logiciel traite les équations précédentes pour identifier les éléments interconnectés et élimine ensuite les variables z_1 et z_2 pour obtenir les équations finales sous la forme suivante:

$$\begin{pmatrix} \dot{x}_A \\ \dots \\ \dot{x}_B \end{pmatrix} = \Phi_x \left(\begin{pmatrix} x_A \\ \dots \\ x_B \end{pmatrix}, \begin{pmatrix} u_A \\ \dots \\ u_B \end{pmatrix} \right), \quad \begin{pmatrix} y_A \\ \dots \\ y_B \end{pmatrix} = \Phi_y \left(\begin{pmatrix} x_A \\ \dots \\ x_B \end{pmatrix}, \begin{pmatrix} u_A \\ \dots \\ u_B \end{pmatrix} \right) \quad (3.3)$$

En l'absence de boucles algébriques impliquant les variables du point d'interconnexion, i.e. z_1 et/ou z_2 ; l'élimination de ces dernières est alors facile à réaliser.

Les équations (3.1) et (3.2) sont issues de modèles à causalité fixe et elles sont facilement interconnectées à travers ce qu'on appelle le "hard-linking" pour générer le modèle dynamique global. Cependant, comme on l'a déjà souligné, on a besoin d'une interconnexion flexible ("soft-linking") des composants en utilisant une interconnexion graphique. L'objectif derrière l'utilisation des modèles bond graphs est de permettre les deux types d'interconnexion: "hard-linking" et "soft-linking" dans un seul outil.

Cependant, le soft-linking ne permet pas une totale flexibilité dans le choix de la causalité. Les règles de la modélisation par bond graphs sont dans ce sens très rigoureuses et strictes et l'affectation de la causalité ne peut être faite de façon aléatoire. Ces contraintes empêchent les logiciels à base des bond graphs d'adopter des causalités leur permettant de faciliter l'élimination des variables inconnues. Prenons l'exemple de la figure 3.4.

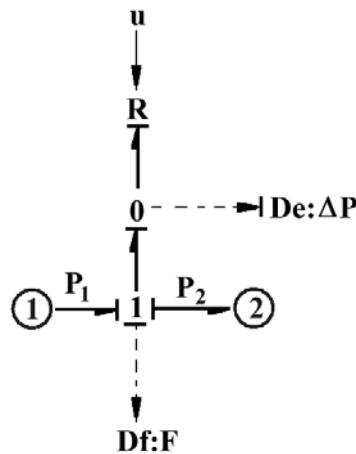
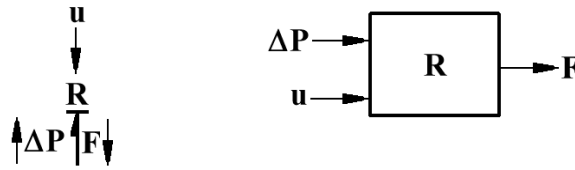


FIG. 3.4 – *Modèle bond graph correspondant à une vanne commandée*

Sur le modèle bond graph, u représente le signal de commande et R la vanne. Les lignes en pointillé représentent l'instrumentation du système physique: $Df : F$ mesure le débit et $De : \Delta P$ la différence de pression au travers de la vanne. En supposons que u et ΔP connus, comme le montre les représentations bond graph et bloc diagramme de la figure 3.5, le débit qui traverse la vanne peut être déterminé par la relation suivante:

$$F = C_d \cdot \sqrt{\Delta P} \cdot \text{sign}(\Delta P) \cdot \Phi(u),$$

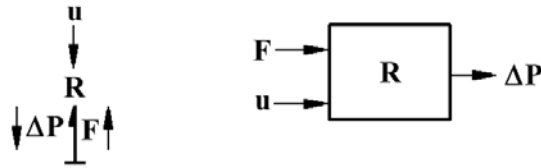
où C_d est le coefficient de décharge de la vanne, Φ est sa fonction caractéristique et sign une fonction pour désigner la direction du flux.

FIG. 3.5 – détermination de F en connaissant u et ΔP

Maintenant, supposons que le débit qui traverse la vanne et la commande u soient connus comme le montrent le bond graph et le bloc diagramme de la figure 3.6. Dans ce cas la différence de pression aux bornes de la vanne peut être calculée par la formule suivante:

$$\Delta P = \left(\frac{F}{C_d \cdot \Phi(u)} \right)^2 \cdot \text{sign.}(F),$$

mais à condition que la valeur de $\Phi(u)$ soit différente de 0. Dans ce cas, on préfère alors la forme causale de l'élément R de la figure 3.5.

FIG. 3.6 – détermination de ΔP en connaissant u et F

Cependant, considérons que les valeurs du débit et de la différence de pression sont connues. Si le débit est non nul, alors on peut toujours déterminer la commande u en connaissant la valeur de retour de la fonction Φ :

$$u = \Phi \left(\frac{C_d \cdot \sqrt{\Delta P} \cdot \text{sign.}(\Delta P)}{F} \right).$$

Cette dernière équation est représentée sous forme de bloc diagramme pour lequel il n'existe pas de représentation bond graph correspondante.

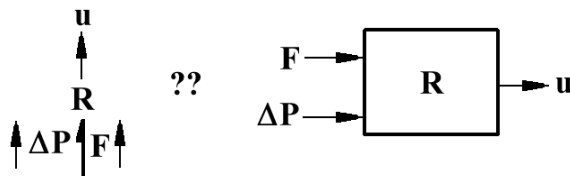


FIG. 3.7 – Cas de causalité non déterminée

En effet, dans ce dernier cas on ne peut pas affecter une causalité à l'élément R pour calculer simultanément les deux informations: l'effort ΔP et le flux F . Les bond graphs ne peuvent représenter que l'ensemble des couplages possibles, en fonction de la causalité, pour éliminer les variables inconnues. Ainsi, la notion de flexibilité de la causalité à laquelle on se réfère est restreinte dans ce sens.

Dans les applications présentées dans ce mémoire, les sorties des régulateurs ont été traitées comme des variables connues pour éviter les problèmes qu'on vient d'évoquer. Ceci étant justifié puisque les sorties des régulateurs sont soit mesurées soit issues des lois de commandes programmées.

3.4 Choix du logiciel pour l'implémentation

Comme on l'a déjà souligné, il existe plusieurs logiciels de modélisation et de simulation par bond graph. Symbols 2000, 20-Sim et Camp-G sont les plus connus et les plus utilisés. Parmi eux, et après une documentation approfondie, nous avons opté pour Symbols 2000 (pour SYstem Modelling through BOnd graph Language and Simulation) afin d'implémenter notre boîte à outils. La dernière version de ce logiciel s'appelle SYMBOLS Shakti, dont le mot Shakti signifie puissance.

Les raisons qui nous ont poussé à choisir Symbols 2000 sont les suivantes:

- Les deux niveaux d'interconnexion: au niveau des équations et au niveau graphique et structurel, sont pris en compte.
- Adaptation de la causalité (en ajoutant ou supprimant des capteurs, composants, etc.).
- Traite des systèmes multi-énergies et permet de créer des modèles dynamiques à partir du P&ID.
- Intégration d'outils d'analyse structurelle.
- Introduction des défaillances dans le modèle.
- Facilité de regrouper des sous-modèles pour construire de nouvelles classes de composants.
- Un bon outil de simulation avec des capacités de traitement de signal.
- Existence d'une collaboration entre notre équipe de recherche et l'Institut Indien de Technologie (Kharagpur) où Symbols 2000 a été développé.

3.5 Implémentation de la boîte à outils

Nous avons utilisé la technologie des capsules [39] pour créer les sous-modèles (objets) des différents composants des systèmes en génie des procédés. Dans Symbols 2000, il existe

déjà une bibliothèque de sous-modèles pour plusieurs composants physiques (mécaniques, électriques, thermiques, hydrauliques), les représentations sous forme de blocs diagrammes, les éléments logiques, les capteurs, etc.. Dans notre boîte à outils, nous avons ajouté les principaux composants des systèmes en génie des procédés (procédés de stockage, de transport, de transformation, les régulateurs, les actionneurs et les capteurs) et nous les avons regroupés avec les composants déjà existants.

Les capsules dans Symbols 2000 sont des objets avec des ports ouverts. Certains ports des capsules peuvent être optionnels, i.e. peuvent ne pas être connectés avec d'autres objets (e.g. un port pour connecter un détecteur). En outre, il existe une possibilité pour créer une capsule générique qui est en fait un template (modèle) contenant plusieurs capsules pour différentes formes causales. Lorsque cette capsule est connectée à d'autres objets externes, la forme causale appropriée est sélectionnée à partir du template. Dans ce sens, Symbols permet une modélisation acausale des composants.

Lorsqu'on crée des capsules, les signaux de différents domaines d'énergie sont regroupés dans des liens simples de telle sorte que, lorsque l'utilisateur utilise ces capsules pour créer un modèle global, le résultat obtenu soit très proche du P&ID. La modélisation bond graph permet de regrouper les liens ayant la même causalité et la même direction dans un lien vectoriel. Cependant, lorsque plusieurs types d'information doivent être transmis en utilisant un seul lien, des transformations sont nécessaires comme on le verra ci-dessous.

Considérons le schéma de la figure 3.8(a) où deux liens scalaires sont utilisés pour connecter les composants A et B. Puisque les causalités des deux liens sont différentes, ils ne peuvent pas être représentés par un seul lien. Toutefois, un lien peut être dissocié en utilisant deux éléments gyrateurs (GY) à module unitaire comme le montre la figure 3.8(b). Ainsi, lorsque c'est nécessaire, on peut mettre un gyrateur dans chaque sous-modèle et créer des causalités similaires aux points d'interconnexion (interface).

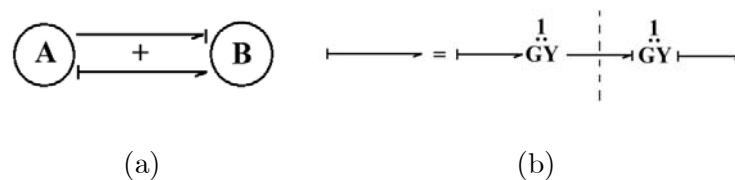


FIG. 3.8 – Transformation d'un lien

Cette façon de modifier le point d'interface (ou le point d'interconnexion) est montrée à la figure 3.9. Les modèles A^* et B^* peuvent alors être interconnectés facilement et sans difficulté pour l'opérateur.

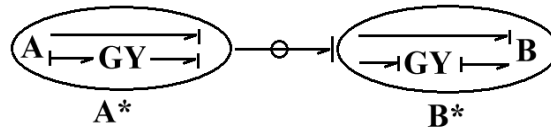


FIG. 3.9 – Interconnexion de deux composants

La procédure de création des modèles dans ModelBuilder est bien détaillée dans le manuel d'utilisation de la boîte à outils.

3.6 Validation du modèle

La validité du modèle architectural construit en interconnectant les différentes capsules prises de la bibliothèque des composants est vérifiée avant tout autre traitement par ModelBuilder. Symbols 2000 possède un ensemble de règles permettant de valider le modèle global obtenu (connexions oubliées ou ambiguës, des variables ou paramètres non déclarés, etc.).

En plus des vérifications existantes, nous avons implémenté les règles supplémentaires suivantes qui sont nécessaires aux processus en génie des procédés.

1. Les dimensions des vecteurs des deux côtés de chaque connexion entre deux capsules doivent être les mêmes.
2. La direction de la puissance échangée entre deux capsules doit être respectée: la sortie de la première capsule devient l'entrée de la deuxième.
3. La causalité lors de l'interconnexion de deux capsules doit être validée: l'information envoyée par une capsule doit être reçue par l'autre capsule. La validité de la causalité assure la compatibilité du modèle obtenu.

En outre, le logiciel vérifie la présence ou non de boucles causales et algébriques sur le modèle. Dans le cas de présence de boucles causales, le système d'équations ne peut pas être généré. Dans le cas de boucles algébriques impliquant les relations constitutives linéaires des éléments, le logiciel génère juste un avertissement puisqu'il est capable de les traiter. Cependant, dans le cas de boucles algébriques impliquant des éléments non linéaires, le logiciel suggère de recourir à des solutions alternatives (e.g. capacités de couplages). Les capacités de couplage permettent de générer des modèles dynamiques approximatifs. Pour des modèles de diagnostic (surveillance) avec présence de boucles algébriques, le logiciel suggère d'ajouter des capteurs pour casser ces boucles. Si les capteurs suggérés ne sont pas ajoutés, alors certaines des variables inconnues ne seront pas éliminées et les résidus correspondant à la partie présentant des boucles ne seront pas générés.

L'interface graphique de ModelBuilder avec le P&ID d'un exemple de processus hydraulique est montrée à la figure 3.10.

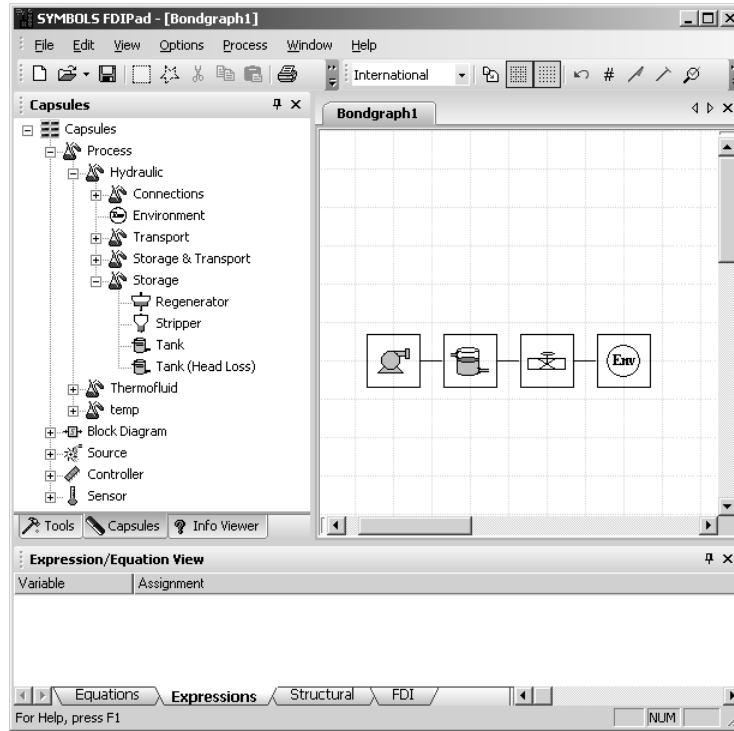


FIG. 3.10 – Interface graphique de ModelBuilder

Sur cette interface, les objets ou composants utilisés en génie des procédés sont classés et affichés sous forme hiérarchique sur la partie gauche de l'interface. La partie en bas de l'interface est utilisée pour afficher le modèle dynamique du processus, les RRAs, la matrice de signature des défaillances et les résultats de l'analyse structurelle. En outre, cette partie est utilisée parfois pour définir les relations constitutives des éléments bond graphs. A noter que la même interface graphique est utilisée, à la fois, pour construire les modèles des processus et pour créer les sous-modèles des objets appelés capsules. Ces dernières sont créées en utilisant les modèles bond graphs des composants déjà existants dans la base de données du logiciel.

Les résultats de validité et de vérification de la syntaxe d'interconnexion des sous-modèles sont affichés dans une fenêtre séparée. Lorsque des erreurs sont détectées, elles sont signalées à l'utilisateur. Ce dernier peut alors accéder à l'erreur par un simple 'click' sur le message d'erreur affiché. La figure 3.11 montre un exemple d'interconnexion non valide entre deux procédés de stockage.

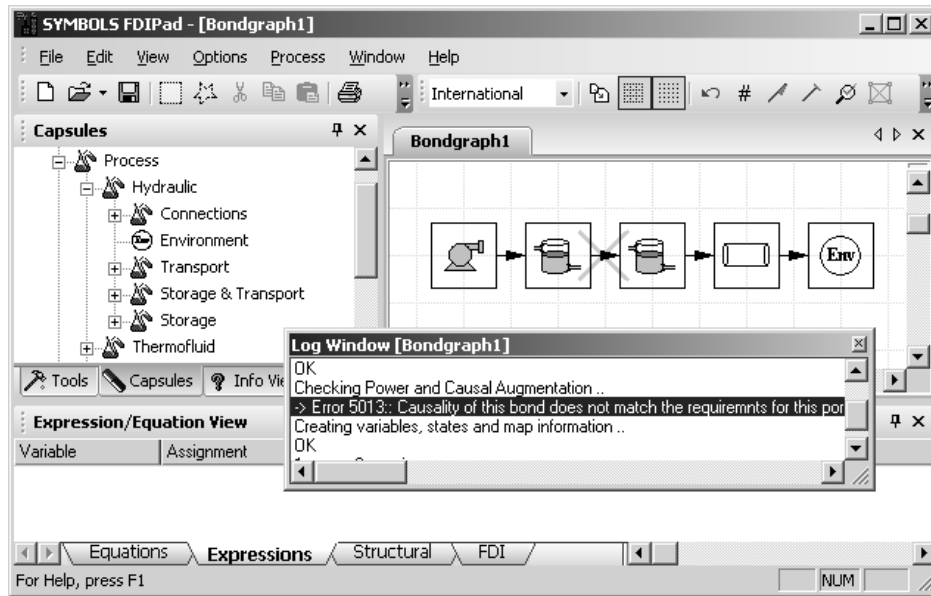


FIG. 3.11 – Exemple d'interconnexion non valide

Pour un modèle valide, ModelBuilder génère le modèle dynamique correspondant sous forme d'équations d'état. Pour générer les RRAs, le modèle global du processus est construit selon les procédures données au premier et au deuxième chapitre. Dans notre implémentation, les étapes suivantes ont été exécutées pour générer les RRAs et la matrice de signature des défaillances.

1. Lors de la création du modèle global, par interconnexion structurelle des capsules, chaque composant possède un nom d'instance (identifiant). Dans le modèle global, les paramètres correspondant à chaque composant sont préfixés (par rapport au nom d'instance).
2. Tous les détecteurs sont convertis en éléments R et une table des conversions est générée. Cela permet d'affecter une quelconque causalité aux détecteurs (directe ou inverse).
3. Affecter la causalité dérivée préférentielle au modèle global.
4. Remplacer tous les éléments R représentant les détecteurs par les détecteurs respectifs (selon la table des conversions).
5. Déterminer les redondances matérielles en analysant les causalités des détecteurs.
6. Apporter les modifications nécessaires sur les détecteurs pour générer le modèle bond graph de diagnostic.
7. Appeler les routines de Symbols pour générer les équations du modèle. Les équations de sortie (correspondant aux détecteurs de résidus) ne comportant pas de variables inconnues sont alors des RRAs.

8. Dans le cas où le nombre de RRAs générées est inférieur au nombre de détecteur de résidus, alors certaines des variables inconnues n'ont pas pu être éliminées. Il faut donc vérifier les équations pour voir s'il reste des variables inconnues. Cela nécessite un parseur. Les routines de Symbols sont utilisées pour déterminer les différents couplages entre variables. Ensuite, procéder à l'élimination des variables inconnues et vérifier que la nouvelle RRA obtenue est structurellement indépendante des autres déjà générées. Cela nécessite aussi un parseur.
9. Trier les RRAs, éliminer les singularités (en multipliant par des variables, paramètres, etc. appropriés) et afficher les résultats à l'opérateur.
10. Sur les RRAs, les variables apparaissent avec des préfixes correspondant aux noms d'instance. Par exemple, Tank1_g et Boiler_g, dont 'g' représente la constante de gravité. Ces variables ne sont pas automatiquement considérées comme égales par le logiciel. S'il le veut, l'utilisateur peut les considérer comme égales. Pour cela, il faut supprimer tous les noms d'instance de tous ces paramètres (e.g. gravité, densité dans un modèle mono-fluide, pression atmosphérique, température ambiante, etc.). Afficher les résultats modifiés à l'opérateur.
11. Parser (analyser) les RRAs et mettre en évidence toutes les variables utilisées dans chaque RRA. Dans chaque terme, trouver les noms d'instance et les utiliser pour construire la matrice de signatures des défaillances.
12. La matrice de signature des défaillances initiale générée par le logiciel suppose que tous les composants peuvent être défaillants. Dans ce cas, le cahier des charges relatif à la surveillance des composants doit être donné par l'utilisateur. Par exemple, dans le modèle de la figure 3.1(a), le composant 'Environnement' peut être déclaré infaillible. Après la spécification des composants infaillibles, certaines lignes correspondantes seront supprimées de la matrice des signatures. La surveillabilité de chaque composant est alors déterminée à partir de l'analyse de la matrice de signature des défaillances.

Les résultats obtenus par ModelBuilder peuvent ensuite être exportés sous format XML (eXtended Mark-up Language) vers d'autres boîtes à outils.

3.6.1 Simulation de défaillances et réponses des résidus

Les RRAs générées par ModelBuilder sont destinées à être utilisées dans des applications de surveillance en temps réel. Cependant, pour tester les résidus correspondants et afin de déterminer leur sensibilité par rapport aux différentes défaillances (certaines défaillances ne peuvent pas être introduites sur le processus réel), un modèle de simulation s'avère nécessaire. Ainsi, chaque modèle de composant, sous forme de capsule, possède deux versions: mode normal et mode défaillant. Sur la capsule en mode défaillant, on peut changer les paramètres

du composants, introduire des sources supplémentaires, etc. afin de créer des défaillances. La capsule correspondant au mode de fonctionnement normal est utilisée pour générer les RRAs et la capsule correspondant au mode défaillant est utilisée pour générer le modèle dynamique (en causalité intégrale).

Dans ModelBuilder, le modèle continu (en causalité intégrale) et le modèle d'évaluation (en temps discret) des RRAs sont tous les deux couplés durant les simulations en utilisant les outils de gestion d'événements offerts par Symbols 2000.

Le simulateur de Symbols 2000 utilise trois paramètres pour gérer les propriétés du pas de simulation: (a) la durée totale de simulation T , (b) le nombre de sorties enregistrées n , et (c) les pas supplémentaires propres à chaque sortie enregistrée. La sortie de simulation est disponible toutes les $\frac{T}{n}$ secondes. La période d'échantillonnage pour introduire les événements discrets est un multiple de $\frac{T}{n}$. Le simulateur possède deux moyens pour traiter les événements:

1. Les événements utilisateur: ils permettent de modifier les valeurs des paramètres durant le déroulement de la simulation. Il existe des curseurs (sliders) sur lesquels les valeurs minimum et maximum des paramètres peuvent être spécifiées. L'utilisateur peut alors positionner le curseur sur n'importe quelle valeur du paramètre durant la simulation. Ceci permet d'introduire des défaillances sur les modèles dynamiques.
2. Les événements temporels: il existe trois routines pour introduire des événements temporels.
 - a. OnSimulationStart: permet l'initialisation des événements, l'ouverture de fichiers pour les entrées/sortie, l'allocation de la mémoire, etc.
 - b. OnSimulationSettle: c'est à travers cette routine que les événements temporels sont introduits (e.g. défaillance à un instant donné). Les RRAs sont aussi calculées pendant ce temps. Cette tâche est exécutée une fois durant chaque pas de simulation. Pour chaque terme, en forme de dérivée dans l'expression d'une RRA, est associée une variable. En outre, d'autres variables globales pour stocker les valeurs précédentes des variables dérivées sont introduites en fonction de l'ordre de la dérivée. Les variables stockées sont ensuite utilisées pour calculer les dérivées totales. Les RRAs sont ainsi calculées et les résultats sont obtenus comme des variables de sortie. Les données fournies par les capteurs d'un processus réel sont en général bruitées. Pour tenir compte du bruit, des nombres pseudo-aléatoires (à un pourcentage fixe de la valeur de sortie des détecteurs) sont ajoutés au modèle.
 - c. OnSimulationStop: est utilisé pour fermer les fichiers ouverts pour les entrées-sorties, libérer l'espace mémoire alloué, etc.

De plus, Symbols 2000 offre plusieurs modes de simulation possibles: simulation à durée fixe, simulation continue et simulation répétitive utilisée dans l'optimisation. Dans notre cas,

la simulation continue produisant des résultats comme sur un oscilloscope a été préférée. Il existe aussi dans le simulateur un ensemble d'outils utilisés dans le post-traitement (e.g. moyenne glissante sur les résidus).

3.7 Exemple d'illustration: banc d'essai du projet CHEM

L'exemple du système des deux réservoirs montré à la figure 3.12 a été utilisé comme banc d'essai dans le projet CHEM. Dans ce qui suit, l'exemple est utilisé pour montrer les fonctionnalités de la boîte à outils ModelBuilder. La fonction principale de ce processus est de fournir un débit d'eau Q_0 continu à un consommateur. Le système consiste en deux réservoirs T_1 et T_2 reliés par une conduite et une vanne. Le réservoir T_1 est rempli en eau jusqu'à un niveau $h_1 = 0.5 m$ au moyen d'une pompe P_1 . Ce niveau est régulé par un PI en agissant sur le débit d'entrée Q_p fourni par la pompe. Le débit d'eau qui traverse la vanne V_b est régulé par un régulateur "On - Off". En agissant sur la vanne V_b , on maintient le niveau d'eau h_2 dans le réservoir T_2 autour d'une valeur donnée ($0.09 m \leq h_2 \leq 0.11 m$). La quantité d'eau Q_0 fournie à un consommateur dépend du degré d'ouverture ou de fermeture de la vanne V_0 .

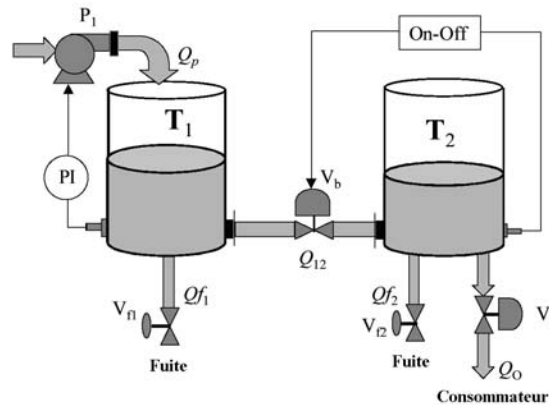


FIG. 3.12 – Le banc d'essai utilisé dans le projet CHEM

Les vannes V_{f1} et V_{f2} sont utilisées pour simuler des fuites dans les réservoirs T_1 et T_2 .

En utilisant ModelBuilder, le modèle architectural du processus est construit par une simple interconnexion des composants qui le composent. Ensuite, les liens du modèle sont numérotés comme le montre la figure 3.13. Ces numéros sont utilisés pour définir les connexions entre les ports des composants.

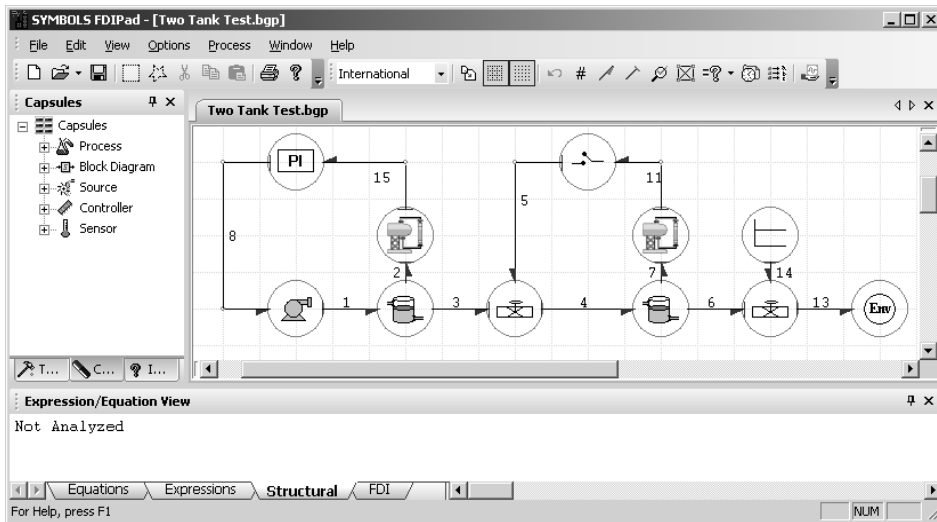


FIG. 3.13 – Modèle architectural du banc d'essai

Chaque composant du modèle architectural peut être identifié et sa connectivité peut être spécifiée à travers une boîte de dialogue (figure 3.14).

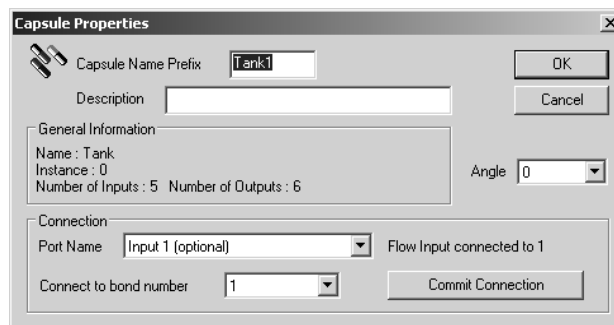


FIG. 3.14 – Boîte de dialogue pour l'identification d'un composant

Sur cette boîte de dialogue l'identification et les connexions du réservoir "Tank1" ont été spécifiées. La même chose est faite pour le réservoir "Tank2". La connexion entre deux composants est prise en compte après avoir indiqué le nom du port et le numéro du lien à connecter et une fois validé par le bouton "Commit Connection". La même procédure est appliquée sur tous les ports et les liens numérotés pour établir la connexion du modèle global. Pour le cas du composant "Tank1" les connexions suivantes ont été établies:

1. L'entrée 1 est connectée au lien numéro 1
2. La sortie 1 est connectée au lien numéro 3
3. Le capteur de niveau est connecté au lien numéro 2

Concernant les autres composants du modèle de la figure 3.13, les identifiants suivants ont été donnés:

1. Tank1 et Tank2 pour les réservoirs.
2. Valve1 et Valve2 pour les vannes V_b et V_0 .
3. Pump pour la pompe.
4. PI et OnOff pour les deux régulateurs du processus.
5. L1 et L2 pour les deux capteurs de niveau d'eau.
6. Env pour le composant Environnement.

Une fois les connexions réalisées, les commandes du menu “Options”: “Power” et “Causal” sont utilisées. Elles sont relatives à la modélisation bond graph et sont nécessaires pour définir la direction de la puissance échangée entre les composants dans le modèle bond graph global. De plus, il est important de vérifier la validité du modèle. Pour cela, la commande “Check Integrity” du menu “Process” est appelée. En cas de détection d’erreur, elles sont signalées dans une boîte de dialogue à l’opérateur. Ce dernier peut visualiser les éléments responsables des erreurs et apporter les modifications nécessaires pour les supprimer.

Pour l’utilisateur de ModelBuilder, il est préférable de faire abstraction sur tous les liens et les numéros correspondants pour obtenir un modèle architectural qui soit le plus proche du P&ID de son processus. Cette fonction est réalisée en choisissant le mode bloc diagramme au moyen de la commande “Block Diagram Mode” du menu “View”. Le modèle du processus sous forme de bloc diagramme obtenu par ModelBuilder est montré à la figure 3.15.

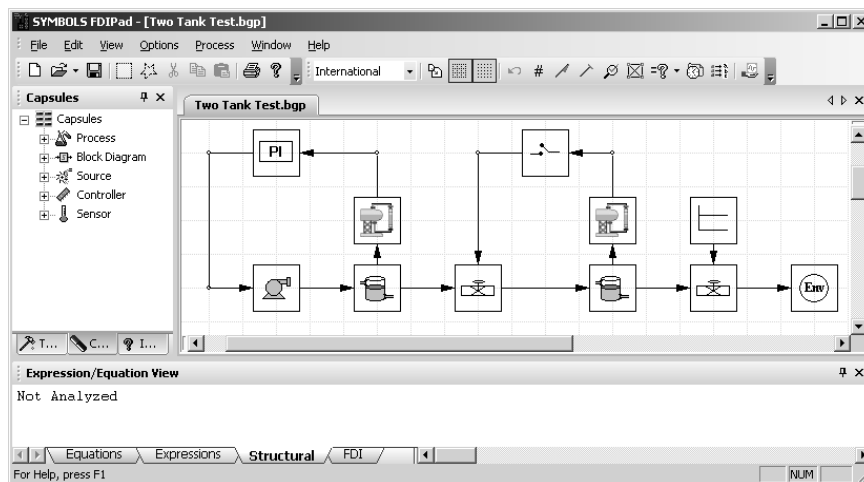


FIG. 3.15 – Bloc diagramme du banc d’essai obtenu avec ModelBuilder

Le modèle global obtenu après validation est utilisé pour la génération des relations symboliques et des outils pour la surveillance. Sur ModelBuilder, toutes les commandes relatives

à la surveillance sont regroupées dans le sous-menu “FDI” du menu “Process” (figure 3.16).

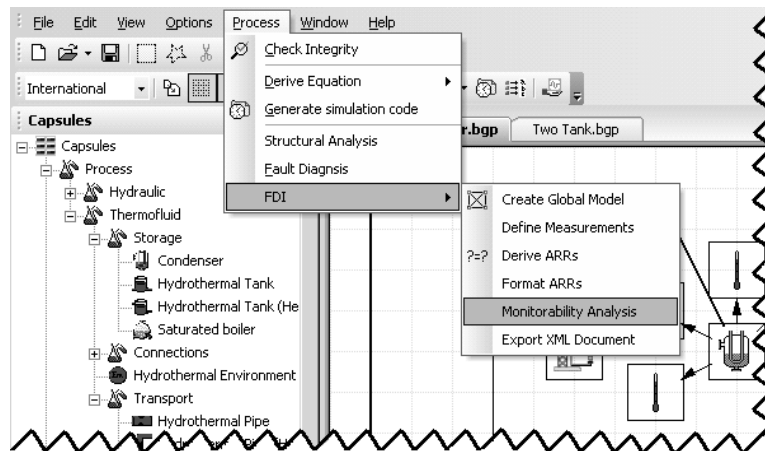


FIG. 3.16 – Le sous-menu “FDI” de ModelBuilder

1. Le modèle global du processus est créé en cliquant sur la commande “Create Global Model”. Cette dernière permet d’interconnecter les différents composants du processus afin de créer en mémoire le modèle bond graph global correspondant. Ce modèle n’est pas affiché à l’utilisateur mais juste utilisé pour effectuer les calculs mathématiques nécessaires.
2. ModelBuilder génère automatiquement les noms des variables connues (sources et mesures). L’utilisateur peut ensuite par la commande “Define Measurements” attribuer à ces variables des noms qui seront ensuite utilisés dans les équations symboliques des RRAs (figure 3.17).

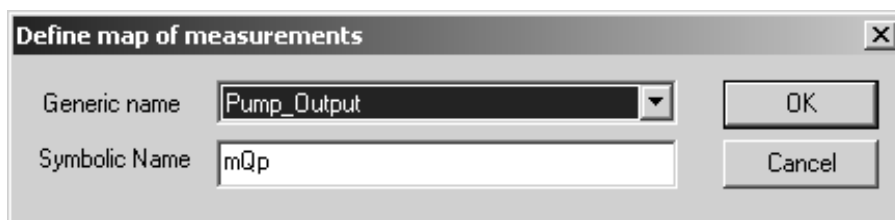


FIG. 3.17 – Définition des variables des mesures et des sources

Concernant l’exemple du banc d’essai, la nomenclature utilisée pour les sources et les mesures est donnée au tableau 3.18.

3. Les RRAs sous forme symbolique du processus sont automatiquement générées en cliquant sur la commande “Derive ARR’s” du sous-menu “FDI”. Elles sont alors affichées

Nom automatique	Variable utilisée
Pump_Output	mQp
OnOff_Output	mUb
User_Output	mUo
Env_Pressure	Po
L1_Measurement	L1
L2_Measurement	L2
PI_Output	mUp

FIG. 3.18 – La nomenclature des variables connues utilisées dans le banc d’essai

dans la partie en bas de l’interface de ModelBuilder. Les RRA obtenues contiennent des nomenclatures différentes de certaines variables ou paramètres. Toutefois, dans certaines applications ces nomenclatures peuvent être identiques (la densité dans un écoulement mono-fluide par exemple). En outre, les RRA peuvent parfois contenir des dérivées de constantes ou des variables singulières dans les dénominateurs. La commande “Format ARRs” est alors utilisée pour simplifier les RRA et les présenter sous une forme acceptable par l’utilisateur, comme le montre la figure 3.19.

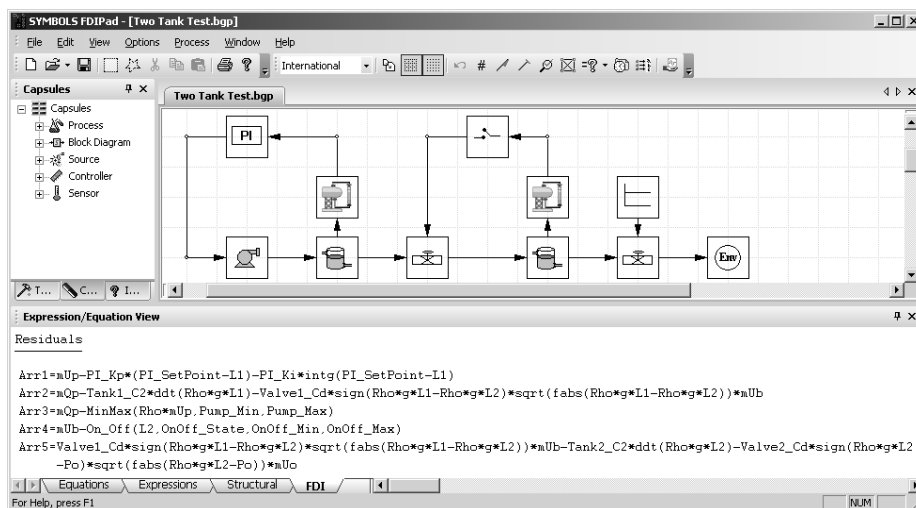


FIG. 3.19 – Les RRA formatées du banc d’essai

4. La surveillabilité (déteçtabilité et isolabilité) des composants du processus peut être

testée en utilisant les RRAs générées par ModelBuilder. Pour cela, il faut d'abord ex-
 clure sur la boîte de dialogue (figure 3.20 obtenue par la commande "Monitoring-ability
 Analysis") les composants qui sont, selon le cahier des charges fixé, infaillibles.

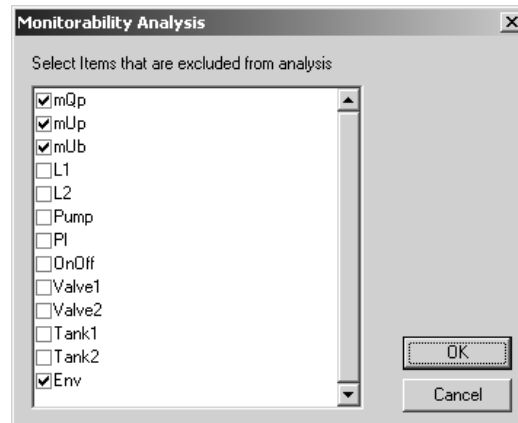


FIG. 3.20 – Spécification des composants infaillibles

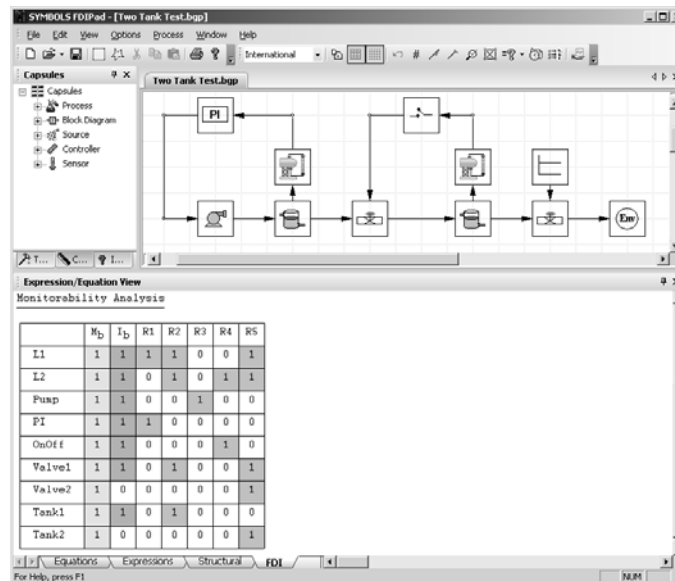


FIG. 3.21 – Surveillabilité du banc d'essai

Sur la boîte de dialogue de la figure 3.20 sont affichées les variables des mesures, des sources et des composants du processus. Sur cette fenêtre, on peut spécifier les composants qu'on a jugé infaillibles dans le domaine d'application de processus. Par exemple, le composant 'Environment' est dans ce cas jamais défaillant. De même, on suppose que l'intervention de

l'utilisateur (mUp) sur la vanne V_0 ne peut pas être défaillante. Ainsi, la matrice de signature des défaillances du banc d'essai est générée par ModelBuilder (figure 3.21).

Sur la matrice des signatures de la figure 3.21, R1, R2,..., R5 représentent les résidus correspondant aux RRAs et M_b et I_b sont respectivement la détectabilité et l'isolabilité des défaillances. Les lignes de la matrice sont les signatures des composants (i.e. dépendance des résidus par rapport aux défaillances des composants). La valeur 1 signifie que la défaillance du composant influe théoriquement sur la réponse du (ou des) résidu(s) (0 sinon). Lorsque la variable associée à un composant apparaît dans au moins un résidu, alors sa défaillance est détectable ($M_b = 1$). Si la signature d'un composant est unique (strictement différente des autres signatures) sa défaillance est isolable ($I_b = 1$).

5. Les résultats obtenus par ModelBuilder peuvent être exportés, sous forme d'un fichier XML, vers d'autres utilisateurs en utilisant la commande "Export XML document". Le document XML créé est alors validé par rapport au DTD (Document Type Definition) fourni par ModelBuilder.

En plus des équations des RRAs, ModelBuilder génère les équations du modèle dynamique en causalité intégrale. Les sorties de ce dernier sont alors utilisées pour l'évaluation numérique des RRAs et pour la simulation de défaillances du processus. Des résultats de simulation correspondant aux niveaux d'eau dans les deux réservoirs sont donnés à la figure 3.22.

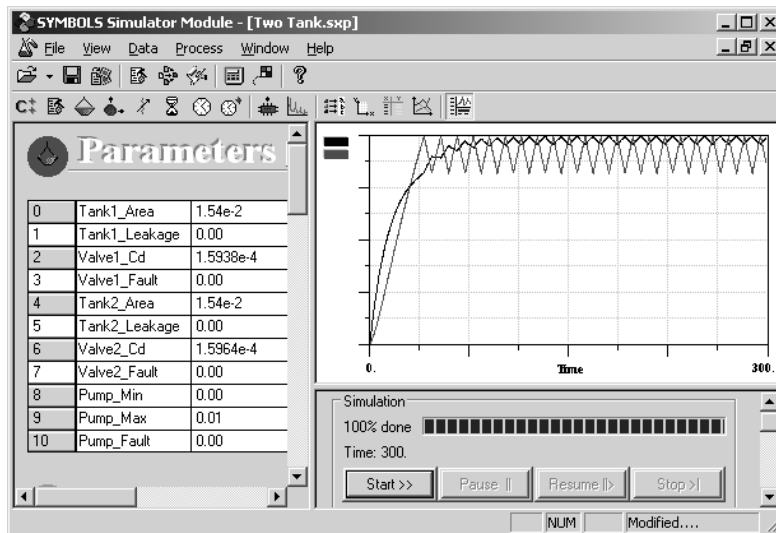


FIG. 3.22 – Le niveau d'eau dans les réservoirs du banc d'essai

Des curseurs sont utilisés pour introduire des défaillances dans le processus (figure 3.23). Une défaillance (blocage de la vanne "Valve 1") a été introduite pendant une durée de 100 s: de 140 s à 240 s, en ramenant le curseur $n^{\circ} 1$ à sa position maximum. Les réponses des

résidus R2 et R5 à cette défaillance sont montrées par la figure 3.23.

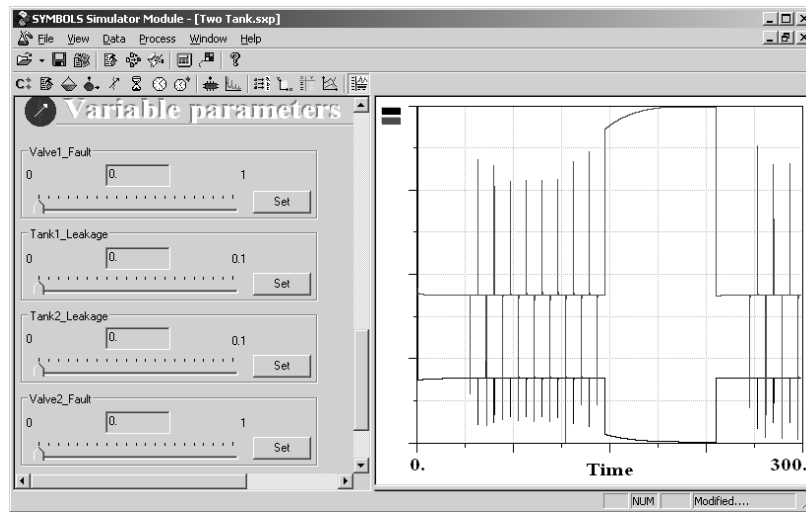


FIG. 3.23 – Introduction d'une défaillance et résultats obtenus

Afin d'atténuer l'effet des bruits sur les réponses des résidus, des traitements supplémentaires peuvent être réalisés. La figure 3.24 montre les résultats obtenus suite à une moyenne glissante sur 5 s appliquée sur les résidus R2 et R5.

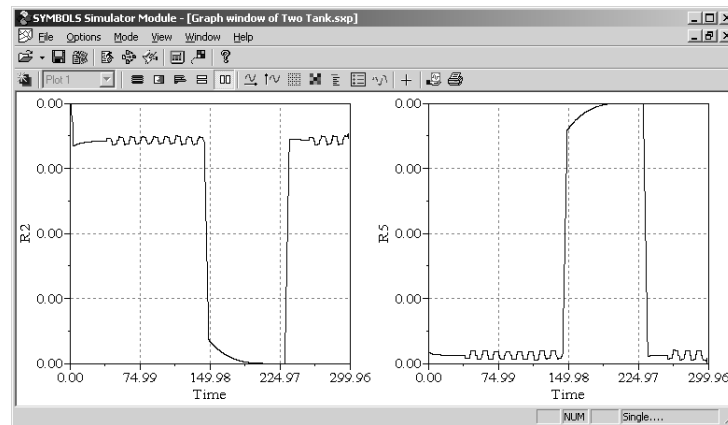


FIG. 3.24 – Moyenne glissante appliquée sur les résidus R2 et R5

3.8 Conclusion

L'implémentation informatique, sous forme d'une boîte à outils, des procédures développées dans le premier et le second chapitre a été présentée dans ce chapitre. Les tests réalisés avec ModelBuilder sur le banc d'essai du projet CHEM ont montré de bons résultats. Dans le

prochain chapitre cette boîte à outils est utilisée pour traiter de la modélisation et de l'analyse de la surveillabilité de deux processus industriels complexes. Les résultats expérimentaux obtenus sont utilisés pour valider la boîte à outils.

Chapitre 4

Application

4.1 Introduction

L'objectif de ce chapitre est de valider la boîte à outils ModelBuilder à partir des tests effectués sur un processus pilote: le générateur de vapeur du LAGIS.

Le générateur de vapeur est une installation pilote située au Laboratoire d'Automatique, Génie Informatique et Signal (LAGIS) de l'Université des Sciences et Technologies de Lille. Il représente le fonctionnement à l'échelle réduite d'une centrale thermique. Cette installation est un procédé énergétique thermofluide bien instrumenté et relié à un PC de supervision. Elle a été choisie comme plateforme de test dans le projet CHEM car nous avons implémenté les algorithmes développés dans ce travail de recherche, du P&ID jusqu'à la supervision en temps réel.

Remarque 4.1.1 *La boîte à outils ModelBuilder a été testée aussi sur un autre processus. Il s'agit du pilote de FCC (Fluid Catalytic Cracking) de l'Institut Français du Pétrole (IFP). Le pilote de FCC est un processus physico-chimique d'une nature différente par rapport au générateur de vapeur. Nous avons voulu par cette application montrer la faisabilité des outils développés, en utilisant les bond graphs, sur des phénomènes physiques complexes décrits par des équations empiriques. Cependant, pour des raisons de confidentialité imposées par l'IFP, les résultats obtenus ne sont pas publiés dans ce mémoire.*

Le travail présenté ici a été réalisé dans le cadre du projet CHEM. Ce dernier regroupe plusieurs équipes de recherches, chacune dans son domaine, pour concevoir une plateforme de supervision des systèmes chimiques et pétrochimiques. Pour cela, les résultats de chaque équipe sont implémentés dans une boîte à outils pour être utilisés en cas de besoin par les autres partenaires du projet. L'intégration des différentes boîtes à outils utilisées dans le projet sont présentées par la figure 4.1.

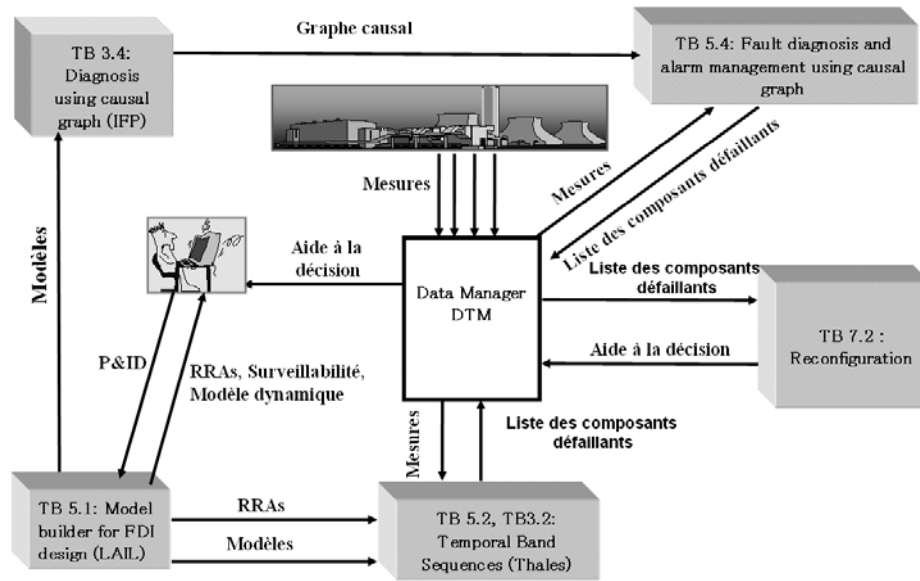


FIG. 4.1 – *Intégration des différentes boîtes à outils dans la plateforme de supervision du projet CHEM.*

La boîte à outils TB 5.1 fournit le modèle et les RRAs aux boîtes à outils TB 3.2 et TB 5.2. Ces dernières transforment les RRAs en formes polynomiales pour générer les séquences de bandes temporelles (SBT) [54] en tenant compte des incertitudes de mesures des capteurs et des paramètres du processus. Les SBTs sont ensuite évaluées pour détecter en temps réel les éventuelles défaillances. Parallèlement, la TB 3.4 reçoit le modèle de la TB 5.1 et génère des indicateurs de pannes en se basant sur le diagnostic à base des graphes causaux. Ces indicateurs sont ensuite transmis à la TB 5.4 qui, en fonction des mesures fournies par le Data Manager, génère la liste des composants défaillants. Cette information est récupérée par la TB 7.2 dans le but de fournir une aide à la prise de décision lors de présence de défaillances. Cette aide est fonction de la disponibilité des composants de l'installation à superviser.

4.2 Application: le générateur de vapeur

4.2.1 Description du processus

Le générateur de vapeur est une installation pilote qui reproduit le fonctionnement, à l'échelle réduite, d'une centrale thermique. L'installation est constituée: d'une chaudière d'une capacité totale de 170 l et qui produit la vaporisation d'eau au moyen d'une résistance thermique d'une puissance de 55kW, d'un système de détente de la vapeur, d'un condenseur couplé à un échangeur de chaleur, d'un réservoir de stockage et d'un circuit d'alimentation en eau. La charge à la sortie de la chaudière est représentée par le système de détente composé

des deux vannes réglées (V_1 et V_2) reliées au composant condenseur-échangeur de chaleur. Le niveau d'eau dans la chaudière est maintenu à ± 3 l autour d'une valeur de consigne donnée en activant une pompe qui l'alimente à partir du réservoir. Pour réguler la pression de la chaudière, un régulateur ToR (Tout ou Rien) agit sur la résistance thermique pour maintenir la pression à ± 0.2 bar d'une consigne préalablement fixée. Le plan des instruments détaillé (P&ID) du générateur de vapeur est donné par la figure 4.2.

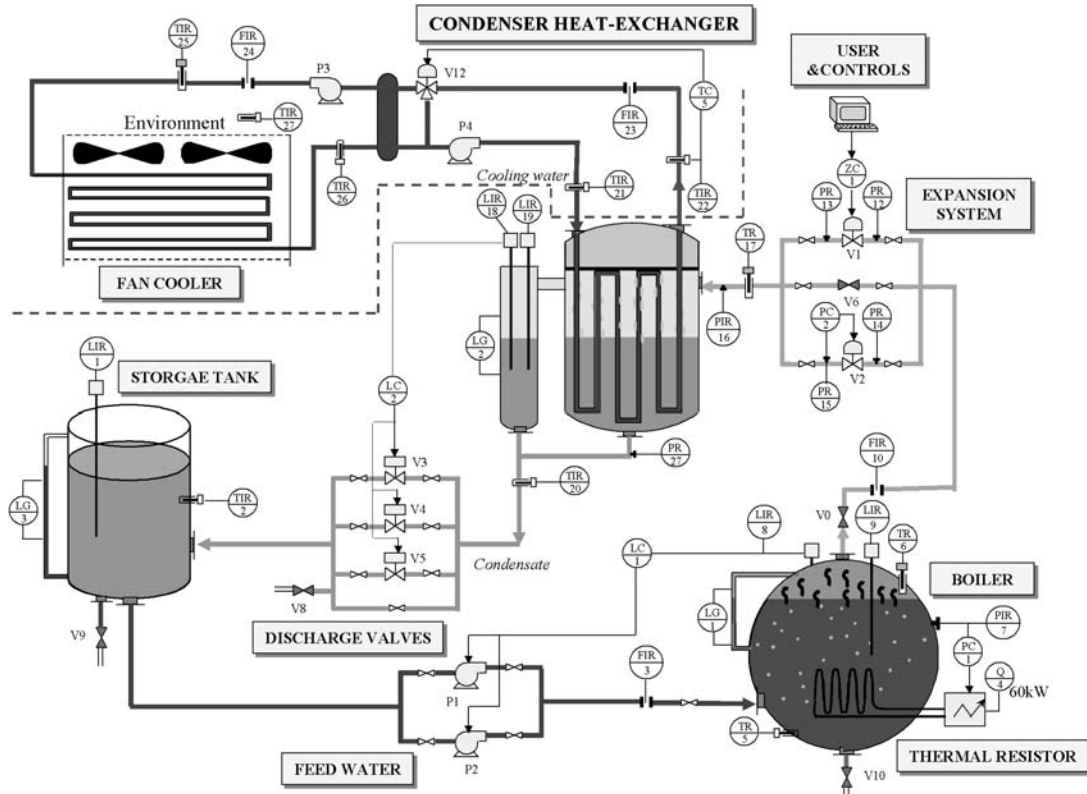


FIG. 4.2 – Plan des instruments détaillé du générateur de vapeur

Dans la modélisation du générateur de vapeur, les hypothèses suivantes ont été considérées:

- la vapeur et l'eau sont saturées,
- les propriétés thermodynamiques sont calculées à partir d'un équilibre thermodynamique établi (on considère que le mélange eau-vapeur est homogène),
- le mélange eau-vapeur est à une pression uniforme et les variables sont à paramètres localisés,
- la phase liquide de l'eau est incompressible.

Le condenseur est un système thermodynamique complexe. La vapeur d'eau qui traverse le système de détente est condensée sur des tubes verticaux dans lesquels circule de l'eau froide.

Le changement de phase, de la vapeur en condensât, est dû à la libération par la vapeur de sa chaleur latente de vaporisation. Le liquide formé est alors refroidi à une température inférieure à la température de saturation. En augmentant en épaisseur, la couche de liquide finit par tomber sous l'effet de son poids au fond du condenseur. Les tubes du serpentin sont divisés en trois sections: (1) à l'entrée du condenseur en contact avec la vapeur, (2) noyés dans le condensât et (3) en contact de la vapeur à la sortie du condenseur. En outre, on suppose que le condenseur est complètement isolé, ses paramètres sont localisés et la vapeur qu'il contient est saturée.

Le niveau du condensât est réglé au moyen de trois vannes tout ou rien (V_3 , V_4 et V_5) placées entre le condenseur et le réservoir. Elles maintiennent le niveau du condensât (L_{18}) à ± 0.5 l d'une valeur de consigne donnée.

L'installation du générateur de vapeur a été spécialement conçue pour tester différentes méthodes de surveillance. Il est possible de simuler, manuellement ou automatiquement, un grand nombre de défaillances. Ces dernières peuvent être:

- physiques: fuite d'eau du condenseur (ouverture de la vanne V_8), du réservoir (ouverture de la vanne V_9) ou de la chaudière (ouverture de la vanne V_{10}); blocage partiel ou total de la sortie de la chaudière au moyen de la vanne V_0 ,
- actionneurs: défaillance de la résistance thermique, des pompes, etc.,
- capteurs: en supprimant par exemple leur alimentation électrique au moyen d'interrupteurs dédiés à cet effet,
- régulateurs: en modifiant les algorithmes de commande en temps réel.

4.2.2 Modèle bond graph des composants du générateur de vapeur

Les composants du générateur de vapeur appartiennent aux classes de composants thermo-fluides définies dans le premier chapitre de ce mémoire. Leurs modèles bond graphs sont créés et sauvegardés dans la bibliothèque de la boîte à outils ModelBuilder présentée au chapitre précédent. L'interconnexion de ces composants permet de construire le modèle dynamique du générateur de vapeur, de générer les RRAs et la matrice de signatures de pannes correspondante, d'analyser la surveillabilité du processus et de réaliser un placement de capteurs.

La section suivante présente le modèle bond graph du condenseur qui n'a pas été donné dans les chapitres précédents.

Modèle bond graph du condenseur

Le condenseur du générateur de vapeur est l'un de ses composants les plus complexes. La vapeur qui arrive de la chaudière est condensée sur des tubes verticaux en forme de U (serpentin) et dans lesquels circule une eau froide. Le condensât produit est stocké au fond

du condenseur et son niveau est régulé par l'ouverture ou la fermeture de trois vannes tout ou rien qui composent le système de décharge. Une partie des tubes est submergée dans le condensât tandis que l'autre partie est en contact permanent de la vapeur. Des transferts de chaleur mutuels entre la vapeur et le liquide ont lieu dans le condenseur et la variation du niveau du condensât a pour effet la variation du volume de la vapeur et donc sa compression ou sa détente.

L'épaisseur du film de condensât formé augmente avec la condensation de la vapeur et engendre davantage d'échange de chaleur du condensât vers les tubes du serpentin. La formule de Nusselt [55] permet de calculer l'épaisseur du film, le coefficient d'échange de chaleur total et la vitesse de condensation.

Le condenseur est aussi considéré comme faisant partie de la charge par rapport à la chaudière. Les valeurs de consigne sur le niveau de condensât correspondent à différentes longueurs des tubes en contact de la vapeur et donc à différentes vitesses de condensation. Le débit de vapeur (et donc la charge) qui sort de la chaudière est alors fonction du niveau du condensât.

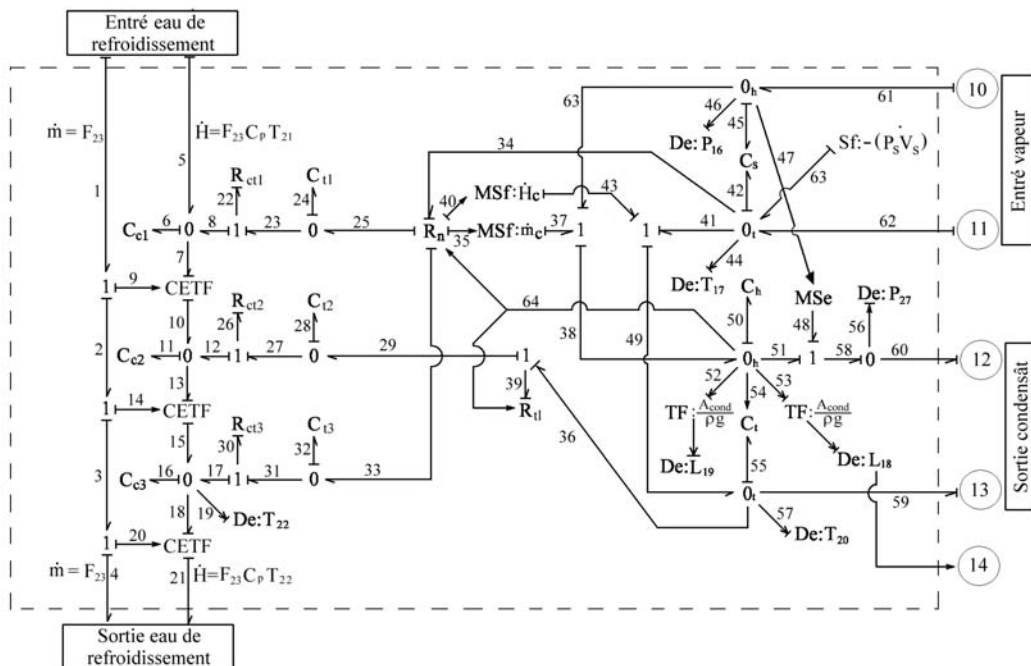


FIG. 4.3 – Modèle pseudo-bond graph du condenseur-échangeur de chaleur.

Pour simplifier la modélisation, on suppose que le condenseur est complètement isolé, que ses paramètres sont localisés et que la vapeur qu'il contient est saturée. Ces hypothèses s'ajoutent à celles formulées pour l'équation de Nusselt [56]:

1. La vapeur est pure et saturée. L'advection est négligée;

2. La température de la surface de condensation est constante;
3. Le condensât tombe, à cause de son poids, sous l'effet de la force de gravité et son écoulement est laminaire;
4. L'épaisseur du film de condensât est faible comparée au diamètre des tubes.

Les hypothèses de vapeur saturée et d'écoulement laminaire sont difficiles à obtenir en pratique en raison des sédiments qui se déposent sur les tubes et des gaz non condensables. Il résulte alors des coefficients d'échange de chaleur différents de ceux de la théorie. Des corrections à la formule de Nusselt [57] sont alors nécessaires pour obtenir un modèle le plus proche possible représentant la vitesse de condensation et les coefficients d'échange de chaleur.

Le modèle pseudo-bond graph du condenseur est donné à la figure 4.3. L'élément C_s , sur le modèle, représente le stockage de la masse et de l'énergie de la vapeur. Sa relation constitutive est identique à celle de l'élément de stockage du mélange dans la chaudière. Contrairement au mélange eau-vapeur, le condensât stocké au fond du condenseur est modélisé par deux éléments C séparés: C_h et C_t . Le premier (C_h) représente la partie hydraulique et le second (C_t) représente la partie thermique. Cette partie a exactement le même modèle que celui du réservoir de stockage. La pression au fond du condenseur est égale à la pression due au condensât plus celle créée par la vapeur (représentée par le lien n° 48 de la jonction 1).

Sur le modèle bond graph du condenseur, ce qui est plus difficile à représenter est la formule de Nusselt. Le phénomène de condensation de la vapeur est représenté en utilisant un élément multiport $R : R_n$. En plus des paramètres géométriques et physiques du condenseur, la vitesse de condensation dépend de la température de la vapeur et de la surface de contact (nombre de tubes). Le débit massique de condensât, représenté par la source de flux modulée $MSf : \dot{m}_c$, peut être alors calculé à partir de l'élément R_n . Physiquement, la condensation est la conséquence de la perte par la vapeur de sa chaleur latente de vaporisation et du film de condensât formé. Cette chaleur est transmise de la vapeur aux tubes par conduction et représentée par les liens 34, 25, et 33 du multiport R_n .

La vitesse de condensation dépend aussi de la longueur des tubes en contact de la vapeur. Cette longueur est fonction du niveau de condensât (produit et stocké) qui est signalé sur le modèle bond graph comme une modulation par le lien 64 de l'élément R_n . En outre, on suppose que l'énergie thermique du condensât formé est instantanément transmise au liquide contenu au fond du condenseur. Ce transfert de chaleur par convection est modélisé par la source de flux thermique modulée $MSf : \dot{H}_c$.

En utilisant la formule de Nusselt, la vitesse de condensation \dot{m}_c et l'échange de chaleur résultant \dot{Q} , de la vapeur vers les tubes, sont donnés par les équations suivantes:

$$\dot{m}_{ci} = \frac{\pi D n_t g \rho_l (\rho_l - \rho_v)}{3\mu_l} \left[\frac{4\mu_l \lambda_l l (T_s - T_{ti})}{g \rho_l (\rho_l - \rho_v) (X L_v + 0.68\eta c_p (T_s - T_{ti}))} \right]^{\frac{3}{4}},$$

$$\dot{m}_c = \sum_{i=1}^{i=2} \dot{m}_{ci}, \quad (4.1)$$

$$\dot{Q} = \sum_{i=1}^{i=2} \dot{m}_{ci} (X L_v + 0.68\eta c_p (T_s - T_{ti})),$$

avec $i = 1, 2$ les deux sections de tubes en contact de la vapeur, n_t le nombre de tubes en forme de U, ρ_l et ρ_v sont les masses volumiques du condensât et de la vapeur respectivement, μ_l est la viscosité dynamique du condensât et λ_l représente le coefficient de transfert de chaleur du condensât. T_s , T_{t1} et T_{t2} sont respectivement la température de la vapeur saturée et la température de la surface de condensation des deux parties de tubes en contact de la vapeur, L_v est la chaleur latente de vaporisation de la vapeur, c_p est la capacité thermique spécifique du condensât et X est la qualité de la vapeur. D et l sont respectivement, le diamètre et la longueur du segment de tubes exposés à la vapeur. L'échange de chaleur entre le condensât et les tubes est corrigé par un facteur η en considérant que le débit de condensât n'est pas laminaire et en tenant compte de l'existence de gaz non condensables dans la vapeur [56, 57].

La longueur l des tubes exposés à la vapeur dans chacune des trois sections est calculée par la formule suivante:

$$l = l_t - \Phi_{cg}(\Delta P_l), \quad (4.2)$$

avec l_t la longueur totale de chaque tube en forme de U, ΔP_l la différence de pression due au condensât récupéré (sa masse est donnée par la variable Q_{50}) et Φ_{cg} une fonction non linéaire utilisée pour calculer la longueur des tubes à l'intérieur du condensât en fonction des paramètres géométriques du condenseur (forme, coudes, section, etc.) et de la masse volumique du condensât. Le flux d'enthalpie (\dot{H}_c) transmis par le condensât au liquide stocké au fond du condenseur est:

$$\dot{H}_c = \dot{m}_c (X L_v + c_p T_s) - \dot{Q}. \quad (4.3)$$

L'élément R_{tl} modélise le flux de chaleur $\lambda_{tl} \Phi_{cg}(\Delta P_l)$ entre le liquide stocké et les tubes de refroidissement. λ_{tl} est le coefficient d'échange de chaleur par unité de longueur entre le condensât stocké et la surface métallique des tubes, et Φ_{cg} une fonction non linéaire définie par l'Eq. (4.2).

Les capacités thermiques de l'eau de refroidissement correspondant aux sections 1, 2, et 3 sont représentées par trois éléments de stockage, C_{c1} , C_{c2} , et C_{c3} , respectivement. Les éléments

R_{ct1} , R_{ct2} et R_{ct3} représentent les échanges de chaleur entre le condensât en écoulement et les surfaces internes des tubes de refroidissement.

4.2.3 Modèle bond graph et architectural du générateur de vapeur

La boîte à outils ModelBuilder, présentée au troisième chapitre, permet:

- de construire de façon automatique et par une simple interconnexion des composants le P&ID et le modèle dynamique du processus,
- de générer la liste des RRAs et la matrice de signature des défaillances,
- et de proposer un placement de capteurs permettant d'améliorer la surveillabilité de l'installation.

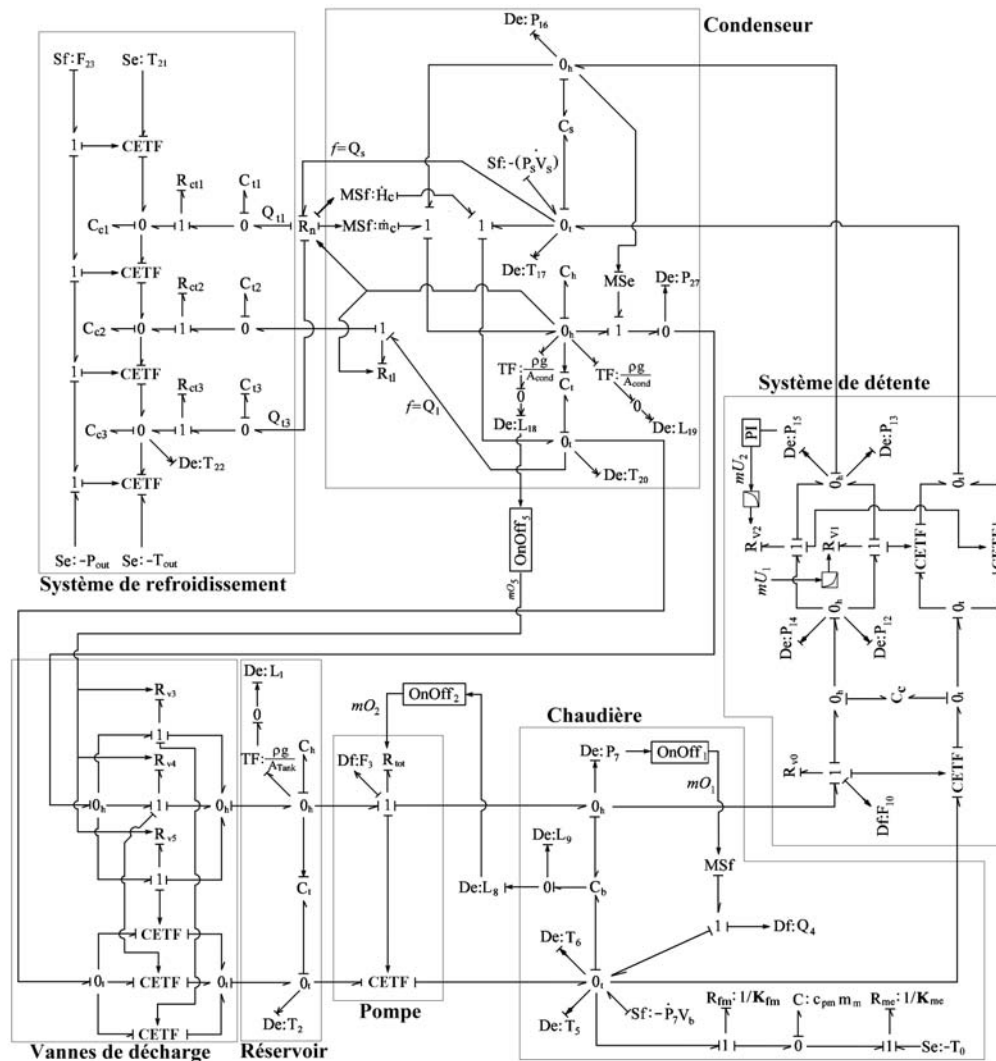


FIG. 4.4 – Modèle bond graph du générateur de vapeur en causalité intégrale préférentielle.

Le modèle bond graph global en causalité intégrale obtenu à partir des sous-modèles des composants du générateur de vapeur est donné à la figure 4.4. Une fois validé, ce modèle est mis en causalité dérivée préférentielle et utilisé pour la génération des RRAs.

Le modèle architectural construit en utilisant ModelBuilder et correspondant au modèle bond graph en causalité intégrale préférentielle est montré à la figure 4.5.

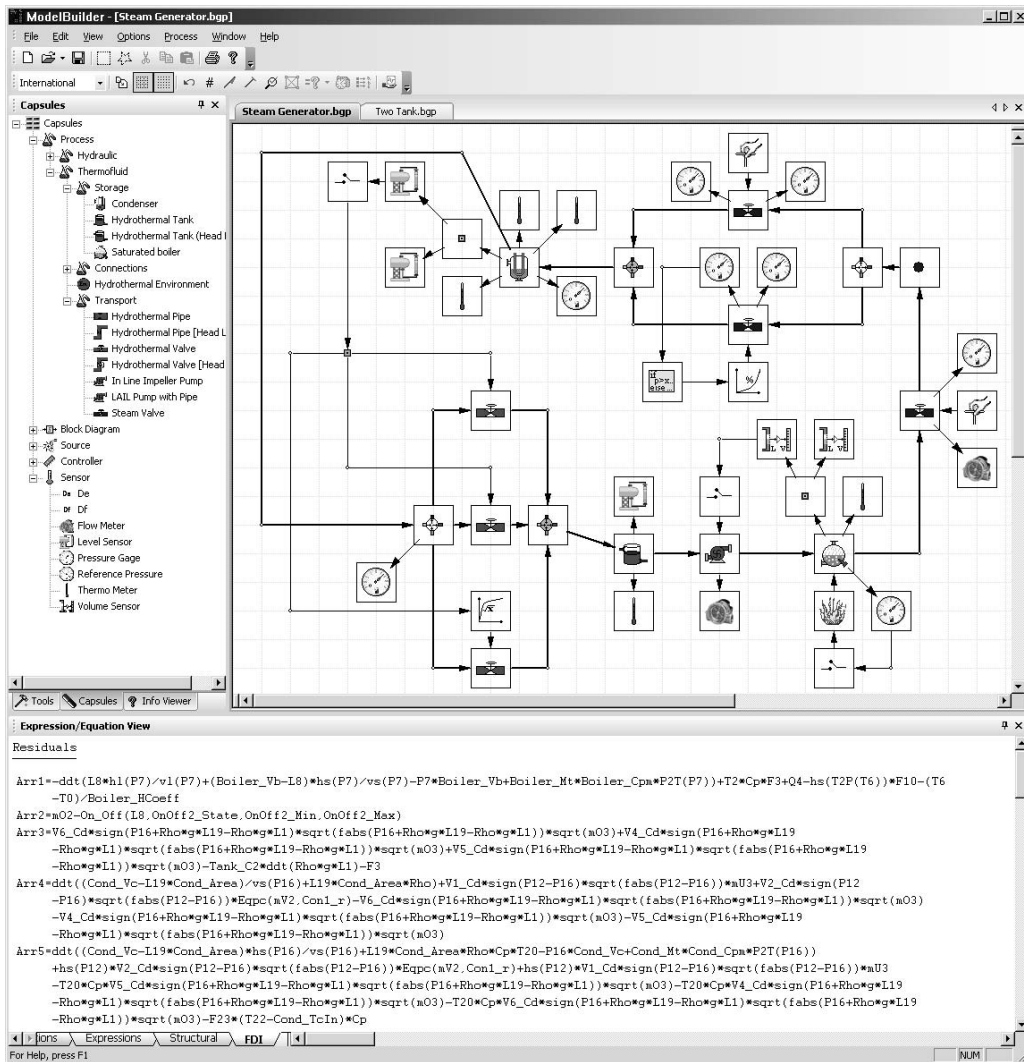























FIG. 4.5 – Modèle architectural du générateur de vapeur généré en utilisant ModelBuilder.

Le modèle architectural obtenu par ModelBuilder est une copie du P&ID de l'installation. Il a été construit en connectant les différentes capsules définies dans le troisième chapitre. La signification de chaque icône (ou capsule) utilisée dans la figure 4.5 est donnée au tableau 4.1.

Icône	Description	Icône	Description
	Réservoir thermo-fluide		Caractéristique égal pourcentage
	Chaudière		Caractéristique "Quick Open"
	Condenseur		Distributeur de flux
	Capacité de couplage		Point d'arrivée des flux
	Pompe thermo-fluide		Distributeur de signal
	Chauffe		Capteur de débit
	Vanne pour vapeur		Capteur de niveau
	Vanne pour liquide		Capteur de pression
	Régulateur logique		Capteur de température
	Régulateur ToR		Capteur de volume
	Régulation manuelle		

TAB. 4.1 – Signification des capsules utilisées dans ModelBuilder

4.2.4 Modèle dynamique et sa validation

Le modèle dynamique de l'installation de générateur de vapeur est généré de façon automatique par ModelBuilder 4.6. Il est donné sous la forme analytique suivante:

$$\begin{aligned}\dot{x} &= f(x,u), \\ y &= g(x,u);\end{aligned}$$

où f et g sont des fonctions nonlinéaires, x est l'état du système, y le vecteur des mesures et u le vecteur d'entrées.

```

Expression/Equation View
dY[23]=BoilerLiquidVolume(Boiler_P.Boiler_X.Boiler_Vb)
dY[24]=T6_mu*Boiler_T
dY[25]=P7_mu*Boiler_P
dY[26]=Cond_SF50-ID5/Cond_Rextv
dY[27]=Cond_SF26-ID3/Cond_Rextv
dY[28]=ID5/Cond_Rextv-Cond_SF70
dY[29]=ID4/Cond_Rext1-Cond_SF63
dY[30]=ID3/Cond_Rextv-Cond_SF55
dY[31]=Cond_SF27-1/Cond_C25/(1/Cond_C23*Y[32])*Y[31]*V5_Cp*V5_mu*V5_Cd*sign(ID12)*sqrt(fabs(ID12))*sqrt(OnOff4_mu1
*On_Off(ID8,OnOff4_State,OnOff4_Min,OnOff4_Max))-1/Cond_C25/(1/Cond_C23*Y[32])*Y[31]*V4_Cp*V4_mu*V4_Cd*sign(ID11)
*sqrt(fabs(ID11))*sqrt(OnOff3_mu1*On_Off(ID8,OnOff3_State,OnOff3_Min,OnOff3_Max))
-1/Cond_C25/(1/Cond_C23*Y[32])*Y[31]*V6_Cp*V6_mu*V6_Cd*sign(ID10)*sqrt(fabs(ID10))*sqrt(OnOff5_mu1
*On_Off(ID8,OnOff5_State,OnOff5_Min,OnOff5_Max))-ID4/Cond_Rext1
dY[32]=Cond_SF20-V6_mu*V6_Cd*sign(ID10)*sqrt(fabs(ID10))*sqrt(OnOff5_mu1*On_Off(ID8,OnOff5_State,OnOff5_Min,OnOff5_Max))
-V4_mu*V4_Cd*sign(ID11)*sqrt(fabs(ID11))*sqrt(OnOff3_mu1*On_Off(ID8,OnOff3_State,OnOff3_Min,OnOff3_Max))
-V5_mu*V5_Cd*sign(ID12)*sqrt(fabs(ID12))*sqrt(OnOff4_mu1*On_Off(ID8,OnOff4_State,OnOff4_Min,OnOff4_Max))
dY[33]=hs(1/Dummy_C2*Y[0])*V3_Cd*sign(ID15)*sqrt(fabs(ID15))*Eqpc(PI_mu1*(PI_Kp*ID9+PI_Ki*intg(ID9)).Con1_r)
+hs(1/Dummy_C2*Y[0])*V2_Cd*sign(ID15)*sqrt(fabs(ID15))*U2_SF1-Cond_SF32
dY[34]=V2_Cd*sign(ID15)*sqrt(fabs(ID15))*U2_SF1+V3_Cd*sign(ID15)*sqrt(fabs(ID15))*Eqpc(PI_mu1*(PI_Kp*ID9+PI_Ki*intg(ID9)).Con1_r)
-Cond_SF20
dY[35]=1/Tank_C1/(1/Tank_C2*Y[38])*Y[37]*Pump_Cp*Pump_mu*(Pump_K1*pow(-(ID1-Pump_Rho*Pump_g*Pump_dZ),Pump_n)
+Pump_K2)*OnOff2_mu1*On_Off(ID6,OnOff2_State,OnOff2_Min,OnOff2_Max)+Heater_mu1*Heater_Power*ID13
-hs(T2P(Boiler_T))*V1_Cd*sign(ID14)*sqrt(fabs(ID14))*U1_SF1-ID2/Boiler_HCoeff
dY[36]=-V1_Cd*sign(ID14)*sqrt(fabs(ID14))*U1_SF1+Pump_mu*(Pump_K1*pow(-(ID1-Pump_Rho*Pump_g*Pump_dZ),Pump_n)
+Pump_K2)*OnOff2_mu1*On_Off(ID6,OnOff2_State,OnOff2_Min,OnOff2_Max)
dY[37]=1/Cond_C25/(1/Cond_C23*Y[32])*Y[31]*V5_Cp*V5_mu*V5_Cd*sign(ID12)*sqrt(fabs(ID12))*sqrt(OnOff4_mu1
*On_Off(ID8,OnOff4_State,OnOff4_Min,OnOff4_Max))+1/Cond_C25/(1/Cond_C23*Y[32])*Y[31]*V4_Cp*V4_mu*V4_Cd*sign(ID11)
*sqrt(fabs(ID11))*sqrt(OnOff3_mu1*On_Off(ID8,OnOff3_State,OnOff3_Min,OnOff3_Max))
+1/Cond_C25/(1/Cond_C23*Y[32])*Y[31]*V6_Cp*V6_mu*V6_Cd*sign(ID10)*sqrt(fabs(ID10))*sqrt(OnOff5_mu1
*On_Off(ID8,OnOff5_State,OnOff5_Min,OnOff5_Max))-1/Tank_C1/(1/Tank_C2*Y[38])*Y[37]*Pump_Cp*Pump_mu*(Pump_K1*pow(-(ID1
-Pump_Rho*Pump_g*Pump_dZ),Pump_n)+Pump_K2)*OnOff2_mu1*On_Off(ID6,OnOff2_State,OnOff2_Min,OnOff2_Max)-Tank_SF97
dY[38]=-Pump_mu*(Pump_K1*pow(-(ID1-Pump_Rho*Pump_g*Pump_dZ),Pump_n)+Pump_K2)*OnOff2_mu1
*On_Off(ID6,OnOff2_State,OnOff2_Min,OnOff2_Max)+V6_mu*V6_Cd*sign(ID10)*sqrt(fabs(ID10))*sqrt(OnOff5_mu1
*On_Off(ID8,OnOff5_State,OnOff5_Min,OnOff5_Max))+V4_mu*V4_Cd*sign(ID11)*sqrt(fabs(ID11))*sqrt(OnOff3_mu1
*On_Off(ID8,OnOff3_State,OnOff3_Min,OnOff3_Max))+V5_mu*V5_Cd*sign(ID12)*sqrt(fabs(ID12))*sqrt(OnOff4_mu1
*On_Off(ID8,OnOff4_State,OnOff4_Min,OnOff4_Max))-Tank_SF98
U[0]=Cond_SF55+Cond_TcIn*Cond_Cp*Cond_SF81
U[1]=Cond_SF63+Cond_SF55+Cond_TcIn*Cond_Cp*Cond_SF81
U[2]=Cond_SF70+Cond_SF63+Cond_SF55+Cond_TcIn*Cond_Cp*Cond_SF81

```

FIG. 4.6 – Équations dynamiques du modèle du générateur de vapeur obtenues en utilisant ModelBuilder.

Les paramètres utilisés dans le modèle et leurs valeurs initiales sont introduits dans ModelBuilder par l'opérateur. En général, ces paramètres sont exprimés dans les modèles bond graphs des composants et sont liés en fonction de la structure du modèle architectural. Une partie de la définition de ces paramètres est montrée à la figure 4.7.

L'intérêt de ce modèle est qu'il peut être utilisé pour élaborer des lois de commande ou encore pour être envoyé aux différentes boîtes à outils du projet. L'échange de données et d'informations est possible grâce à la capacité de ModelBuilder de générer, sous forme de fichier XML (eXtended Markup Language), le modèle dynamique obtenu.

```

Expression/Equation View
Variable | Assignement
Tank_S64 Tank_S64=0;
Tank_S65 Tank_S65=0;
Tank_Cp (< Tank_Cp=Cp;
Tank_T0 (<
Tank_C1 (< Tank_C1=Tank_Area/Tank_g*Tank_Cp;
if (first_time) Tank_O1=Tank_Rbo*Tank_Area*Tank_h0*Tank_Cp*Tank_T0; // O1 is the state for total enthalpy + aCpT
Tank_Leakage
Tank_SF98 Tank_SF98=Tank_Leakage;
Tank_SF97 Tank_SF97=Tank_QLoss+Tank_Leakage*Tank_Cp*Tank_O1/Tank_C1+Tank_C2/Tank_O2;
Boiler_Vb
Boiler_Po
Boiler_Xo
Boiler_err
Boiler_wet Boiler_weight=0;
Boiler_ent Boiler_enthalpy=0;
Boiler_ent Boiler_entropy=0;
Boiler_sat Boiler_saturation_temperature=0;
Boiler_sat Boiler_saturation_pressure=0;
Boiler_deg Boiler_degrees_superheat=0;
Boiler_deg Boiler_degrees_subcooling=0;
Boiler_vis Boiler_viscosity=0;
Boiler_cri Boiler_critical_velocity=0;
Boiler_T ( Boiler_T=0;
Boiler_F0
Boiler_Pat Boiler_Pat=101325.0;
Boiler_Mt
Boiler_Cps
Boiler_X (if (first_time)
{
Boiler_X=Boiler_Xo*100;
Boiler_PO=(Boiler_PO*100000.0+Boiler_Pat)/6894.757; // bar to psi and absolute
Boiler_err=atan(7/(Boiler_T &Boiler_PO &Boiler_X
&Boiler_weight &Boiler_enthalpy &Boiler_entropy
&Boiler_saturation_temperature
&Boiler_saturation_pressure &Boiler_degrees_superheat
&Boiler_degrees_subcooling &Boiler_viscosity
&Boiler_critical_velocity));
Boiler_weight=Boiler_weight*0.028316846/0.45359237;
Boiler_enthalpy=Boiler_enthalpy*1055.06/0.45359237;
Boiler_entropy=Boiler_entropy*1055.06/0.45359237;
Boiler_O2=Boiler_Vb/Boiler_weight;
Boiler_O1=Boiler_O2*Boiler_enthalpy-(Boiler_PO*100000.0+Boiler_Pat)*Boiler_Vb/Boiler_Mt+Boiler_Cps*Boiler_T;
}

```

FIG. 4.7 – Paramètres utilisés dans le modèle dynamique du générateur de vapeur.

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE TB51 (View Source for full doctype...)>
- <TB51 xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="extended">
  <TB_Name>Functional Model Builder for Qualitative and Structural Analysis</TB_Name>
  <TB_Desc>This toolbox deals with model builder for FDI design</TB_Desc>
  - <Input_Data>
    - <Model Name="Steam Generator Final">
      - <Plant-Schematics.Graph-View>
        - <Plant-Items>
          <Plant-Item ID="Tank" BaseClass="Process" Class="Hydrothermal Tank (Head Loss)" Description="" />
          <Plant-Item ID="Boiler" BaseClass="Process" Class="Saturated boiler" Description="" />
          <Plant-Item ID="Cond" BaseClass="Process" Class="Condenser" Description="" />
          <Plant-Item ID="P7" BaseClass="Sensor" Class="Pressure Gage" Description="" />
          <Plant-Item ID="T6" BaseClass="Sensor" Class="Thermo Meter" Description="" />
          <Plant-Item ID="L8" BaseClass="Sensor" Class="Volume Sensor" Description="" />
          <Plant-Item ID="OnOff2" BaseClass="Controller" Class="OnOff Controller" Description="" />
          <Plant-Item ID="T2" BaseClass="Sensor" Class="Thermo Meter" Description="" />
          <Plant-Item ID="L1" BaseClass="Sensor" Class="Level Sensor" Description="" />

```

FIG. 4.8 – Liste des composants du générateur de vapeur et leur classification en format XML.

```

  <Plant-Item ID="L9" BaseClass="Sensor" Class="Volume Sensor" Description="" />
  <Plant-Item ID="OffOn3" BaseClass="Controller" Class="OffOn Controller" Description="" />
</Plant-Items>
- <Topology>
  - <Connection>
    <Source xlink:href="#Tank" xmlns:xlink="http://www.w3.org/1999/xlink" />
    <Target xlink:href="#Pump" xmlns:xlink="http://www.w3.org/1999/xlink" />
  </Connection>
  - <Connection>
    <Source xlink:href="#Tank" xmlns:xlink="http://www.w3.org/1999/xlink" />
    <Target xlink:href="#L1" xmlns:xlink="http://www.w3.org/1999/xlink" />
  </Connection>
  - <Connection>
    <Source xlink:href="#Tank" xmlns:xlink="http://www.w3.org/1999/xlink" />
    <Target xlink:href="#T2" xmlns:xlink="http://www.w3.org/1999/xlink" />
  </Connection>
- <Connection>

```

FIG. 4.9 – Format XML de l'interconnexion des composants dans le modèle architectural.

La structure du modèle (classification, connectivité, équations dynamiques, paramètres), sous forme d'un fichier XML, est générée par ModelBuilder (figures 4.8 et 4.9). Ce fichier peut être envoyé en ligne, sur une simple demande, aux autres boîtes à outils. Par exemple, la TB 7.2 a besoin de l'information sur la connectivité pour construire l'arbre des défaillances utilisé dans la reconfiguration.

Avant d'être utilisé dans la génération des RRAs, le modèle bond graph du générateur de vapeur a été validé à travers des comparaisons entre sa sortie (résultats de simulation) et les observations expérimentales fournies par les différents capteurs.

Les conditions initiales utilisées dans les simulations sont des approximations des valeurs expérimentales en régime permanent correspondant à des valeurs de consignes données. Par exemple, pour la chaudière à une pression initiale P_0 (mesurée par le capteur P_7) et un volume initial L_0 (mesuré par les capteurs L_8 et L_9); les conditions initiales pour les variables d'état m (masse du mélange eau-vapeur) et H (enthalpie du mélange) sont calculées par les relations suivantes:

$$X_0 = \frac{(V_b - L_0) \cdot \nu_l(P_0)}{(V_b - L_0) \cdot \nu_l(P_0) + L_0 \cdot \nu_v(P_0)},$$

$$m = \frac{V_b}{(1 - X_0) \cdot \nu_l(P_0) + X_0 \cdot \nu_v(P_0)},$$

$$H = m \cdot ((1 - X_0) \cdot h_l(P_0) + X_0 \cdot h_v(P_0)).$$

Avec X_0 la qualité de vapeur, V_b le volume total de la chaudière, et $\nu_l()$, $\nu_v()$, $h_l()$ et $h_v()$ représentent respectivement les volumes et les enthalpies spécifiques de l'eau et de la vapeur en fonction de la pression P_0 .

Les résultats de simulation concernant la pression P_7 et le niveau d'eau L_8 , à l'intérieur de la chaudière, sont montrés à la figure 4.10(a). La figure 4.10(b) montre les résultats de simulation de la pression de la vapeur P_{16} à l'entrée du condenseur, de la température de sortie de l'eau de refroidissement T_{22} et de la température du condensât T_{20} au fond du condenseur.

On voit bien sur la figure 4.10(a) que le niveau L_8 dans la chaudière continue de décroître jusqu'à atteindre la valeur de consigne fixée. Ce résultat correspond bien à la réalité puisqu'en chauffant, on génère de la vapeur et donc le niveau diminue. La fréquence de déclenchement de la pompe pour alimenter la chaudière en eau dépend du niveau L_8 et donc de la charge (ouverture et fermeture des vannes du système de détente). Sur la figure 4.10(b), on voit bien qu'initialement, la température du condensât stocké augmente puisqu'elle s'ajoute à celle du liquide qui vient juste de se condenser sur les tubes. L'échange de chaleur à partir du condensât vers la partie submergée des tubes augmente avec l'augmentation du niveau du condensât. En régime établi, la température du condensât stocké tendra vers une valeur limite constante.

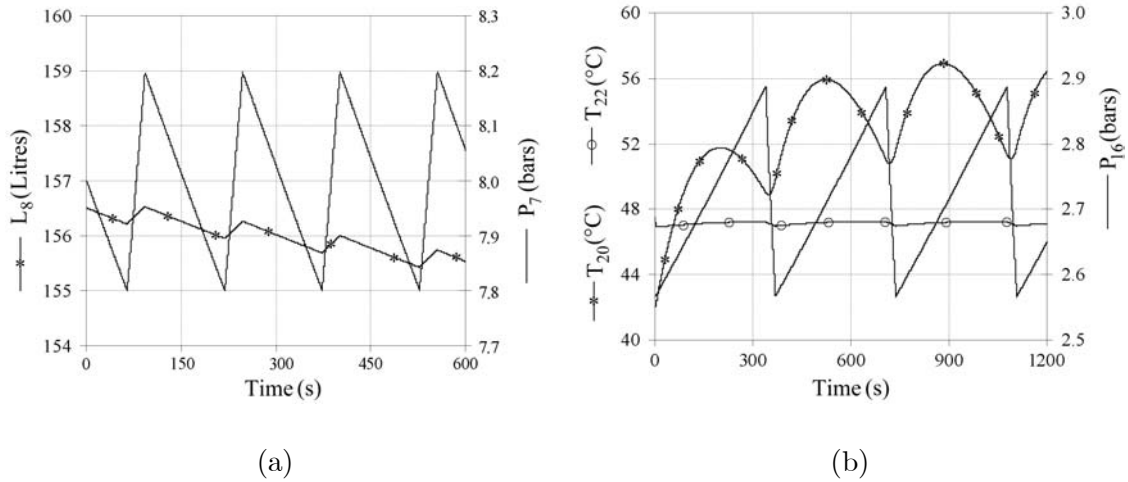


FIG. 4.10 – Résultats de simulation pour (a) la chaudière et (b) le condenseur

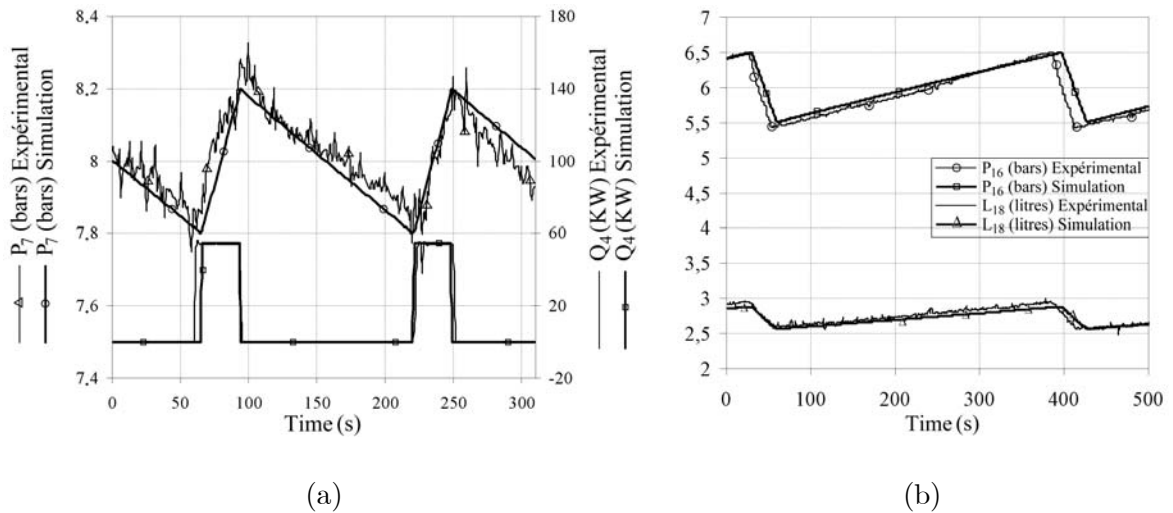


FIG. 4.11 – Comparaison entre les résultats de simulation et expérimentaux pour (a) la chaudière et (b) le condenseur

Les résultats issus des simulations ont été ensuite comparés aux résultats des sorties expérimentales. Une comparaison des résultats correspondant à la chaudière est montrée à la figure 4.11(a). On y observe une bonne concordance entre les résultats de simulation et ceux issus de l'expérience (P_7 et Q_4) d'un côté et une bonne synchronisation entre la fréquence de la chauffe et la dynamique de la pression P_7 de l'autre côté. La comparaison entre les résultats de simulation et expérimentaux correspondant au condenseur est donnée par la figure 4.11(b). Cette comparaison montre une bonne coïncidence en amplitude mais une faible erreur en phase. Cette erreur est attribuée aux bruits des mesures et aux retards dans les réactions des actionneurs (ouverture et fermeture des vannes).

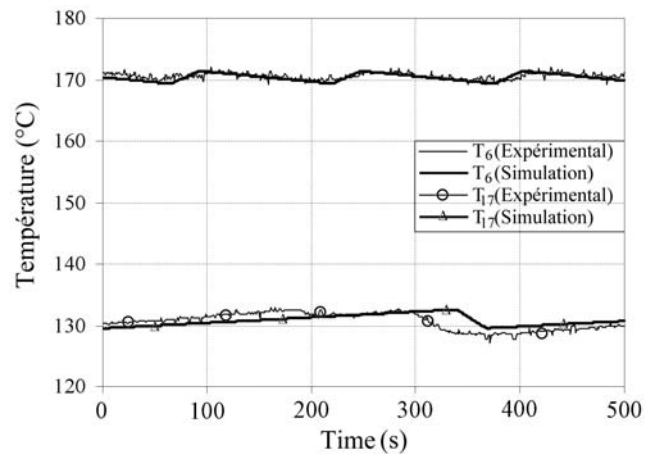


FIG. 4.12 – Comparaison entre les températures simulées et expérimentales de la chaudière et du condenseur

La figure 4.12 montre sur un même graphe les valeurs expérimentales et de simulation de la température dans la chaudière (T_6) et celles dans le condenseur (T_{17}). On a constaté que la température (expérimentale) de l'eau dans la chaudière est approximativement de 3 à 4 °C (de 1.8 à 2.3%) inférieure à la température obtenue par les simulations. Cette différence est due aux hypothèses de modélisation formulées précédemment et dans lesquelles l'eau et la vapeur ont été considérées comme un mélange homogène. À noter que ces erreurs seront prises en compte dans l'élaboration du système de surveillance de l'installation.

4.2.5 Génération de RRAs

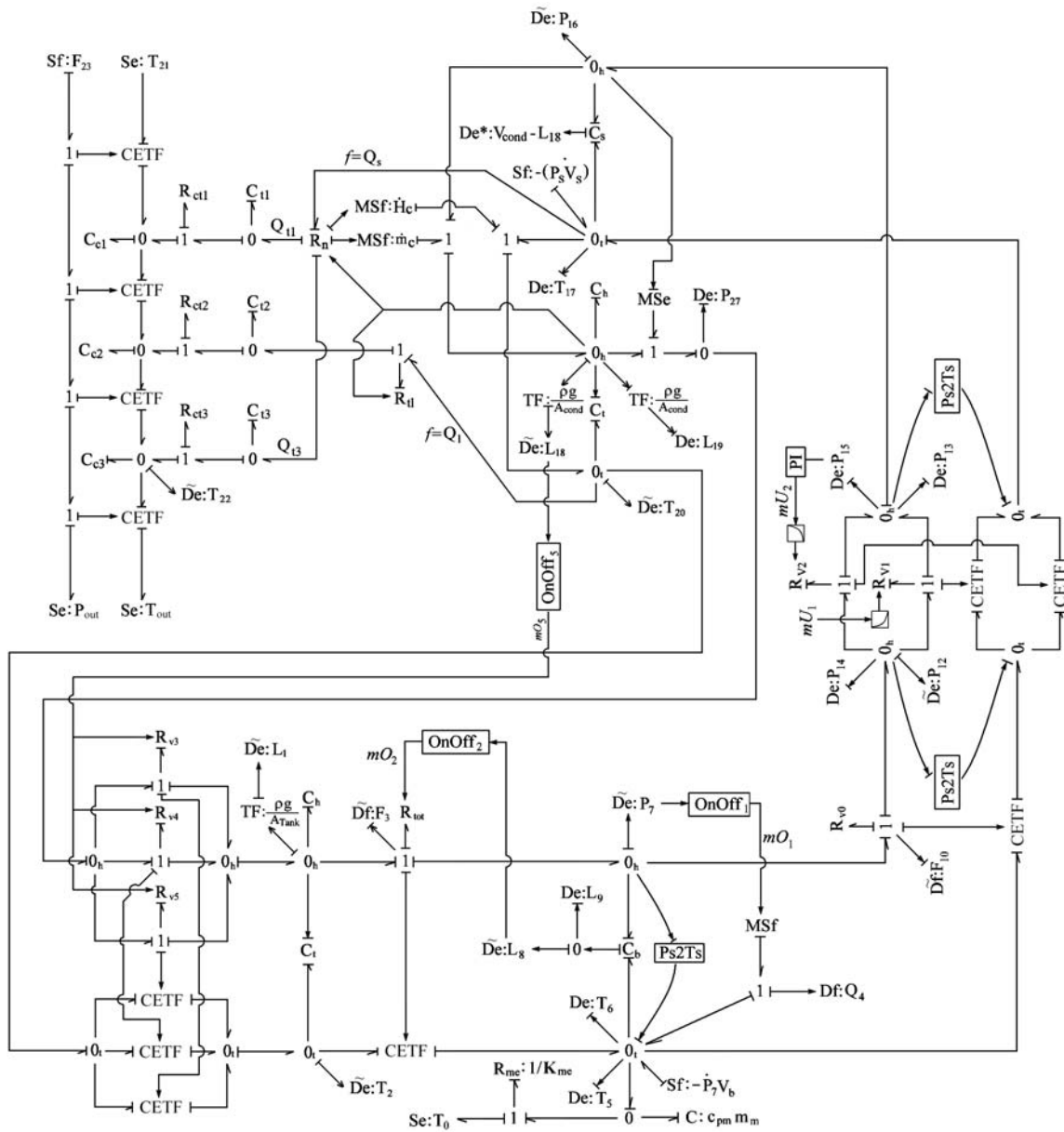


FIG. 4.13 – Le modèle pseudo-bond graph global en causalité dérivée préférentielle du générateur de vapeur

Les relations de redondance analytique concernant le générateur de vapeur sont générées à partir de son modèle bond graph global en causalité dérivée préférentielle. En plus de la causalité dérivée affectée aux éléments de stockage et aux détecteurs, quelques contraintes statiques entre variables ont été ajoutées sur le modèle bond graph (comme par exemple la

relation qui existe entre la pression et la température de la vapeur saturée).

Le modèle bond graph global en causalité dérivée préférentielle est montré à la figure 4.13.

La pression et la température de la vapeur saturée sont liées par les conditions de l'équilibre thermodynamique. Sur le modèle bond graph de la figure 4.13, la température de la vapeur est donnée en fonction de la pression et représentée par le bloc $Ps2Ts$. De même, sachant que le volume du condenseur est constant, la mesure du volume du condensât, par les capteurs L_{18} et L_{19} , permet de calculer le volume de la vapeur à l'intérieur du condenseur.

On a vu, au deuxième chapitre de ce mémoire, que lorsque la causalité d'un détecteur ne peut pas être inversée, ce dernier est alors matériellement redondant. Les redondances matérielles présentes sur l'installation du générateur de vapeur sont déduites directement à partir du modèle bond graph en causalité dérivée. Ces redondances sont données par le tableau 4.2.

Capteur redondant	Redondance avec?
L_9	L_8
T_6	P_7
T_5	P_7
P_{14}	P_{12}
P_{13}	P_{16}
P_{15}	P_{16}
T_{17}	P_{16}
L_{19}	L_{18}
P_{27}	P_{16}, L_{18}

TAB. 4.2 – Les capteurs redondants du générateur de vapeur

Les neuf redondances matérielles du tableau 4.2 conduisent à neuf RRAs statiques. Ces dernières améliorent l'isolabilité des défaillances et fournissent des moyens pour la reconfiguration et la commande tolérante aux fautes.

Les RRAs, au nombre de 23, du générateur de vapeur sont automatiquement et systématiquement générées à partir du modèle architectural de l'installation (figure 4.14).

A noter que les RRAs du générateur de vapeur ont été aussi générées manuellement [21] et correspondent exactement à celles obtenues par ModelBuilder. La liste des RRAs (et les résidus correspondants) peut être envoyée à tout moment et sur une simple requête aux

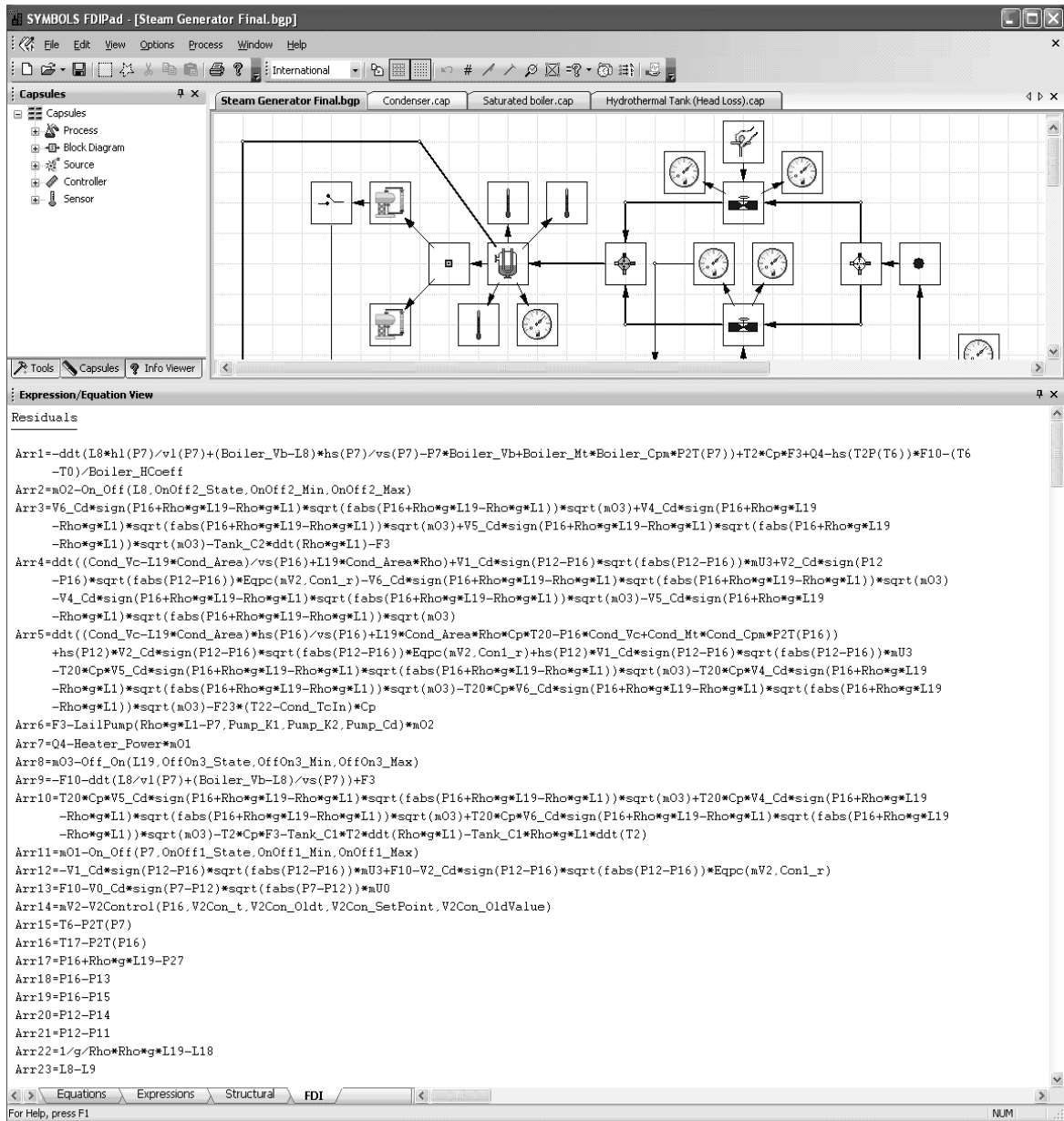


FIG. 4.14 – Liste des RRAs pour l'installation du générateur de vapeur obtenue en utilisant ModelBuilder.

différentes boîtes à outils du projet CHEM.

Les résidus générés pour le générateur de vapeur sont des résidus déterministes mais peuvent être exploités par la TB 5.2 (SBT) pour traiter des incertitudes dans les paramètres et les mesures. Dans ce cas, un fichier XML (figure 4.15) contenant tous les résidus est généré et envoyé à la TB 5.2.

```

</Bond-Graph>
</Global_Model>
<Functional-Model />
<Behavioral-Model />
<Structural-Model />
- <Residuals>
  <Equation>-ddt(L8*hl(P7)/vl(P7)+(Boiler_Vb-L8)*hs(P7)/vs(P7)-P7*Boiler_Vb+Boiler_Mt*Boiler_Cpm*P2T(P7))+T2*Cp*F3+Q4-hs
  (T2P(T6))*F10-(T6-T0)/Boiler_HCcoeff=0</Equation>
  <Equation>m02-On_Off(L8,OnOff2_State,OnOff2_Min,OnOff2_Max)=0</Equation>
  <Equation>V6_Cd*sign(P16+Rho*g*L19-Rho*g*L1)*sqrt(fabs(P16+Rho*g*L19-Rho*g*L1))*sqrt(m03)+V4_Cd*sign(P16+Rho*g*L19-
  Rho*g*L1)*sqrt(fabs(P16+Rho*g*L19-Rho*g*L1))*sqrt(m03)+V5_Cd*sign(P16+Rho*g*L19-Rho*g*L1)*sqrt(fabs(P16+Rho*g*L19-
  Rho*g*L1))*sqrt(m03)-Tank_C2*ddt(Rho*g*L1)-F3=0</Equation>
  <Equation>ddt((Cond_Vc-L19*Cond_Area)/vs(P16)+L19*Cond_Area*Rho)+V1_Cd*sign(P12-P16)*sqrt(fabs(P12-P16))
  *mU3+V2_Cd*sign(P12-P16)*sqrt(fabs(P12-P16))*Eqpc(mV2,Con1_r)-V6_Cd*sign(P16+Rho*g*L19-Rho*g*L1)*sqrt(fabs
  (P16+Rho*g*L19-Rho*g*L1))*sqrt(m03)-V4_Cd*sign(P16+Rho*g*L19-Rho*g*L1)*sqrt(fabs(P16+Rho*g*L19-Rho*g*L1))*sqrt
  (m03)-V5_Cd*sign(P16+Rho*g*L19-Rho*g*L1)*sqrt(fabs(P16+Rho*g*L19-Rho*g*L1))*sqrt(m03)=0</Equation>
  <Equation>ddt((Cond_Vc-L19*Cond_Area)*hs(P16)/vs(P16)+L19*Cond_Area*Rho*Cp*T20-P16*Cond_Vc+Cond_Mt*Cond_Cpm*P2T
  (P16))+hs(P12)*V2_Cd*sign(P12-P16)*sqrt(fabs(P12-P16))*Eqpc(mV2,Con1_r)+hs(P12)*V1_Cd*sign(P12-P16)*sqrt(fabs(P12-
  P16))*mU3-T20*Cp*V5_Cd*sign(P16+Rho*g*L19-Rho*g*L1)*sqrt(fabs(P16+Rho*g*L19-Rho*g*L1))*sqrt(m03)-
  T20*Cp*V4_Cd*sign(P16+Rho*g*L19-Rho*g*L1)*sqrt(fabs(P16+Rho*g*L19-Rho*g*L1))*sqrt(m03)-T20*Cp*V6_Cd*sign
  (P16+Rho*g*L19-Rho*g*L1)*sqrt(fabs(P16+Rho*g*L19-Rho*g*L1))*sqrt(m03)-F23*(T22-Cond_Tcn)*Cp=0</Equation>
  <Equation>F3-LailPump(Rho*g*L1-P7,Pump_K1,Pump_K2,Pump_Cd)*m02=0</Equation>
  <Equation>Q4-Heater_Power*m01=0</Equation>
  <Equation>m03-Off_On(L19,OffOn3_State,OffOn3_Min,OffOn3_Max)=0</Equation>
  <Equation>-F10-ddt(L8/vl(P7)+(Boiler_Vb-L8)/vs(P7))+F3=0</Equation>
  <Equation>T20*Cp*V5_Cd*sign(P16+Rho*g*L19-Rho*g*L1)*sqrt(fabs(P16+Rho*g*L19-Rho*g*L1))*sqrt(m03)+T20*Cp*V4_Cd*sign
  (P16+Rho*g*L19-Rho*g*L1)*sqrt(fabs(P16+Rho*g*L19-Rho*g*L1))*sqrt(m03)+T20*Cp*V6_Cd*sign(P16+Rho*g*L19-Rho*g*L1)
  *sqrt(fabs(P16+Rho*g*L19-Rho*g*L1))*sqrt(m03)-T2*Cp*F3-Tank_C1*T2*ddt(Rho*g*L1)-Tank_C1*Rho*g*L1*ddt(T2)
  =0</Equation>
  <Equation>m01-On_Off(P7,OnOff1_State,OnOff1_Min,OnOff1_Max)=0</Equation>
  <Equation>-V1_Cd*sign(P12-P16)*sqrt(fabs(P12-P16))*mU3+F10-V2_Cd*sign(P12-P16)*sqrt(fabs(P12-P16))*Eqpc(mV2,Con1_r)
  =0</Equation>

```

FIG. 4.15 – La structure en format XML des RRAs du générateur de vapeur.

4.2.6 Surveillabilité et placement de capteurs

L'analyse structurelle effectuée sur les RRAs permet de construire la matrice de signature des défaillances. Pour cela, le cahier des charges doit être respecté. En effet, il arrive que des composants du système soient par définition non faillibles et donc exclus de la matrice des signatures. Concernant l'installation du générateur de vapeur, les régulateurs sont des lois de commande programmées (donc logiciels) et leur valeur de retour sont considérées comme des sorties de capteurs logiciels. Elles sont donc considérées comme des composants infaillibles de l'installation. Les composants exclus de l'analyse de surveillabilité sont spécifiés dans ModelBuilder par l'opérateur comme le montre la figure 4.16.

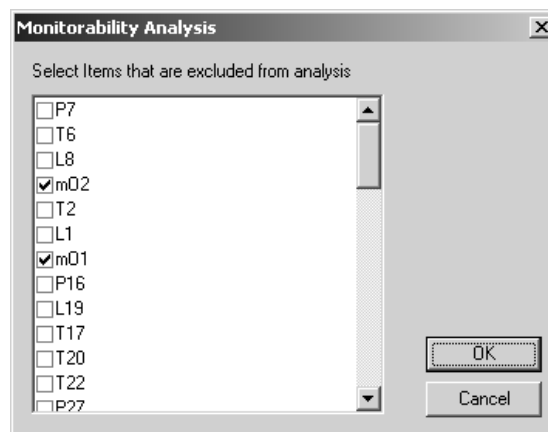


FIG. 4.16 – Spécification sur ModelBuilder des composants à surveiller

Sur la matrice de signature des défaillances de la figure 4.17 on constate que, théorique-

ment, toutes les défaillances des composants sont détectables et isolables sauf celles relatives aux composants (V_1, V_2) et $(V_3, V_4$ et $V_5)$. En effet, les termes entre parenthèses ont les mêmes signatures. A partir de cette observation, on peut proposer de placer des capteurs supplémentaires sur ces composants pour isoler leurs défaillances. Pratiquement, cela se traduit par la génération à nouveau des RRAs et de la matrice des signatures. Refaire ce travail manuellement devient alors fastidieux. C'est pour cela que la boîte à outils ModelBuilder est très souhaitable pour générer automatiquement et avec beaucoup de flexibilité les indicateurs de défaillance. En outre, ModelBuilder nous permet de générer sous forme d'un fichier XML la matrice de signature des défaillances (figure 4.18).

	Mb	Ib	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23
P7	1	1	1	0	0	0	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
T6	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
L0	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T2	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I1	1	1	0	0	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
P16	1	1	0	0	1	1	1	0	0	0	0	1	0	1	0	1	1	1	1	1	0	0	0	0	0
L19	1	1	0	0	1	1	1	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0
T17	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
T20	1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
T22	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P27	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
P13	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
P12	1	1	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0
P15	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P14	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
F3	1	1	1	0	1	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
F10	1	1	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0
P11	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Q4	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mV2	1	1	0	0	0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
L18	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
L9	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Tank	1	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Boiler	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Cond	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OnOff2	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OnOff1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
V3	1	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V4	1	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V5	1	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Heater	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
V1	1	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
V2	1	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
V2Con	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Pump	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OffOn3	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FIG. 4.17 – La matrice de signature des défaillances du générateur de vapeur obtenue en utilisant ModelBuilder.

4.2.7 Implémentation en temps réel

Dans cette section est présentée une implémentation en temps réel des résidus du générateur de vapeur et dont les équations sont données dans [21]. ModelBuilder génère les mêmes équations des résidus mais dans un ordre différent.



FIG. 4.20 – L'installation du générateur de vapeur

4.2.7.1 Matrice pratique de signature des défaillances

Dans un système de surveillance, il existe deux types de matrices de signature des défaillances: une matrice théorique qui est déduite directement de l'analyse des RRAs et une autre matrice pratique qui est réellement implémentée et utilisée dans les procédures de détection et de localisation des défaillances. La principale différence entre ces deux matrices est que la matrice pratique tient compte de la sensibilité des résidus par rapport aux variations et ordres des mesures et des paramètres.

Considérons le résidu r_3 correspondant à la RRA_3 suivante:

$$\begin{aligned}
 RRA_3 : F_3 - \frac{(V_7 - C_d)^2}{2.K_1} . mO_2 \\
 + \frac{V_7 - C_d \cdot \sqrt{(V_7 - C_d)^2 - 4.K_1 \cdot \left(K_2 + K_1 \cdot (P_7 - P_0 - \rho \cdot g \cdot \frac{L_1}{A_{Tank}}) \right)}}{2.K_1} . mO_2 = 0.
 \end{aligned} \tag{4.4}$$

Théoriquement, ce résidu est fonction de la caractéristique de la pompe (K_1 et K_2), de la vanne V_7 ($V_7 - C_d$), des mesures des capteurs de niveau (L_1) dans le réservoir, de la pression

(P_7) dans la chaudière et du débit de la pompe (F_3). En régime établi du générateur de vapeur, on a $P_7 = 8.10^5 Pa$ et $\frac{L_1}{A_{Tank}} \leq 1 m$. Le terme $\rho.g.\frac{L_1}{A_{Tank}} \leq 10^4 Pa$ peut donc être négligé dans le résidu devant la valeur de P_7 . Pratiquement, l'effet de L_1 et A_{Tank} dans l'Eq. (4.4) est même inférieur à l'influence des incertitudes de mesure de P_7 (la précision du capteur étant de $\pm 1.5\%$). Sur la matrice de signature des défaillances S_{ji} , on peut alors écrire: $S_{L_1,3} = S_{Tank,3} = 0$, ce qui signifie que le résidu r_3 est insensible aux défaillances du capteur L_1 et à celles du réservoir (fuites, etc.).

L'analyse de la sensibilité de chaque résidu se poursuit pour construire la signature pratique de chaque composant. Cette analyse est réalisée sur la base des résultats de simulation et des données en hors ligne acquises lors des différents modes de fonctionnement de l'installation du générateur de vapeur. Pour revenir au résidu r_3 , seules les défaillances du système d'alimentation (représenté par la pompe et la conduite) et des capteurs P_7 et F_3 peuvent ramener sa valeur au delà du seuil ε_3 de fonctionnement normal fixé. La structure de r_3 correspond alors à celle donnée au tableau 4.3. La même logique est appliquée à tous les résidus pour obtenir la matrice pratique de signature des défaillances donnée par le tableau 4.4.

Résidu	Pompes (K_1 et K_2)	P_7	F_3	Autres composants
r_3	1	1	1	0 0.....0

TAB. 4.3 – Structure du résidu R_3

En analysant la matrice de signature des défaillances du générateur de vapeur, on constate que le capteur L_1 et le composant réservoir ont des signatures identiques. On peut donc dire que les défaillances affectant ces deux composants ne sont pas isolables. Un détecteur d'effort supplémentaire (e.g. capteur de niveau) dans cette partie de l'installation peut alors remédier à ce problème.

De même, les vannes V_1 et V_2 ont des signatures identiques et par conséquent leurs défaillances ne sont pas isolables. En analysant la structure causale du modèle bond graph en causalité dérivée, on peut conclure qu'un seul détecteur de flux (e.g. capteur de débit) supplémentaire sur le système de détente suffit pour localiser les défaillances de ce sous-système. Ceci étant le résultat de ce qu'on a vu dans le premier chapitre de ce mémoire. En effet, le système de détente est composé d'éléments R connectés à des jonctions 1 sur lesquels seuls des détecteurs de flux sont permis.

Les défaillances des vannes V_3 , V_4 et V_5 ne sont pas aussi isolables. Des capteurs supplémentaires sont alors nécessaires pour améliorer leur surveillance.

	D _b	I _b	r ₁	r ₂	r ₃	r ₄	r ₅	r ₆	r ₇	r ₈	r ₉	r ₁₀	r ₁₁	r ₁₂	r ₁₃	r ₁₄	r ₁₅	r ₁₆	r ₁₇	r ₁₈	r ₁₉	r ₂₀	r ₂₁	r ₂₂	r ₂₃
L ₁	1	0	1	1																					
T ₂	1	1		1			1																		
F ₃	1	1	1	1	1	1	1																		
L ₈	1	1					1	1																	1
L ₉	1	1																							1
P ₇	1	1			1	1	1		1									1							
Q ₄	1	1					1	1																	
T ₅	1	1															1								
T ₆	1	1															1	1							
F ₁₀	1	1				1	1		1	1															
P ₁₂	1	1							1	1	1	1								1					
P ₁₃	1	1																		1					
P ₁₄	1	1																		1					
P ₁₅	1	1																			1				
P ₁₆	1	1	1	1						1	1	1								1	1	1	1		
T ₁₇	1	1																					1		
L ₁₈	1	1									1	1										1			1
L ₁₉	1	1																							1
P ₂₇	1	1																				1			
T ₂₉	1	1		1								1													
Réservoir	1	0	1	1																					
Pompe	1	1			1																				
Chaudière	1	1				1	1																		
Condenseur	1	1									1	1													
Chauffé	1	1						1																	
OnOff ₁	1	1											1												
OnOff ₂	1	1												1											
OnOff ₅	1	1														1									
PI	1	1													1										
V ₀	1	1							1																
V ₁	1	0								1	1	1													
V ₂	1	0								1	1	1													
V ₃	1	0	1	1							1	1													
V ₄	1	0	1	1							1	1													
V ₅	1	0	1	1							1	1													

TAB. 4.4 – Matrice de signature des défaillances du générateur de vapeur

4.2.7.2 Procédure de décision

Généralement, la logique adoptée pour conclure si oui ou non une défaillance est apparue, dépend du type de connaissance disponible sur le processus, de l'approche utilisée pour générer les indicateurs de défaillances et du type des défaillances traitées. Pour plus de détails, un grand nombre d'algorithmes de décision est publié dans [45].

Une fois la procédure de décision choisie, elle est appliquée sur l'ensemble des résidus pour générer le vecteur de cohérence binaire $V = [v_1 v_2 \dots v_n]$ (2.4). Il y a alors présence d'une défaillance lorsqu'au moins un élément v_i de V est égal à 1 (i.e. $V \neq [0 \ 0 \dots 0]$).

Théoriquement, dans le cas où le système est en bon fonctionnement, la valeur de chaque résidu r_i ($i = 1 \dots n$) doit être nulle. Mais en pratique, ceci n'est pas toujours le cas.

Concernant le générateur de vapeur, le mode de fonctionnement normal vérifie la condition suivante:

$$|\Psi_{\Delta T}(r_i)| \leq \varepsilon_i,$$

avec $\Psi_{\Delta T}$ un pré-traitement sur chaque résidu (une moyenne glissante sur un intervalle de

temps ΔT). Comme expliqué dans le deuxième chapitre, ε_i est dû aux erreurs de modélisation ainsi qu'aux bruit des mesures.

Dans l'application du générateur de vapeur, le problème est alors de déterminer le seuil ε_i au delà duquel les résidus traités ne sont plus cohérents avec la condition de fonctionnement normal du processus. La valeur choisie de ε_i ne doit pas être assez élevée pour éviter les non-détections et pas assez faible pour éviter les fausses alarmes.

En outre, la valeur de chaque résidu traité est normalisée par rapport à une valeur maximale (correspondant à l'amplitude du résidu en présence de défaillance). Pour déterminer le seuil de chaque résidu, une base de données représentatives a été enregistrée pendant différents modes de fonctionnements du générateur de vapeur. Elle est ensuite utilisée en hors ligne pour tester et vérifier les réponses des résidus et pour déterminer pour chacun d'eux la valeur correspondante du seuil ε .

Une moyenne glissante sur une fenêtre temporelle de 10 s a été appliquée sur tous les résidus. Ce traitement permet d'atténuer les pics correspondant aux bruits et aux variables présentes sous forme de dérivées dans les relations des résidus. Les seuils choisis pour les résidus normalisés de l'installation de générateur de vapeur sont donnés par le tableau 4.5, avec les facteurs de normalisation ζ correspondants.

r_i	ζ_i	ε_i	r_i	ζ_i	ε_i	r_i	ζ_i	ε_i
r_1	1	0.2	r_9	1	0.025	r_{17}	10^5	0.2
r_2	8.3×10^4	0.2	r_{10}	2.67×10^6	0.0019	r_{18}	10^5	0.2
r_3	0.25	0.4	r_{11}	1	0.5	r_{19}	10^5	0.2
r_4	1	1.0	r_{12}	1	0.5	r_{20}	10^5	0.2
r_5	2.67×10^6	0.1	r_{13}	1	0.5	r_{21}	10	1.0
r_6	5.5×10^4	0.5	r_{14}	1	0.5	r_{22}	1	0.05
r_7	0.022	0.225	r_{15}	10	1.0	r_{23}	1	1.0
r_8	0.022	0.675	r_{16}	10	1.0			

TAB. 4.5 – Les seuils expérimentaux des résidus après une moyenne glissante de 10s

4.2.7.3 Résultats expérimentaux

Les résultats expérimentaux présentés dans les sections suivantes ont été obtenus à partir d'un fonctionnement réel du générateur de vapeur pour une pression de la chaudière réglée à

une valeur de 8 bars. Le niveau d'eau L_8 (calibré en volume) dans la chaudière est maintenu à 154 ± 3 litres, la pression P_{15} en aval de la vanne V_2 est réglée à 2.5 bars et le niveau L_{18} du condensât à 6 ± 0.5 litres. L'acquisition des données issues des capteurs et l'envoi des commandes sont effectués au rythme d'une seconde d'intervalle. Les alarmes générées, suite à la présence de défaillances dans le générateur de vapeur, sont signalées en temps réel à l'opérateur. Ce dernier peut avoir le privilège et la permission, selon les autorisations (administrateur, technicien, utilisateur invité, etc.), d'acquiescer ou d'admettre une alarme qui lui est signalée et ce en fonction de son expérience de l'installation.

Les défaillances dans l'installation du générateur de vapeur ont été classées en quatre groupes:

- les défaillances physiques du processus,
- les défaillances des actionneurs,
- les défaillances des capteurs,
- et les défaillances des régulateurs.

Défaillances physiques du processus Ce sont des défaillances qui peuvent survenir sur le générateur de vapeur telle que: l'usure des vannes, une fuite dans la chaudière, dans le condenseur ou dans le réservoir, des obstructions dans les conduites et les vannes, etc..

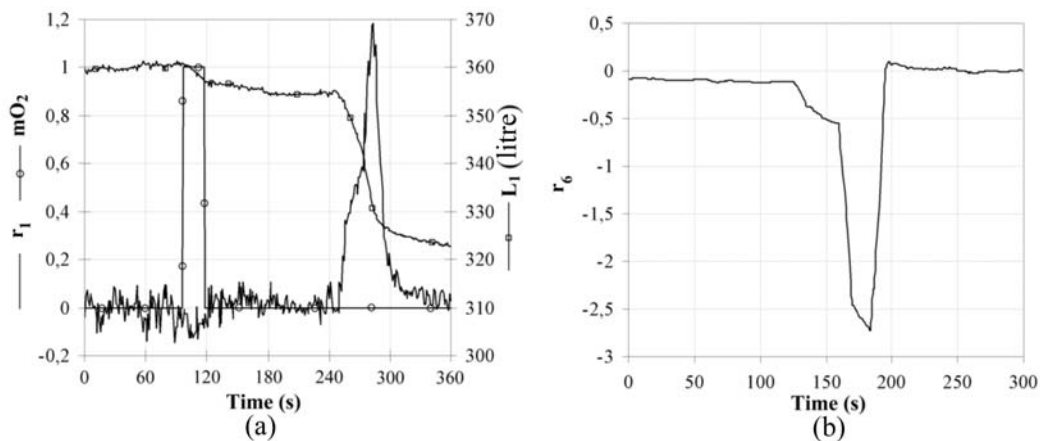


FIG. 4.21 – Détection d'une (a) fuite rapide et (b) obstruction totale de la sortie de la chaudière

Deux sortes de défaillances physiques sont traitées dans cette section. Il s'agit d'une fuite d'eau du réservoir de stockage et d'une obstruction au niveau de la vanne V_0 située à la sortie de la chaudière. Le rythme de la fuite peut être lent ou rapide. Lorsque la fuite est rapide (dans notre cas d'un débit d'environ 0.4 l/s pour la fenêtre temporelle 250 – 300 s), elle peut être détectée par une simple comparaison entre la valeur normalisée du résidu r_1 et son seuil

correspondant ($\varepsilon_1 = 0.2$) comme le montre la figure 4.21(a). De même, durant toute la durée de la fuite d'eau, l'amplitude du résidu r_2 excède le seuil ε_2 . Pendant ce temps, les autres résidus restants ne dépassent pas leurs seuils respectifs. A partir de cette fuite on peut générer un vecteur de cohérence $V = [1,1,0,\dots,0]$ qui a deux coïncidences dans la matrice de signature des défaillances donnée au tableau 4.4. Ces coïncidences correspondent au composant 'Tank' et au capteur ' L_1 '. On peut alors conclure que même si cette défaillance est bien détectable, sa localisation (isolation) n'est pas possible en utilisant un simple test par rapport au seuil. On peut observer sur la figure 4.21(a) que durant le fonctionnement normal de la pompe ($mO_2 = 1$), la variation de la valeur du niveau L_1 est normale et n'affecte pas l'amplitude des résidus.

Le même test par rapport au seuil est appliqué sur le résidu r_7 , comme le montre la figure 4.21(b), pour détecter le blocage total de la vanne V_0 dans la fenêtre temporelle allant de 120 à 180 s. Cette défaillance qui affecte seulement le résidu r_7 et engendre une signature unique, $V = [0,0,0,0,0,0,1,0,\dots,0]$, est donc facile à détecter et à localiser.

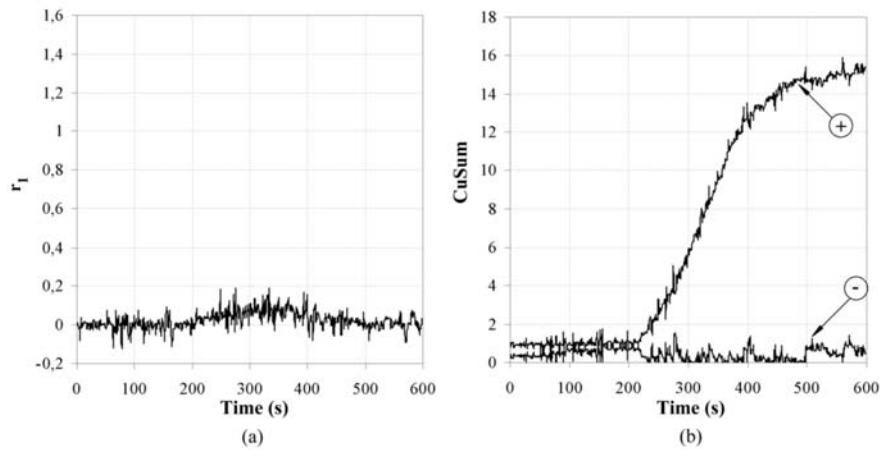


FIG. 4.22 – Détection d'une faible fuite d'eau du réservoir: (a) moyenne glissante (b) $CuSum_2$.

La détection d'une fuite d'eau (à faible débit) du réservoir (environ 0.03 l/s , sur la fenêtre temporelle allant de 180 à 480 s) peut ne pas être possible par un simple test de seuil (figure 4.22(a)). Selon la matrice de signature des défaillances, la fuite dans le réservoir doit provoquer des changements seulement dans les résidus r_1 et r_2 . La fuite peut être interprétée comme une variation du niveau d'eau dans le réservoir, donnée par les mesures du capteur L_1 et qui est présent sous forme de dérivée dans les résidus correspondants. Donc, la variation de l'amplitude des résidus r_1 et r_2 ne doit pas être importante (figure 4.22(a)). Afin de détecter ce genre de fuite, il est nécessaire de fixer une très faible valeur de seuil pour les résidus r_1 et r_2 . Mais un faible seuil rend l'algorithme de détection très sensible aux perturbations

ce qui provoque de nombreuses fausses alarmes. Pour détecter de telles défaillances, nous avons utilisé l'algorithme de la somme cumulée CuSum₂ [45]. En appliquant cet algorithme, le temps t_a de déclenchement d'une alarme suite à une défaillance est donnée par les relations suivantes:

$$\begin{cases} g_0^+ = g_0^- = 0, \\ g_k^+ = \sup[0, (g_{k-1}^+ + r_{ik} - \mu_0 - \frac{\nu}{2})], \\ g_k^- = \sup[0, (g_{k-1}^- - r_{ik} + \mu_0 - \frac{\nu}{2})], \\ t_a = \min \{k : (g_k^+ \geq h) \cup (g_k^- \geq h)\}, \end{cases} \quad (4.5)$$

où g_k^+ et g_k^- sont respectivement les sommes cumulées positive et négative après k échantillons, h est le seuil de déclenchement fixé pour un résidu r_i , r_{ik} est la valeur du $i^{\text{ème}}$ résidu à l'instant k , μ_{0i} est la valeur moyenne du $i^{\text{ème}}$ résidu en l'absence de défaillance et ν est l'écart type.

Le résultat de l'implémentation de l'algorithme CuSum₂ est montré à la figure 4.22(b). Les signes \oplus et \ominus correspondent respectivement aux sommes cumulées positive (g^+) et négative (g^-) du résidu r_1 pour une valeur de $\nu = 0.05$. Le faible débit de la fuite est détecté lorsque la partie positive (g^+) de CuSum₂ est supérieur au seuil ($h_c = 2$) déterminé expérimentalement.

	r_1	r_2	r_4	r_5	r_9	r_{10}
ν	0.05	0.05	0.1	0.05	0.005	0.001
h_c	2	2	5	1	0.3	0.02

TAB. 4.6 – Seuils choisis pour l'algorithme CuSum₂

Dans l'application du générateur de vapeur, l'algorithme CuSum₂ est utilisé pour construire les éléments v_i du vecteur de cohérence V correspondant à tous les résidus comportant des formes dérivées des mesures. Les résidus, leur seuils CuSum₂ correspondants (h_{ci} pour le $i^{\text{ème}}$ résidu) et les valeurs de ν choisies sont données par le tableau 4.6.

La réponse des résidus r_9 et r_{10} à une fuite (à faible débit: 0.05 l/s) du condenseur, pendant la durée allant de 150 à 210 s, est montrée à la figure 4.23. Les parties positives de CuSum₂ correspondant aux résidus r_9 et r_{10} excèdent leur seuils fixés. En outre, on constate que ces deux résidus sont affectés par la défaillance. Cette dernière est donc détectable et isolable.

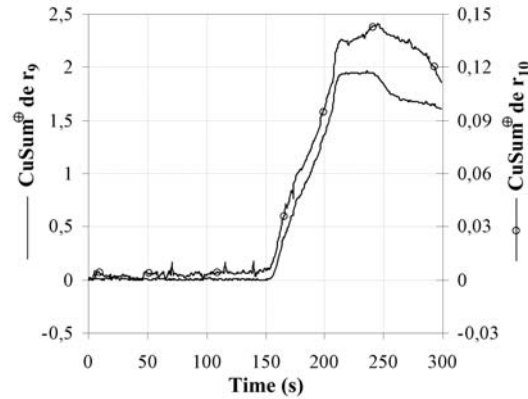
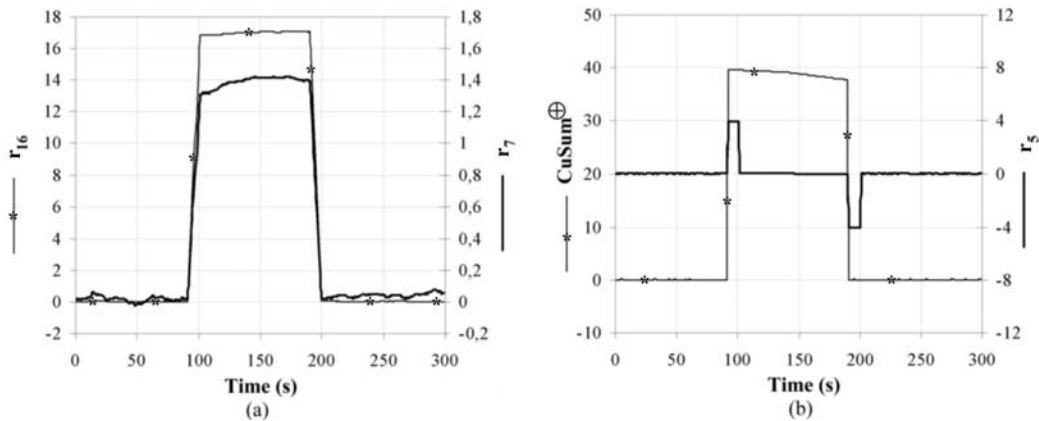


FIG. 4.23 – Détection d'une fuite dans le condenseur

Défaillances capteurs Concernant l'application du générateur de vapeur, toutes les défaillances des capteurs peuvent être détectées et localisées sauf celle du capteur L_1 . Toutefois, il est important de noter que seules les défaillances abruptes des capteurs sont considérées dans cette implémentation. Lorsqu'un capteur est défaillant, les variations d'amplitude des résidus correspondants dépendent de la forme mathématique de la variable capteur présente dans les équations.

FIG. 4.24 – Défaillance du capteur P_7 : (a) Détection normale avec r_7 et r_{16} (b) Détection avec $CuSum_2$ pour r_5 .

Considérons le cas de la défaillance du capteur P_7 . La signature de défaillance correspondant à ce composant est donnée au tableau 4.4. Au moment de la défaillance, la pompe n'était pas en marche ($mO_2 = 0$) et en plus, seulement quatre résidus (r_4 , r_5 , r_7 et r_{16}) sont sensibles à la défaillance dans ce cas. La détection des dépassements de seuils dans les résidus

r_4 et r_5 a été réalisée en utilisant l'algorithme CuSum₂; tandis que des tests simples ont été utilisés pour détecter les déviations de r_7 et r_{16} . Les réponses de ces résidus à la défaillance du capteur P_7 , durant la fenêtre temporelle allant de 90 à 190 s, sont données à la figure 4.24. Cette défaillance est donc localisable puisque le vecteur de cohérence généré possède une seule correspondance dans la matrice des signatures.

De même, les défaillances des autres capteurs peuvent être détectées par de simples tests sur les seuils des résidus et de la somme cumulée CuSum₂. A l'exception du capteur L_1 , toutes les défaillances des autres capteurs peuvent être localisées.

Défaillances des actionneurs Ces défaillances correspondent à des anomalies de fonctionnement dans les actionneurs telles que les pompes, les vannes ou encore la chauffe. Dans cette section on considère une défaillance survenue sur la résistance thermique (la chauffe) de la chaudière. Cette défaillance peut être causée par un problème de fusible, un court-circuit ou encore une anomalie dans la résistance.

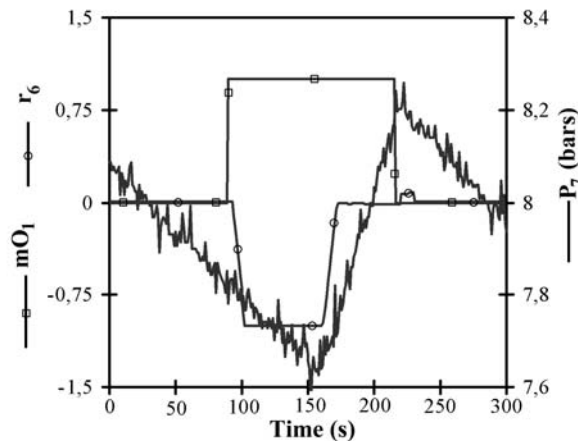


FIG. 4.25 – Détection de la défaillance de la chauffe.

Les résultats de cette défaillance, correspondant à une déviation de la valeur du résidu r_6 , sont montrés à la figure 4.25. On peut constater que durant toute la durée de la défaillance (60 à 160 s) et pour une commande de la chauffe mO_1 non nulle, la pression de la vapeur P_7 continue de descendre à moins de 7.8 bars ($P_{7_ref} = 8 \pm 0.2$ bars). Ce constat est dû au fait que la chauffe ne fonctionnait pas correctement durant cette période et donc ne pouvait pas fournir de la chaleur à la chaudière. A remarquer que de 60 à 90 s, lorsque la commande mO_1 était nulle, la défaillance n'a pas été détectée.

Défaillances des régulateurs Une défaillance dans un régulateur est expliquée par une anomalie entre sa sortie actuelle et la sortie prédite donnée comme une fonction des valeurs des consignes et des mesures impliquées dans la régulation. Les défaillances dans les régulateurs physiques peuvent être simulées en changeant les paramètres du régulateur (e.g. gains, distorsion, etc.). Dans le générateur de vapeur les régulateurs sont des lois de commande programmées et implémentées dans la plateforme de supervision. Les défaillances des régulateurs ont été obtenues par la modification logicielle des paramètres de régulation pendant le fonctionnement de l'installation.

Comme exemple, considérons la défaillance du régulateur de niveau dans le condenseur. Le niveau de condensât est maintenu autour d'une valeur constante, selon la valeur de la consigne ($L_{18_ref} \pm 0.5 l$), par un régulateur tout ou rien (on-off). Lorsque la limite supérieure de la consigne est atteinte les trois vannes tout ou rien du système de décharge s'ouvrent et le condensât est évacué vers le réservoir de stockage jusqu'à ce le niveau atteigne la limite inférieure de la valeur de la consigne.

Pour réaliser la défaillance sur ce régulateur, on a changé sa loi de commande de telle sorte qu'il ne réagisse pas même si le niveau du condensât a excédé la valeur de la consigne ($L_{18_ref} = 6 \pm 0.5 l$). Parallèlement, la valeur de sortie prédite du régulateur est toujours calculée selon l'algorithme initial. Par conséquent, il y a anomalie entre la sortie mesurée du régulateur et sa valeur initialement prédite.

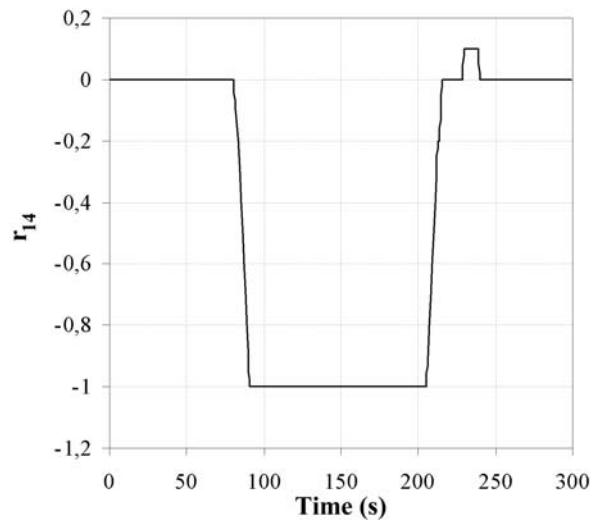


FIG. 4.26 – Détection d'une défaillance dans le régulateur de niveau du condenseur

Comme montré à la figure 4.26, la défaillance du régulateur est détectée lorsque la déviation du résidu r_{14} excède son seuil préalablement fixé ($\varepsilon_{14} = 0.5$).

4.3 Conclusion

Dans ce chapitre une application industrielle, le générateur de vapeur, a été considérée pour tester les fonctionnalités de la boîte à outils ModelBuilder. Les modèles bond graphs des composants des processus ont été créés et implémentés sous forme d'objets dans la boîte à outils. Le P&ID de l'installation a été construit en utilisant ModelBuilder et les RRAs ainsi que les matrices de signatures des défaillances correspondantes ont été générées. L'analyse de la matrice de signature des défaillances nous a permis de déterminer la liste des composants non surveillables et de proposer ensuite un placement de capteurs pour améliorer leur surveillance. Enfin, pour mettre en application et valider les outils développés, une implémentation en temps réel des résidus du générateur de vapeur a été réalisée et des résultats expérimentaux ont été obtenus.

Conclusions et perspectives

1 Conclusions générales

Dans ce mémoire nous avons présenté une nouvelle approche intégrée pour la supervision des processus en génie des procédés. Nous avons utilisé un seul outil pour la modélisation, la générations des indicateurs de défaillances, l'analyse de la surveillabilité et le placement de capteurs en vue de satisfaire le cahier des charges pour la surveillance. Ce travail de recherche a été finalisé par l'automatisation de ces procédures à l'aide d'une boîte à outils développée dans le logiciel SYMBOLS 2000.

Dans le premier chapitre a été présentée une classification de la plupart des composants utilisés en génie des procédés. Cette classification est basée sur l'analyse fonctionnelle et comportemental des composants. Ensuite, des règles de connexion entre les composants ont été définies pour construire le modèle architectural du processus et pour vérifier sa validité du point de vue fonctionnel et causal. Enfin, le caractère graphique et générique du modèle bond graph a permis d'automatiser les procédures de génération de modèles dynamiques à partir d'une simple connexion des modèles validés de chaque composant.

Ensuite dans le deuxième chapitre, nous avons développé trois principales méthodes pour la générations des RRAs, des résidus et des matrices de signature des défaillances. La première méthode a été utilisée pour la générations de RRAs sous forme de relations analytiques symboliques. Nous avons alors présenté un algorithme qui permet de les générer à partir d'un modèle bond graph en causalité dérivée préférentielle. Cependant, cette méthode s'est avérée non applicable en présence de boucles (algébriques ou différentielles) et/ou de relations non inversibles. Une deuxième méthode a été alors développée pour contourner ce problème et générer cette fois des résidus sous forme numérique. Pour cela, des modifications ont été apportées au modèle bond graph initial en causalité dérivée de façon à ce que nous puissions obtenir un modèle dit Bond Graph de Diagnostic (BGDg) dans lequel les entrées sont les mesures des capteurs et les valeurs des sources et les sorties sont des valeurs numériques données par les détecteurs de résidus numériques. Dans la troisième partie la méthode du BGDg a été étendue pour la génération de résidus dépourvus de toutes formes intégrale ou singulière.

Les procédures et les algorithmes développés dans le premier et le second chapitre ont été ensuite informatisés sous forme d'une boîte à outils appelée ModelBuilder. L'interface graphique de la boîte à outils permet d'une façon conviviale:

- de construire les modèles dynamiques à partir d'une bibliothèque de modèles et par une simple connexion des modèles architecturaux validés,
- de générer les RRAs et la matrice de signatures des défaillances,
- d'analyser la surveillabilité du processus et de proposer un placement de capteurs pour satisfaire le cahier des charges pour la surveillance,
- et enfin de produire sous format XML: les équations du modèle dynamique, le modèle architectural, la liste des résidus, la matrice des signatures et les résultats de l'analyse de la surveillabilité.

Enfin, dans le dernier chapitre, nous avons appliqué ModelBuilder sur une installation industrielle (le générateur de vapeur du LAGIS) faisant partie de la plateforme de supervision du projet CHEM. Cette application nous a permis de tester la boîte à outils et les résultats obtenus ont été conformes aux cahiers des charges fixés permettant ainsi de la valider.

2 Perspectives

Durant la durée du travail de recherche nous avons relevé quelques points que nous aimerions développer plus tard:

- Dans ce travail, nous avons utilisé pour la modélisation le bond graph classique (une barre de causalité sur chaque lien) qui ne permet pas d'affecter arbitrairement la causalité. Dans ce cas, il nous arrive de ne pas représenter tous les couplages (entre variables) ce qui peut parfois réduire le nombre de RRAs générées. Nous projetons alors d'étendre notre travail à l'utilisation de la notion de la bicausalité (qui a été utilisée pour l'inversion des systèmes, estimations des paramètres et le dimensionnement des actionneurs) pour exploiter tous les couplages possibles à la génération de RRAs.
- Les RRAs générées dans le présent travail sont déterministes ne tenant pas compte des incertitudes des paramètres. L'outil bond graph permet de rajouter graphiquement ces incertitudes et de générer ainsi des RRAs robustes par rapport aux incertitudes paramétriques. Ainsi, dans la procédure de décision nous pouvons déterminer des seuils de déclenchement adaptatifs.
- Les modèles que nous avons considérés sont à paramètres localisés et par conséquent les défaillances détectées dépendent seulement du temps. Notre souhait est alors d'étendre les procédures développées dans ce mémoire aux processus représentés par des modèles à paramètres distribués (en utilisant toujours le bond graph comme outil). Ainsi, nous

seront en mesure de détecter des défaillances dépendant et du temps et de l'espace. La détection de fuites dans les gazoducs ou dans les réacteurs (qui sont décrits surtout par des EDPs) reste un domaine d'application privilégié pour lequel nous pourrions étendre notre approche.

- Les tests, sur le FCC, de la méthodologie développée dans ce travail ont permis de mettre en évidence des difficultés d'application dans les systèmes où il y a transformation de la matière et pour lesquels les modèles sont souvent donnés sous forme empirique. Cette complexité présente dans les processus chimiques introduit une voie de recherche intéressante car elle s'inscrit dans le cadre de la protection de l'environnement.

Bibliographie

- [1] Advanced decision support system for Chemical/Petrochemical manufacturing processes. project N° G1RD-CT-2001-00466, CHEM Project 2000; <http://www.chem-dss.org>.
- [2] H.M. Paynter. *Analysis and design of Engineering Systems*. M.I.T. Press, 1961.
- [3] J.U. Thoma. *Introduction to Bond Graphs and their Applications*. Pergamon Press, Oxford; New York, 1975.
- [4] D.C. Karnopp, D. Margolis, and R. Rosenberg. *Systems Dynamics: A Unified Approach*. John Wiley, New York, second edition, 1990.
- [5] G. Dauphin-Tanguy. *Les Bond Graphs*. Hermès Science, 2000.
- [6] A Mukherjee and R. Karmakar. *Modelling and Simulation of Engineering Systems through Bond Graphs*. Alpha Sciences International, Pangbourne, UK, 2000.
- [7] D. C. Karnopp. Bond graphs in control: Physical state variables and observers. *Journal of the Franklin Institute*, 308(3):221–234, 1979.
- [8] G. Dauphin-Tanguy, A. Rahmani, and C. Sueur. Bond graph aided design of controlled systems. *Simulation Practice and Theory*, 7(5-6):493–513, 1999.
- [9] P.J. Gawthrop. Physical interpretation of inverse dynamics using bicausal bond graphs. *Journal of the Franklin Institute*, 337(6):743–769, 2000.
- [10] R. Fotsu Ngwompo, S. Scavarda, and D. Thomasset. Inversion of linear time-invariant siso systems modelled by bond graph. *Journal of the Franklin Institute*, 333(2):157–174, 1996.
- [11] M. Tagina. *Application de la Modélisation Bond Graph À la Surveillance Des Systèmes Complexes*. Thèse de doctorat, Université des Sciences et Technologies de Lille, France, 1995.
- [12] D. Linkens and H. Wang. Qualitative bond graph reasoning in control engineering: Fault diagnosis. In *International Conference on Bond Graph Modeling and Simulation (ICBGM'95)*, pages 189–194. Simulation Series, Vol.27, No.1, ISBN 1-56555-037-4, 1995.
- [13] T. Kohda, K. Inoue, and H. Asama. Computer aided failure analysis using system bond graphs. In *International Conference on Bond Graph Modeling and Simulation (ICBGM'01)*, pages 71–76. Simulation Series, Vol.33, No.1, ISBN 1-56555-103-6, 2001.

-
- [14] P.J. Mosterman and Biswas G. Diagnosis of continuous valued systems in transient operating regions. *IEEE Transactions on Systems, Man, and Cybernetics*, 29(6):554–565, 1999.
- [15] M. Tagina, J.Ph. Cassar, G. Dauphin-Tangy, and M. Staroswiecki. Monitoring of systems modelled by bond graph. In *International Conference on Bond Graph Modeling and Simulation (ICBGM'95)*, pages 275–280. Simulation Series, Vol.27, No.1, ISBN 1-56555-037-4, 1995.
- [16] B. Ould Bouamama, M. Staroswiecki, B. Riera, and E. Cherifi. Multi-modelling of an industrial steam generator. *Control Engineering Practice*, 8(11):1249 – 1260, 2000.
- [17] J.U Thoma and B. Ould Bouamama. *Modelling and Simulation in Thermal and Chemical Engineering. Bond Graph Approach*. Springer Verlag, Telos, 2000.
- [18] B. Ould Bouamama. Bond graph approach as analysis tool in thermofluid model library conception. *Journal of the Franklin Institute*, 340(1):1–23, 2003.
- [19] B. Ould Bouamama, A.K. Samantaray, K. Medjaher, M. Staroswiecki, and G. Dauphin-Tanguy. Model builder using functional and bond graph tools for fdi design. *Control Engineering Practice*, 13(7):875–891.
- [20] B. Ould Bouamama, K. Medjaher, A.K. Samantaray, and M. Staroswiecki. Supervision of an industrial steam generator. part i: Bond graph modelling. *Control Engineering Practice*, 2005. In press, available online at www.sciencedirect.com.
- [21] K. Medjaher, A.K. Samantaray, B. Ould Bouamama, and M. Staroswiecki. Supervision of an industrial steam generator. part ii: Online implementation. *Control Engineering Practice*, 2005. In press, available online at www.sciencedirect.com.
- [22] A.K. Samantaray, K. Medjaher, B. Ould Bouamama, M. Staroswiecki, and G. Dauphin-Tanguy. Component based modelling of thermo-fluid systems for sensor placement and fault detection. *SIMULATION: Transactions of The Society for Modeling and Simulation International, special issue of Simulation journal on "Component Based Modelling and Specifications"*, 80(7-8):381–398, 2004.
- [23] C. Feliot, M. Staroswiecki, and J.P. Cassar. Towards a formal language of physical systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 4:2768–2773, 1996.
- [24] T. Murakami and N. Nakajima. Computer-aided design-diagnosis using feature description. In J.S. Gero, editor, *Artificial Intelligence in Engineering: Diagnosis and Learning*, pages 199–226. Elsevier, 1988.
- [25] D.C. Karnopp. State variables and pseudo-bond graphs for compressible thermo-fluid systems. *Journal of Dynamic Systems, Measurement and Control*, 101(3):201–204, 1979.
- [26] D.C. Karnopp and S. Azerbaijani. Pseudo bond graphs for generalised compartmental models in engineering and physiology. *Journal of the Franklin Institute*, 312(2):95–108, 1981.

-
- [27] L.S. Bonderson. Vector bond graphs applied to one-dimensional distributed systems. *Journal of Dynamic Systems, Measurement and Control*, pages 75–82, 1975.
- [28] F.T. Brown. Convection bonds and bond graphs. *Journal of the Franklin Institute*, 325(5/6):871–886, 1991.
- [29] C.A. Meyer, R.B. McClintock, G.J. Silvestri, and R.C. Spencer. ASME steam tables: Thermodynamic and transport properties of steam, 6th edition, ASME 1993.
- [30] L. Tylee. Pseudo bond graph representation of PWR pressurised dynamic. *Trans. ASME J. Daic Sys. Measure Control*, 105:255–261, 1983.
- [31] J.U. Thoma. *Simulation by Bondgraphs. Introduction to a Graphical Method*. Springer-verlag, Berlin Heidelberg, 1990.
- [32] P. O. Moksnes. *Modeling Two-Phase Thermo-Fluid Systems Using Bond Graph*. Dr. ing thesis, University of Sciences and Technology, Department of Marine Engineering, Norway, 1997.
- [33] A. W. Ordys, A. W. Pike, M. A. Johnson, R. M. Katebi, and M. J. Grimble. *Modelling and Simulation of Power Generation Plants*. Springer-Verlag, 1994.
- [34] M. Kothare, B. Mettler, M. Morari, P. Bendotti, and C.M. Falinower. Level control in the steam generator of a nuclear power plant. *IEEE Transactions on Control Systems Technology*, 8(1):55–69, 2000.
- [35] K.J. Åström and R.D. Bell. Drum-boiler dynamics. *Automatica*, 36:363–378, 2000.
- [36] A. N. Nahavandi and R. F. Von Rollen. A space dependent dynamic analysis of a bwr. *Nuclear Science Engineering*, 20:392–413, 1964.
- [37] S. Péneau, J. P. Humeau, and Y. Jarny. Front motion and convective heat flux determination in a phase change process. *Inverse Problems in Engineering*, 4(1):53–91, 1996.
- [38] C. Feliot. *Modélisation de Systèmes Complexes: Intégration et Formalisation de Modèles*. Ph.d. thesis, Université des Sciences et Technologies de Lille, 1997.
- [39] A. Mukherjee and A.K. Samantaray. System modelling through bond graph objects on SYMBOLS 2000. In *International Conference on Bond Graph Modeling and Simulation (ICBGM'01)*, volume 33, pages 164–170. Simulation Series, Vol.33, No.1, ISBN 1-56555-103-6, 2001.
- [40] R. Cacho, J. Felez, and C. Vera. Deriving simulation models from bond graphs with algebraic loops: The extension to multibond graph systems. *Journal of the Franklin Institute*, 337(5):579–600, 2000.
- [41] C. Sueur and G. Dauphin-Tanguy. Bond graph approach for structural analysis of MIMO linear systems. *Journal of the Franklin Institute*, 328(1):55–70, 1991.
- [42] B. Ould Bouamama, K. Medjaher, M. Bayart, A.K. Samantaray, and B. Conrard. Fault

- detection and isolation of smart actuators using bond graphs and external models. *Control Engineering Practice*, 13(2):159–175, 2005.
- [43] K. Medjaher, A.K. Samantaray, and B. Ould Bouamama. Diagnostic bond graphs for direct residual evaluation. In *International Conference on Bond Graph Modeling and Simulation (ICBGM'05)*, pages 307–312. Simulation Series, Vol.37, No.1, ISBN: 1-56555-287-3, 2005.
- [44] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault Tolerant Control*. Springer-Verlag, 2003.
- [45] M. Basseville and I.V. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Prentice Hall, ISBN 0-13-126780-9, 1993.
- [46] G.N Roberts and C.A.J. Tubb, editors. *Computer-Aided Design of Mechatronic Systems Using 20-SIM 3.0*, Newport (S. Wales), UK, 199. 2nd Workshop on European Scientific and Industrial Collaboration (WESIC'99).
- [47] J. Van Amerongen and P. Breedveld. Modelling of physical systems for the design and control of mechatronic systems. *Annual Reviews in Control*, 27(1):87–117, 2003.
- [48] J. Van Dijk and P.C. Breedveld. Simulation of system models containing zero-order causal paths-i. classification of zero-order paths. *Journal of the Franklin Institute*, 328(5), 1991.
- [49] J. Van Dijk and P.C. Breedveld. Simulation of system models containing zero-order causal paths-ii. numerical implications of class 1 zero-order paths. *Journal of the Franklin Institute*, 328(5), 1991.
- [50] W. Borutzky and F.E. Cellier. Tearing algebraic loops in bond graphs. *Trans. Soc. Computer Simulation Internat.*, 13(2):102–115, 1996.
- [51] W. Borutzky. Supporting the generation of a state space model by adding tearing information to the bond graph. *Simulation Practice and Theory*, 7(5-6):419–438, 1999.
- [52] B. Zupancic. Extension software for real-time control system design and implementation with matlab-simulink. *Simulation Practice and Theory*, 6(8):703–719, 1998.
- [53] M.J. Er, C.B. Low, K.H. Nah, M.H. Lim, and S.Y. Ng. Real-time implementation of a dynamic fuzzy neural networks controller for a scara. *Microprocessors and Microsystems*, 26(9-10):449–461, 2002.
- [54] P. Taillibert. Residual evaluation without derivative computation. In A.M. Edelmayer and C. Banyasz, editors, *SAFEPROCESS 2000*. IFAC, Pergamon Press, ISBN:0-08043-250-6, 2000.
- [55] W. Nusselt. Die oberflächenkondensation des wasserdampfes. *VDI Zeitschrift*, 60(27):541–546 and 569–575, 1916.
- [56] A.L. Chapman. *Heat Transfer*. Macmillan Publishing Company, 4th edition, 1984.

- [57] W.M. Rohsenow, J.P. Hartnett, and E.N. Ganic. *Handbook of Heat Transfer Fundamentals*. McGraw-Hill Book Company, 2nd edition, 1985.