Diagnosticabilité et diagnostic de systèmes technologiques pilotés : développement d'une chaîne de conception outillée d'un système de diagnostic appliquée aux systèmes technologiques pilotés

Michel Batteux

UNIVERSITÉ PARIS-SUD 11

ÉCOLE DOCTORALE INFORMATIQUE PARIS-SUD
Laboratoire de Recherche en Informatique

Discipline : **Informatique**

THÈSE DE DOCTORAT

soutenue le 13 décembre 2011

par

Michel BATTEUX

DIAGNOSTICABILITÉ ET DIAGNOSTIC DE SYSTÈMES TECHNOLOGIQUES PILOTÉS

Développement d'une chaîne de conception outillée d'un système de diagnostic appliquée aux systèmes technologiques pilotés

Directeur de thèse :	Philippe DAGUE	Professeur, Université Paris-Sud
Co-encadrant de thèse :	Nicolas RAPIN	Ingénieur-Chercheur, CEA LIST, Saclay
Co-encadrant de thèse :	Philippe FIANI	Responsable bureau d'étude, Sherpa Engineering

Composition du jury :

Rapporteurs :	Louise TRAVÉ-MASSUYÈS	Directeur de recherche, LAAS-CNRS, Toulouse
	Luca CONSOLE	Full Professor, Universita' di Torino
Examinateurs :	Marie-Odile CORDIER	Directeur de Recherche, IRISA, Rennes
	BurkhartWOLFF	Professeur, Université Paris-Sud
	Philippe DAGUE	Professeur, Université Paris-Sud
	Nicolas RAPIN	Ingénieur-Chercheur, CEA LIST, Saclay
	Philippe FIANI	Responsable bureau d'étude, Sherpa Engineering
Membre invité :	Jean BRUNET	Cogérant, Sherpa Engineering

Remerciements

Ce manuscrit de thèse marque l'aboutissement de trois années d'un travail de recherche que j'ai réalisé, dans le cadre d'une convention CIFRE, au sein de trois partenaires : l'entreprise Sherpa Engineering, le Laboratoire de Recherche en Informatique (LRI) de l'Université Paris-Sud et du CNRS ainsi que le Laboratoire d'Ingénierie des Systèmes Embarqués (LISE) du CEA LIST.

Cette thèse n'aurait pu se dérouler et s'achever sans l'aide précieuse de trois personnes qui ont toujours su me conseiller et m'aider. Monsieur Philippe FIANI, responsable du bureau d'étude de Sherpa Engineering, qui m'a toujours apporté les explications nécessaires, et ce de manière simple, des concepts fondamentaux du monde de l'automatique. Monsieur Nicolas RAPIN, ingénieur-chercheur du LISE, avec qui j'ai commencé à travailler depuis mon stage de fin de master et qui m'a sans cesse ramené vers la compréhension lors de doutes. Monsieur Philippe DAGUE, professeur au LRI, qui a dirigé cette thèse et qui, bien que fortement accaparé par son rôle de directeur du LRI, s'est toujours rendu disponible pour toutes mes questions. Je vous remercie infiniment tous les trois pour tout ce que vous m'avez apporté, mais aussi pour l'humilité dont vous avez toujours fait preuve en gardant toujours pour objectif la réussite du projet commun et non la mise en avant d'objectifs propres.

Je remercie par ailleurs Messieurs Jean BRUNET et Atilla YAZMAN, cogérants de Sherpa Engineering, pour avoir pris en charge cette thèse et avoir été attentifs à ce qu'elle se réalise dans de bonnes conditions.

Je tiens également à adresser mes plus vifs remerciements aux membres du jury pour l'honneur qu'ils m'ont fait d'examiner ces travaux. Je remercie d'abord Madame Louise TRAVÉ MASSUYÈS, Directeur de Recherche au LAAS-CNRS de Toulouse et dont j'ai eu l'honneur de suivre un exposé lors de l'école d'été sur le diagnostic à Girona en 2010, et Monsieur Luca CONSOLE, professeur à l'université de Turin, d'avoir accepté d'étudier mes travaux et d'en être les rapporteurs ainsi que pour l'intérêt et l'attention qu'ils ont accordés à cette étude. J'ai pu, à la lueur des commentaires pertinents figurant dans leurs rapports, envisager mon travail sous des angles nouveaux et entrevoir d'intéressantes perspectives de recherche. Je remercie aussi Madame Marie-Odile CORDIER, Responsable Scientifique au sein de l'IRISA de Rennes, et Monsieur Burkhart WOLFF, professeur au LRI (dont j'ai eu la chance de suivre un cours sur le test de systèmes informatiques en 2009), de faire parti de ce jury.

Mes remerciements vont aussi à tous ceux que j'ai croisés, de manière plus ou moins importante, durant ces trois années. Tous mes collègues de Sherpa Engineering, autant pour des discussions ciblées techniques industrielles que pour les « croissantages masqués ». Tous mes collègues de l'équipe LISE du CEA connus durant mon stage de fin d'étude puis durant le CDD ayant suivi mais que je n'ai que très peu vus durant ces trois années. Tous mes collègues du LRI qui ne m'ont que très peu vu mais dont j'ai pu obtenir conseils de certains.

Pour finir, je ne peux oublier de remercier tout mon entourage personnel m'ayant accompagné durant ces trois ans et même bien avant. En premier lieu et sans mesure, ma famille sans qui je n'aurais pu arriver jusque là. Mes amis pour tout ce que cela implique. Enfin celle qui a réussie à me supporter durant ce temps et qui me donne toujours sourire.

Une dernière mention à ERAGON, mon Latitude D830, qui a toujours répondu présent jours et nuits sans jamais se plaindre même lors de demandes chargés de travail.

Table des matières

Liste des figures

Liste des tableaux

Introduction générale

Contexte

De l'automatisation des systèmes aux défauts potentiels

Depuis plusieurs années déjà, l'automatisation prend une part de plus en plus importante dans la réalisation de fonctions de systèmes industriels et grands publics. Cela s'explique d'une part par un accroissement de la complexification des systèmes en eux-mêmes (i.e. : augmentation du nombre de composants et de leurs interactions) afin de réaliser des fonctions de plus en plus complexes, et d'autre part par les enjeux économico-stratégiques actuels que sont la réduction des coûts de maintenance ainsi que l'amélioration des performances de fonctionnement, de fiabilité et de sécurité et sûreté des systèmes.

De par leur nature automatique, de tels systèmes sont des combinaisons complexes et structurées de multiples composants mécaniques, électroniques et informatiques en interaction permanente et combinant de multiples phénomènes physiques. La réalisation des fonctions du système est ainsi répartie sur les différents composants, dont une des conséquences est un accroissement des défauts potentiels (aussi nommés défaillances) pouvant apparaître dans le système. Dysfonctionnements et pannes du système, ne permettant plus son maintien en conditions opérationnelles, peuvent alors résulter de ces défauts. Ceci ayant alors un impact néfaste, voire catastrophique, sur le système en lui-même ainsi que son environnement : pertes de performances du système, casse de composants internes au système, dégradation de l'environnement, pertes de vies humaines, et dans tous les cas un impact financier et sur l'image de marque. Il est alors important de prendre en compte cette problématique d'occurrences de défauts potentiels dans le système afin d'éviter toutes conséquences dommageables, autant pour le système que son environnement.

Une solution : le diagnostic en ligne des défauts

Une solution adéquate, permettant de répondre à cette problématique de défauts, est le développement d'outils de surveillance en ligne du système basés sur les méthodologies de diagnostic de défauts. L'objectif du diagnostic de défauts est de détecter et de localiser, le plus précisément possible, les défauts pouvant apparaître dans un système ; ceci dans le but de prendre les décisions adéquates quant aux poursuites de fonctionnement appropriées suivant la sévérité du défaut diagnostiqué : poursuite en fonctionnement normal pour des défauts n'impactant que les performances du système, ou poursuite en mode dégradé, voire mise à l'arrêt du système, pour des défauts ayant de plus graves conséquences. Pour le diagnostic en ligne, il s'agit concrètement d'embarquer directement dans le système un dispositif qui vise à déterminer, par une fonction de surveillance en ligne, si le fonctionnement du système (aussi nommé comportement du système) est conforme à celui espéré par son concepteur. Dans le cas contraire, ce dispositif doit être capable de déterminer le plus précisément possible quelles sont les parties en défaut du système et de quels types de dysfonctionnements elles souffrent.

Autant les exigences de sécurité que la réduction des coûts d'exploitation ainsi que la maîtrise de la disponibilité des systèmes, mises en avant lors de l'explication de la croissance de l'automatisation,

donnent donc au diagnostic un rôle prépondérant. En effet, en accomplissant une surveillance du système, le diagnostic permet une détection et une identification des défauts de manière précise et rapide. Il est ainsi possible de n'intervenir qu'en présence de composants défectueux, de minimiser le temps de réparation, et de fournir une réponse (i.e. : un diagnostic) fiable et facilement interprétable malgré la complexité des systèmes.

Ambiguïté du diagnostic : la diagnosticabilité

Pour être exploitable, un tel dispositif de diagnostic, aussi nommé diagnostiqueur, doit être capable d'une part de déterminer qu'un comportement est anormal, et d'autre part de n'attribuer qu'un seul défaut préalablement répertorié à un tel comportement anormal. Le diagnostic devient donc ambigu lorsque le diagnostiqueur n'est pas capable de discerner un fonctionnement normal du système d'un fonctionnement de défaut, ou qu'il n'est pas capable de discerner deux fonctionnements de défauts distincts du système.

Or comme tout dispositif de surveillance d'un système, un diagnostiqueur n'a accès qu'à une certaine partie dite « observable » du fonctionnement du système ; ce qui signifie qu'un comportement observable n'est donc qu'une abstraction d'un comportement plus « complet » du système. Ainsi et selon le « niveau » réel d'observation du système, il pourrait de ce fait exister deux comportements distincts du système, l'un contenant un défaut et l'autre en contenant un autre ou n'en contenant pas, mais qui s'abstrairaient néanmoins tous les deux par le même comportement observable.

Par ailleurs, le fonctionnement du diagnostiqueur repose sur la vérification de la conformité du comportement observable du système vis-à-vis de la connaissance disponible sur son fonctionnement normal ou sous la présence d'un défaut ; conformité généralement donnée par des relations liant les données issues autant de l'observation du système que de cette connaissance disponible. Ici encore, il pourrait exister deux telles relations distinctes, dont l'une spécifiant le fonctionnement du système sous la présence d'un défaut et l'autre spécifiant son fonctionnement normal ou sous la présence d'un autre défaut distinct, mais qui soient néanmoins toutes les deux vérifiées par les mêmes comportements observables.

L'ambiguïté du diagnostic peut donc être inhérente tant à cette vérification qu'à la complétude de l'observation du système lui-même. Il est dans ce cas important de pouvoir déterminer à l'avance si non seulement le « niveau » réel d'observation du système, mais aussi les relations utilisées par la vérification permettent toujours un diagnostic non ambigu.

La diagnosticabilité répond à ce problème en permettant de déterminer si, quelque soit le comportement réellement observé du système, il est toujours possible d'en donner un diagnostic non-ambigu. Il s'agit d'une étude menée lors de la phase de conception du diagnostiqueur et qui permet de s'assurer que les défauts potentiellement répertoriés du système seront bien détectés et identifiés par ce diagnostiqueur. Elle consiste à faire différentes vérifications : d'une part vérifier que le diagnostiqueur sera toujours capable de détecter un défaut lorsqu'il apparaît, ceci en s'assurant qu'un défaut engendre toujours un comportement observable anormal ; d'autre part vérifier qu'il sera toujours capable d'identifier un unique défaut répertorié pour un comportement observable anormal donné, ceci en s'assurant que plusieurs défauts n'engendrent pas les mêmes comportements observables.

Problématique

Le diagnostic automatique est une discipline relativement ancienne dont les premiers travaux sont apparus dès les années 1970, en même temps que l'intégration des premiers calculateurs numériques permettant le contrôle automatique des systèmes. Depuis, et bien que de nombreuses méthodologies furent développées par différentes communautés issues de domaines de recherche variés, peu de théories générales existent et beaucoup d'approches sont plus *ad-hoc* sans démarche systématique ni générique.

Les communautés issues de l'intelligence artificielle et de l'automatique ont, pour leur part, élaboré des méthodologies dites *à base de modèles*. Elles consistent à comparer le comportement réellement observé du système à un comportement prédit, issu d'un modèle de fonctionnement normal ou anormal du système. Ce modèle, représentant une abstraction du fonctionnement normal ou anormal du système, est simulé en temps réel par le diagnostiqueur suivant les mêmes « entrées » que le système réel. Ses « sorties » sont ainsi comparées aux sorties du système réel afin d'y détecter une différence traduisant un fonctionnement anormal. Or suivant le type de modélisation utilisé, reflétant ainsi l'abstraction réalisée du système, la comparaison ne tient pas toujours compte du fonctionnement temporel du système : c'est-à-dire de l'évolution de son fonctionnement dans le temps. Le diagnostic en devient donc beaucoup plus difficile du fait du délai potentiellement différent entre la réponse du système et du modèle aux différentes entrées.

Par ailleurs et bien qu'étant une problématique assez récente, l'étude de la diagnosticabilité d'un système a été bien investie elle aussi. Remarquons qu'il s'agit principalement des communautés du diagnostic à base de modèles et que la majorité des travaux exploitent, pour ce faire, les concepts classiques de diagnostic afin de réaliser cette étude de diagnosticabilité. Néanmoins, le lien entre l'étude de diagnosticabilité et le passage au diagnostiqueur (i.e. : la génération d'un diagnostiqueur issu de l'étude de la diagnosticabilité) n'est souvent pas clairement établi. Il semble en effet que ces études supposent un cas « idéal » de diagnostiqueur ayant la capacité de connaître l'état courant du système surveillé, de calculer l'état de référence ainsi que tous les états de défauts, puis de tous les comparer ; ce qui risque de ne pas être réalisable en pratique car il faudrait alors soit simuler beaucoup de modèles en même temps, soit stocker un nombre important de données. Ainsi ces études permettent de s'assurer que si un défaut n'est pas diagnosticable pour un tel diagnostiqueur « idéal », il ne le sera alors pas pour toute solution de fonctionnement d'un diagnostiqueur « réaliste » ayant accès qu'à une partie moins précise de l'état du système. Or il parait important, de notre point de vue, que cette étude de la diagnosticabilité soit menée suivant une telle solution de fonctionnement d'un diagnostiqueur « réaliste ».

Enfin et malgré l'abondance des travaux traitant du diagnostic, la représentation des défauts reste assez marginale et incomplète. Il y a en particulier difficulté à prendre en compte la diversité des défauts : défaut permanent, intermittent, brusque, progressif, etc.

Nous observons qu'il est ainsi difficile de concevoir et réaliser un diagnostiqueur d'un système. Cette difficulté venant de l'inexistence d'une méthodologie générique définie sur un fond théorique commun et permettant la conception et la réalisation d'un diagnostiqueur. Il serait alors utile d'élaborer une telle méthodologie générique qui doit être cohérente et complète.

Il s'agirait d'une chaîne de conception d'un diagnostiqueur, d'une part établie suivant un cadre théorique commun pour répondre à la cohérence, et intégrant d'autre part les étapes suivantes pour la complétude : les étapes préalables de représentation du système et des défauts potentiels, ainsi que celles permettant ensuite l'étude de la diagnosticabilité puis la génération du diagnostiqueur associé à cette étude.

Contribution de la thèse

Nos travaux de recherche, relatés dans ce mémoire, sont le résultat d'un projet collaboratif réunissant l'entreprise Sherpa Engineering, le Laboratoire de Recherche en Informatique (LRI) unité mixte de recherche (UMR8623) de l'Université Paris-Sud et du CNRS et enfin le Laboratoire d'Ingénierie des Systèmes Embarqués (LISE) du CEA LIST. Ces travaux ont eu pour objectif de définir un cadre théorique commun permettant de réaliser une telle chaîne de conception appliquée aux systèmes technologiques pilotés, tout en exploitant les méthodes et outils habituellement utilisés dans le diagnostic ainsi que dans la conception des systèmes. Ces systèmes technologiques pilotés sont des systèmes mé-

catroniques pour lesquels nous distinguons particulièrement la partie pilotage de la partie opérante. Nous justifierons cette représentation par le fait que, bien que la majorité des travaux de diagnostic considère uniquement la partie opérante, ces systèmes sont généralement bouclés avec leur partie pilotage afin d'accroître leurs performances et de les maintenir en dépit d'entrées inconnues pouvant les affecter.

Le lien entre l'étude de la diagnosticabilité et le passage au diagnostiqueur

Nous avons au préalable établi que le fonctionnement d'un diagnostiqueur est basé sur la vérification de relations de conformité entre le fonctionnement observé du système et la connaissance disponible sur ses fonctionnements normal et anormaux. Ces relations représentent les règles d'analyse du diagnostiqueur et sont données par une famille de propriétés (une propriété par défaut répertorié en incluant aussi le fonctionnement sans défaut) définies suivant un même formalisme et liant les variables observables du système. C'est ce que nous nommerons une *caractérisation de défauts*.

Nous avons ainsi élaboré une méthodologie qui permet de définir de telles propriétés suivant un ensemble de défauts potentiels préalablement répertoriés. Comme elles représentent les règles de fonctionnement du diagnostiqueur, ces propriétés sont ensuite utilisées afin d'étudier la diagnosticabilité des différents défauts, puis d'en générer le diagnostiqueur associé à cette étude de diagnosticabilité. Cela permet donc une cohérence durant l'élaboration du diagnostiqueur.

Utilisation d'un cadre théorique commun

Notre cadre théorique commun est fondé sur l'utilisation de modèles. Cela se justifie par le fait que les méthodes de conception des systèmes sont de plus en plus basées sur l'utilisation d'un ensemble cohérent de modèles génériques représentant le système.

Dans la pratique il s'agira d'utiliser les outils de simulation, tel que MATLAB/Simulink© par exemple qui intègrent des bibliothèques génériques, afin de modéliser et simuler divers systèmes. La combinaison à des outils de génération automatique de code à partir de modèles, tel que dSPACE© par exemple, permet ensuite d'obtenir directement le logiciel du contrôleur issu de ces modèles, en y incluant par conséquent le diagnostiqueur.

Ajout d'un aspect temporel dans le diagnostic à base de modèles

Nous avons par ailleurs choisi de reprendre l'idée des méthodologies de diagnostic à base de modèles qui consiste à comparer le comportement réellement observé du système au comportement prédit issu d'un modèle simulé, de bon ou mauvais fonctionnement. Ce choix nous fut pertinent dans le sens où, comme nous venons de le justifier, les méthodes de conception des systèmes ont de plus en plus recours à des modèles, certains de ces modèles pouvant aussi être utilisés comme modèles de comparaison. Mais souvent comme nous l'avons déjà souligné, ces méthodologies n'intègrent pas l'aspect temporel du fonctionnement du système, notamment pour des représentations par modèles continus.

Cet aspect temporel nous a néanmoins paru important car le fonctionnement d'un système n'est généralement pas du type instantané : par exemple l'application d'une consigne de l'opérateur nécessite un délai de réponse du système, il s'agit de la dynamique de réponse du système. De plus, et comme nous le verrons lors de la présentation des différentes méthodologies de diagnostic au chapitre 1, la fiabilité d'un diagnostic à base de modèles provient, pour une très grande part, de la pertinence du modèle utilisé. Or il existe une contradiction entre un modèle fortement représentatif d'un système complexe, qui nécessitera généralement d'être non linéaire, et son utilisation dans un contrôleur embarqué, qui pour des contraintes de capacité de calcul nécessitera d'être linéarisé autour de certains points de fonctionnement. En se basant uniquement sur des observations instantanées, la comparaison réel-modèle pourrait ainsi ne pas être pertinente selon le point de fonctionnement.

Identification et prise en compte des défauts potentiels

Nous avons aussi introduit une méthodologie permettant d'inventorier les défauts potentiels à prendre en compte et de les modéliser convenablement, c'est-à-dire de les intégrer au plus juste dans le modèle de bon fonctionnement du système. En faisant le lien avec la sûreté de fonctionnement, nous avons établi le point d'entrée du diagnostic : les défauts potentiels devant être traités par un diagnostiqueur sont répertoriés durant l'étude de sûreté de fonctionnement du système. Ensuite et toujours suivant cette même démarche de conception par utilisation de modèles, nous avons déterminé les traits caractéristiques des défauts afin de les intégrer au plus juste dans le modèle de bon fonctionnement du système.

Organisation du mémoire

Ce mémoire est organisé en deux grandes parties. Faisant suite à une introduction classique aux méthodologies de diagnostic, la première partie présentera les systèmes étudiés (i.e. : les systèmes technologiques pilotés) ainsi que la manière d'y considérer les défauts potentiels. La seconde partie présentera quant à elle l'étude de la diagnosticabilité d'un tel système suivant les différents défauts répertoriés puis la manière de générer un diagnostiqueur du système issu de cette étude préalable de la diagnosticabilité.

Le premier chapitre va donc présenter un état de l'art du diagnostic. Nous introduirons en préliminaire l'inclusion du diagnostic comme composante de la maintenance des systèmes, puis les concepts fondamentaux du diagnostic que sont la détection, la localisation ainsi que la reconfiguration, et enfin les différentes exigences liées au diagnostic. Nous ferons ensuite une présentation des différentes méthodologies de diagnostic issues de la littérature. Nous remarquerons alors qu'il est plus judicieux d'adopter une classification basée sur la connaissance du système car les techniques de recherche, utilisées pour diagnostiquer les défauts d'un système, dépendent fortement de ces connaissances disponibles sur le système et les défauts potentiels. Enfin nous introduirons la notion de diagnosticabilité telle qu'elle est traitée dans la littérature. Nous remarquerons que, bien qu'une étude de diagnosticabilité soit généralement issue des méthodologies du diagnostic, le lien entre cette étude et le passage au diagnostic n'est souvent pas clairement établi.

Dans le deuxième chapitre, nous présenterons les systèmes considérés pour lesquels nous souhaitons intégrer un diagnostiqueur : les systèmes technologiques pilotés que nous abrègerons par *systèmes pilotés*. En faisant un bref retour sur le passage des systèmes mécaniques aux systèmes mécatroniques, nous présenterons ces systèmes pilotés comme des systèmes mécatroniques pour lesquels nous faisons une séparation en deux parties de leurs architectures : une partie pilotage et une partie opérante assurant la fonction d'usage du système. Nous montrerons ensuite comment modéliser ces systèmes afin de les utiliser pour l'étude de la diagnosticabilité et le diagnostic.

Le troisième chapitre introduira pour sa part la typologie des défauts. Suite au choix de ne considérer des défauts que sur la partie opérante d'un système piloté, nous présenterons comment identifier, grâce à une étude de sûreté de fonctionnement, les défauts potentiels du système qui doivent être pris en compte par un diagnostiqueur. Nous ferons ensuite un état de l'art des différentes caractéristiques d'un défaut pour conclure qu'un défaut peut être caractérisé par son comportement et son effet sur le système. Enfin nous montrerons comment intégrer dans le modèle de bon fonctionnement, grâce à ces deux caractéristiques, les défauts potentiels identifiés ; nous présenterons une librairie MATLAB/Simulink© développée spécifiquement pour réaliser ce type d'intégration dans des modèles de simulation.

Dans le quatrième chapitre, le plus important de ce mémoire, nous présenterons la méthodologie d'étude de la diagnosticabilité des défauts du système. Suite à un retour sur le fonctionnement d'un diagnostiqueur, nous mettrons en avant le fait que cette étude est basée sur la satisfaction, par les comportements observables du système, des propriétés d'une caractérisation de défauts qui représentent

les règles de fonctionnement du diagnostiqueur. Nous définirons alors les comportements observables du système sous la présence ou non de défauts, puis nous introduirons cette notion de diagnosticabilité qui est donnée en fonction d'une caractérisation de défauts. Nous présenterons ensuite deux caractérisations de défauts : une parfaite qui nous permettra de nous assurer de manière intrinsèque de la diagnosticabilité des défauts, mais qui sera néanmoins impraticable, puis une caractérisation basée sur un formalisme de logique temporelle qui sera adéquate suivant les contraintes de l'embarqué.

Le cinquième chapitre, quant à lui, montrera comment générer un diagnostiqueur du système suivant l'étude de la diagnosticabilité, et donc suivant une caractérisation de défauts. Nous présenterons de manière formelle comment fonctionne un diagnostiqueur, notamment son comportement pour des détections non répertoriées, puis comment l'implanter dans la partie pilotage du système piloté. Nous ferons ensuite une étude de la complexité de ce fonctionnement en indiquant sa nécessité avant toute implémentation, ceci afin de s'assurer que les capacités de stockage et de puissance de calcul soient suffisantes en fonction de la caractérisation de défauts utilisée lors de l'étude de la diagnosticabilité.

Enfin, le dernier chapitre conclura ce document. Nous mettrons en avant les différents travaux réalisés qui nous auront permis de définir cette chaîne de conception outillée d'un diagnostiqueur suivant un cadre théorique commun. Nous discuterons ensuite de quelques points pour lesquels d'intéressantes perspectives peuvent être mises en œuvre pour de futurs travaux complémentaires.

Un exemple applicatif, présenté au chapitre 2 d'introduction aux systèmes technologiques pilotés, sera repris durant toute la suite du document afin de rendre compte des différentes notions théoriques introduites. Il s'agira d'un générateur électrique à pile à combustible, qui est généralement nommé *système pile à combustible* ou simplement *pile à combustible*. Il s'agit d'un système développé dans le cadre du projet FISYPAC dont le but était d'intégrer un tel système dans un véhicule électrique et de l'hybridiser avec une batterie de puissance.

Chapitre 1

Les méthodologies de diagnostic

L'objectif du diagnostic est l'identification des défauts d'un système à partir de l'étude de leurs effets occasionnés. Le diagnostic s'inscrit dans les solutions de surveillance des systèmes dans le but de prendre les décisions adéquates quant à leurs poursuites de fonctionnement suite à des défauts apparus et diagnostiqués ; ces décisions adéquates pouvant aller de la poursuite en fonctionnement normal ou de la mise en mode dégradé, si seules les performances du système sont impactées, à la mise à l'arrêt du système si sa sécurité ou celle de son environnement sont impactées.

Ce chapitre a pour but de présenter les différentes méthodologies de diagnostic qu'il est possible de rencontrer dans la littérature. Nous introduirons, au préalable, les contraintes et exigences liées à la mise en place d'un diagnostiqueur d'un système. Ensuite, et en remarquant que les deux principales composantes d'un système de diagnostic sont d'une part la connaissance *a priori* du système et d'autre part la technique de recherche utilisée, qui dépend fortement des connaissances disponibles du système, nous présenterons les différentes méthodologies de diagnostic pour lesquelles nous aurons adopté une classification basée sur la connaissance du système. Nous les classerons en deux catégories : les approches basées sur les modèles et les approches basées sur les données. Enfin, nous présenterons la problématique de diagnosticabilité d'un système qui consiste à s'assurer que le diagnostiqueur sera toujours capable de diagnostiquer sans ambiguïté un défaut préalablement répertorié.

1.1 Préliminaires

1.1.1 La maintenance des systèmes industriels

La maintenance des systèmes industriels désigne le domaine de recherche permettant de maintenir ou de rétablir un système dans son état de fonctionnement normal. Il s'agit de l'un des problèmes stratégiques qui se pose à l'industriel, depuis la conception d'un système jusqu'à son exploitation en passant bien entendu par sa mise en œuvre. Nous allons donc brièvement présenter cette thématique générale et montrer qu'elle englobe différentes thématiques telles que la sûreté de fonctionnement et le diagnostic.

En se basant sur la norme AFNOR NFX 13-306 ([AFN01]), la maintenance est définie comme l'ensemble des actions permettant de maintenir ou de rétablir un bien dans un état spécifié ou en mesure d'assurer un service déterminé. Elle a donc pour objectif de garantir le « bon fonctionnement » d'un système, aussi bien pour des questions de sécurité et de sûreté de fonctionnement, que pour des questions de rentabilité.

Une maintenance performante, en terme de fiabilité et de rentabilité, nécessite la mise en œuvre d'outils de contrôle permettant de surveiller l'état du système et de déclencher les actions appropriées en limitant l'arrêt du système. Elle s'appuie sur la notion de diagnostic dès lors qu'il s'agit de réparer ou

de prévenir un défaut (aussi nommé défaillance). Différentes stratégies de maintenance sont admises, dont une classification peut être trouvée dans [KHV06] ainsi que d'importants états de l'art présentés dans [Rib09], [Pey09] et [Lef09]. Citons entre autres trois stratégies, préalablement introduites d'une façon générale dans [BRKG02], qui vont nous permettre d'observer que celles mettant en œuvre les méthodologies de diagnostic sont à développer en priorité :

- La maintenance corrective, qui est effectuée après occurrence d'un défaut dont les conséquences ont impacté le fonctionnement nominal du système, a pour objectif de rétablir le système de manière à ce qu'il soit capable de fournir à nouveau ses fonctions. Deux types de maintenance corrective sont établis : la maintenance curative qui permet de remettre définitivement en état le système et la maintenance palliative qui revêt un caractère temporaire et qui est principalement constituée d'opérations de dépannage qui devront toutefois être suivies d'opérations curatives.
- La maintenance préventive est basée sur des critères prédéterminés dans l'intention de réduire la probabilité des défauts du système ou la dégradation de son service rendu. Les critères sont généralement définis suivant la connaissance du temps de fonctionnement moyen du système et les pièces sont ainsi changées à chaque fois que ce délai est dépassé, qu'elles soient détériorées ou non.
- La maintenance conditionnelle ou prédictive est une maintenance préventive subordonnée à un type d'événement prédéterminé révélateur de l'état de dégradation du système. Elle consiste à surveiller et analyser en temps réel l'état d'une pièce afin de détecter l'apparition d'une dégradation (i.e. : un défaut) pour ensuite avertir les services de maintenance qui prendront la décision la plus appropriée. Cette maintenance est basée sur les techniques de diagnostic.

Nous observons bien que dans un but de rentabilité et de sûreté et sécurité du système, la troisième politique de maintenance, mettant en œuvre les méthodologies de diagnostic, est à développer en priorité. Concernant la première politique, comme elle s'effectue après conséquences d'un défaut, elle peut ainsi engendrer des coûts assez élevés que ce soit pour des pertes d'exploitation ou des destructions de composants, mais aussi engendrer des dommages sur l'environnement du système (dommage corporels, pollution, etc.). La deuxième politique, quant à elle, peut impacter le coût d'exploitation du système notamment pour des remplacements inutiles de composants.

Ainsi et dans le but de non seulement réduire les coûts d'exploitation du système (niveau rentabilité) mais aussi de prévenir ses dommages potentiels sur lui-même ou son environnement (soucis sûreté et sécurité), il est nécessaire de développer dès les phases de conception des outils de surveillance du système : outils basés sur les méthodologies de diagnostic par exemple.

1.1.2 Les concepts fondamentaux du diagnostic

Comme nous l'avons signifié au début de ce chapitre, le diagnostic a pour objectif l'identification du ou des défauts probables d'un système à partir de l'étude de leurs effets occasionnés. Le cas idéal serait bien sûr, comme l'indique [PM01], qu'il existe une correspondance biunivoque (i.e. : une relation bijective) entre les défauts potentiels du système et les effets occasionnés par ces défauts : c'est-à-dire qu'à chaque défaut serait associée une unique liste d'effets et réciproquement qu'à chaque liste d'effets serait associé un unique défaut. Il serait ainsi facile d'obtenir le défaut apparu suite à l'étude des effets occasionnés. Cela n'est néanmoins pas le cas en pratique car plusieurs défauts distincts occasionnent généralement des effets similaires voire identiques.

Le diagnostic est donc une procédure consistant à détecter et localiser un composant ou un élément en défaut d'un système. La détection désigne la capacité du diagnostiqueur à mettre en évidence l'apparition d'un ou plusieurs défauts, et la localisation désigne la capacité à être de plus capable de préciser la nature du ou des défauts apparus. La structure générale d'une procédure de diagnostic est représentée par la figure 1.1 suivante de la page 21 où le diagnostiqueur est alimenté par toute la connaissance disponible sur le système : les mesures des variables et toute autre information pouvant

être utile pour le diagnostic (un modèle par exemple). Ce diagnostiqueur traite cette connaissance et produit un « diagnostic » qui est une liste de défauts possibles pouvant affecter le système au cours du temps. Remarquons bien que même en fonctionnement normal du système, le diagnostiqueur « diagnostique » que le système fonctionne normalement.

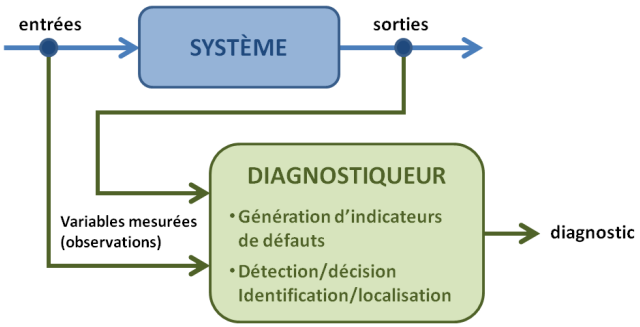

Figure 1.1 – Structure générale d'une procédure de diagnostic.

Le diagnostic repose sur les deux concepts fondamentaux suivants : d'une part signaler un défaut lorsqu'il apparaît et d'autre part le localiser le plus précisément possible afin que les actions correctrices adéquates puissent être entreprises. Ces deux concepts sont généralement spécifiés comme la *détection* et l'*isolation* de défauts. La détection consiste à reconnaître un comportement anormal du système et l'isolation consiste à déterminer quel est ce comportement anormal. Remarquons cependant que cette terminologie du diagnostic n'est pas figée et qu'il est possible de trouver différentes notions résultant toutes dans le même sens final. D'une manière générale, les étapes fondamentales du diagnostic sont les suivantes :

1. détection d'un défaut : opération permettant de décider si le système est ou n'est pas en état de fonctionnement normal ; il s'agit d'une opération logique dont la réponse doit être binaire (*oui* ou *non*).
2. localisation et estimation d'un défaut : opération permettant de déterminer l'endroit du système où se trouve le défaut (i.e. : déterminer le ou les composants en défaut), d'en déterminer sa cause ainsi que sa sévérité.
3. reconfiguration du système : opération permettant, à partir des informations fournies par les deux étapes précédentes, d'appliquer les actions correctrices au système afin soit qu'il retrouve un fonctionnement normal, soit qu'il soit mis en mode dégradé ou arrêté afin de préserver son intégrité et/ou son environnement.

La mise en place de processus permettant la réalisation de ces trois fonctions est généralement définie par l'expression *tolérance aux défauts* ([BKLS03]). Bien que la majorité des travaux distingue clairement la troisième phase de la notion de diagnostic, certains travaux n'incluent néanmoins pas non plus la fonction de détection dans le diagnostic ([BJL$^+$90] par exemple), celle-ci consistant donc juste à identifier les défauts détectés. Dans ce document, nous faisons le choix de considérer que le diagnostic est défini par ces deux étapes de détection et d'identification, dont nous nommerons la seconde par *isolation* d'un défaut.

Comme indiqué dans [BJL$^+$90], toutes ces fonctions n'apparaissent généralement pas dans chaque application de diagnostic et ceci suivant le type de défaut à surveiller, ce que nous verrons au chapitre 3 présentant une typologie générique de défauts potentiels d'un système. Cette liste d'étapes n'est par ailleurs pas complète dans le sens où il est nécessaire de rajouter en amont l'acquisition des données. Cette étape consiste, en partant des signaux électriques analogiques ou numériques, à les conditionner et les pré-traiter (extraction, amplification, atténuation, filtre, etc). Nous reviendrons brièvement sur cette étape lorsque nous présenterons au chapitre 2 la modélisation d'un système piloté.

1.1.3 Les exigences liées au diagnostic

Nous allons répertorier un ensemble d'exigences bien établies dans les travaux de diagnostic (voir par exemple [VRYK03]) et importantes à prendre en compte lors de la conception et le développement d'un diagnostiqueur. Remarquons cependant que ces exigences, à la base élaborées pour permettre la comparaison entre diagnostiqueurs conçus suivant différentes méthodologies de diagnostic, ne satisferont généralement pas toutes ces exigences inventoriées.

Par ailleurs, il est important de comprendre qu'un diagnostiqueur est au préalable basé sur une connaissance des défauts potentiels à prendre en compte. Quelle que soit la méthodologie de diagnostic adoptée, il est avant tout important de répertorier les défauts à traiter (ce que nous présenterons au chapitre 3 sur la typologie des défauts). Ces défauts pouvant être divers et variés, autant leurs formes que leurs apparitions ou leurs localisations, il est donc important de pouvoir déterminer leurs caractéristiques principales suivant la méthodologie choisie, et donc suivant l'algorithme de traitement utilisé par la méthodologie. Comme indiqué par [VRYK03], remarquons néanmoins un compromis entre la complétude et la finesse de caractérisation de cet ensemble de défauts hypothétiques répertoriés : c'est-à-dire le compromis entre le fait que cet ensemble soit d'une part complet (i.e. : permettant de diagnostiquer sans ambiguïté tout comportement anormal) et qu'il soit d'autre part « fin » dans la représentation des différents défauts (i.e. : dans leurs caractérisations). Suivant la méthodologie choisie ainsi que l'implantation du diagnostiqueur, les capacités de stockage et de calcul pourront être plus ou moins limitées. La définition d'un ensemble complet de défauts potentiels se fera nécessairement au détriment de la résolution de cet ensemble.

1.1.3.1 Exigences fonctionnelles

Les exigences fonctionnelles d'un diagnostiqueur doivent rendre compte de son fonctionnement attendu sans prendre nécessairement en compte les solutions techniques.

Rapidité du diagnostic Suite à l'occurrence d'un défaut, le temps nécessaire à sa détection et son isolation doit être rapide. Le diagnostiqueur doit donc rapidement fournir un résultat afin de prendre les décisions adéquates avant l'apparition d'effets néfastes sur le système et/ou son environnement. Cette rapidité doit, bien entendu, être mise en relation avec la sévérité du défaut et la dynamique de ses conséquences. Remarquons néanmoins que cette performance de rapidité de diagnostic peut impacter le maintien des performances du système. En effet, un diagnostiqueur conçu dans le but d'être rapide sera très certainement sensible aux bruits ou perturbations furtives (courtes et temporaires), ce qui impliquera une augmentation potentielle des fausses alarmes en fonctionnement normal et impactera ainsi les performances du système.

Discernement entre les défauts Le diagnostiqueur doit être capable de faire la différence entre plusieurs défauts. Il s'agit donc de la fiabilité de la partie isolation du diagnostic. Notons que dans le cas idéal d'absence de bruit et d'incertitudes de modélisation, cela signifie que la réponse du diagnostiqueur à un défaut est « orthogonale » aux défauts qui ne sont pas apparus. Cette exigence nécessite cependant une grande précision dans la définition des caractéristiques des défauts (lors de la phase de conception) et entraîne par conséquent une fragilité face aux incertitudes de modélisation du système et des défauts.

Identification de défauts multiples Il s'agit de la capacité du diagnostiqueur à identifier plusieurs défauts survenus simultanément ou dans une fenêtre temporelle très courte. La difficulté de cette « simultanéité » provient d'une part de l'interaction entre les conséquences des différents défauts apparus, et d'autre part de l'important volume de calcul nécessaire à une détection multiple.

Identification de nouveaux défauts Un atout supplémentaire pour un diagnostiqueur est sa capacité d'identifier de nouveaux défauts non-préalablement répertoriés. Cela signifie qu'à l'apparition d'un défaut non-répertorié, le diagnostiqueur doit être capable d'une part de le détecter (i.e. : de reconnaître que le fonctionnement du système est anormal), et d'autre part de l'isoler comme étant un défaut inconnu. Le fait de reconnaître le fonctionnement du système comme étant anormal est généralement assez simple à obtenir. Par contre pour un fonctionnement anormal, décider que le défaut est connu ou inconnu est plus délicat car l'ensemble des données nécessaires à cette décision n'est généralement pas complet : les variables nécessaires à l'observation du nouveau défaut n'ont généralement pas été surveillées, du moins pas assez longtemps.

1.1.3.2 Exigences non fonctionnelles

Les exigences non fonctionnelles d'un diagnostiqueur doivent rendre compte des propriétés qu'un tel système doit posséder.

Robustesse Le diagnostiqueur doit être robuste vis-à-vis des bruits de mesure et des incertitudes sur le modèle ou les règles de fonctionnement adoptées. Bien que l'augmentation de cette robustesse puisse être obtenue par une augmentation des seuils de tolérance, utilisés pour détecter un comportement anormal ou isoler un défaut apparu, elle peut néanmoins impacter les performances du diagnostiqueur.

Adaptabilité Le diagnostiqueur étant couplé au système durant toute la durée de son exploitation, il doit pouvoir s'adapter aux modifications du système mais aussi intégrer facilement de nouveaux paramètres ou informations obtenus après son installation. En effet, les conditions opérationnelles du système peuvent évoluer à cause de perturbations ou de changements des conditions d'exploitation. Il faut donc que le diagnostiqueur puisse s'adapter à tous changements potentiels.

Implantation Selon le type de défauts à diagnostiquer, deux types d'implantations peuvent être adoptés ([BKLS03]) : une implantation embarquée (*on-board*) ou une implantation débarquée (*off-board*). Les performances désirées du diagnostiqueur impliquent un besoin en espace de stockage et en puissance de calcul. En étant embarqué, le diagnostiqueur doit fonctionner avec des contraintes de puissance de calcul et d'espace de stockage pouvant limiter la complexité de l'algorithme de diagnostic. Un compromis peut donc être à trouver entre les performances du diagnostiqueur et les performances du matériel utilisé pour son implantation. En débarqué, le diagnostiqueur a une capacité de puissance de calcul et d'espace de stockage « quasi illimitée », mais doit néanmoins traiter des données limitées voire biaisées.

Estimation des erreurs du diagnostiqueur Il s'agit de déterminer la confiance que peut avoir l'utilisateur final du système sur la fiabilité du diagnostiqueur : ses erreurs potentielles de diagnostic (fausses alarmes, non détections ou encore mauvaises isolations). Cela peut se faire en établissant au préalable une estimation et une classification des erreurs pouvant être faites par le diagnostiqueur, ce qui pourra ainsi accroître sa fiabilité car l'opérateur aura l'opportunité de mieux interpréter les conclusions du diagnostiqueur suite à ces estimations d'erreurs.

1.2 Les méthodologies de diagnostic

Les deux principales composantes d'un système de diagnostic sont d'une part le type de connaissance *a priori* du système ainsi que des défauts potentiels, et d'autre part la technique de recherche utilisée pour diagnostiquer les défauts. Sachant que la technique de recherche dépend fortement des

connaissances disponibles il est donc plus judicieux d'adopter une classification basée sur la connaissance du système. C'est d'ailleurs traité comme cela dans [VRYK03], qui réalise un important état de l'art des différentes méthodologies de diagnostic, ou dans [PM01] qui incorpore une classification des méthodologies de diagnostic dans un important état de l'art de la surveillance automatique).

Les différentes techniques de recherche des méthodes de diagnostic ont pour principe de fonctionnement une comparaison du fonctionnement réel du système à une référence illustrant son fonctionnement normal ou ses fonctionnements anormaux. Elles exploitent donc toutes une certaine forme de redondance de l'information. Cette redondance peut être développée à partir de la compréhension du système par l'utilisation de modèles, c'est l'approche dite à base de modèles. Par contraste, elle peut provenir des expériences passées sur le système, c'est l'approche dite à base de données.

Indiquons néanmoins le premier type de redondance qui est la redondance physique ou matérielle. Cette technique consiste à doubler ou mieux tripler les capteurs, actionneurs, processeurs et logiciels du système pour mesurer et/ou contrôler des variables particulières. La comparaison des grandeurs redondantes permet ainsi de décider si un défaut est présent ou non. Néanmoins et même si cette méthode de diagnostic s'avère fiable et simple à implanter, elle entraîne bien évidemment un surcoût important en instrumentation et s'avère ainsi mise en œuvre essentiellement sur des systèmes à hauts risques tels que les centrales nucléaires ou en aéronautique.

1.2.1 Les méthodes de diagnostic à base de modèles

Les méthodes de diagnostic à base de modèles consistent à comparer le comportement réellement observé du système à un comportement prédit, issu d'un modèle de fonctionnement de ce système. Ces méthodes furent développées dès le début des années 70 avec [MP71], [CFW75] ainsi que [Wil76]. Fortement intense durant les années 80 et 90, dont les principaux travaux de référence du domaine sont [PFC89], [BN93], [Ger98] ainsi que [CP99], le diagnostic à base de modèles est toujours un domaine de recherche en expansion de nos jours. Les modèles utilisés par ces méthodes peuvent être de deux types : les modèles quantitatifs et les modèles qualitatifs.

1.2.1.1 Le diagnostic à base de modèles quantitatifs

Ce sont les méthodes utilisées par la communauté de l'automatique et plus connues sous le terme de *Model-Based Fault Detection and Isolation* (dont l'acronyme est FDI). L'utilisation d'un modèle de bon fonctionnement du système permet d'engendrer des incompatibilités entre le comportement réel du système et celui prédit par le modèle. Ces incompatibilités, appelées *indicateurs de défauts* ou *résidus*, sont générées à partir des mesures effectuées sur les variables connues du système (i.e. : ses entrées et ses sorties) et de calculs fondés sur le modèle du système. Ces résidus, notés r_i, sont des signaux devant refléter la cohérence des données mesurées du système par rapport au modèle de fonctionnement. L'objectif d'un résidu r_i est d'être sensible aux défauts : c'est-à-dire qu'il doit refléter l'éventuelle présence d'un défaut. Cela signifie donc qu'un résidu est en général proche d'une valeur de référence si aucun défaut n'affecte le système, et qu'il est dévié vers une valeur différente de celle qu'il avait lors du fonctionnement normal dès qu'un défaut apparaît. Ce sont ces résidus qui sont ensuite évalués pour réaliser les différentes fonctions du diagnostic.

Comme le montre la figure 1.2 suivante de la page 25, ces méthodes de diagnostic nécessitent deux étapes :

- La première étape génère les résidus r_i à partir des mesures effectuées sur les variables connues du système (i.e. : ses variables d'entrées et de sorties).
- La seconde étape est une règle de décision pour le diagnostic basée sur ces résidus r_i générés.

Plusieurs méthodes de génération de résidus existent et nous allons présenter les plus fréquemment utilisées.

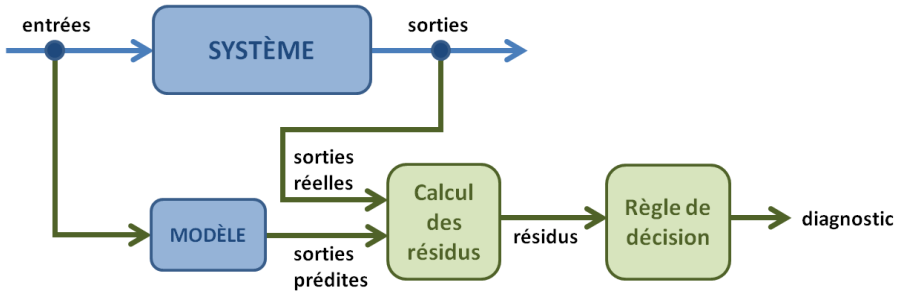

Figure 1.2 – Étapes du diagnostic à base de modèles.

Approche d'identification Il s'agit d'une approche pour laquelle le résidu est engendré par la différence entre les estimations en ligne des paramètres du modèle du système et les paramètres nominaux du système définis pour un fonctionnement normal. L'erreur d'estimation est ainsi utilisée comme résidu ([Ise84]). Bien que cette méthode soit bien adaptée aux défauts de paramètres, leurs conditions d'estimation restent néanmoins très contraignantes et le retour aux paramètres physiques du système n'est pas toujours possible ([VRYK03]).

Approche de l'espace de parité Cette approche, préalablement introduite par [Wil76] puis développée par [GS90], [GCF+95] puis [GM95], a pour principe de transformer les équations du modèle de manière à obtenir des relations particulières appelées des relations de redondance analytique (dont l'acronyme est RRA). Ces relations n'utilisent que des grandeurs connues et disponibles en ligne du système et les résidus sont alors obtenus en substituant dans ces relations les variables connues par leurs valeurs réelles, prélevées sur le système en fonctionnement. L'ensemble des valeurs que peuvent prendre les résidus compose alors un espace appelé espace de parité, comportant autant de dimensions qu'il existe de résidus. Dans cet espace, le vecteur de parité y est défini comme la valeur des résidus à un instant donné et prend alors une direction dans le cas de l'apparition d'un défaut. Tout l'enjeu consiste alors à transformer l'espace de parité de départ en un espace permettant de découpler les défauts : c'est-à-dire que chaque défaut soit uniquement représentatif d'un élément du vecteur.

Approches à base d'observateurs Les approches à base d'observateurs ou de filtres, bien connues du monde de l'automatique à des fins de commande en boucle fermée, sont les plus couramment utilisées et les premiers travaux datent des années 70 ([CFW75] et [Jon73]). Le principe général étant de concevoir un système permettant de donner une image (ou estimation) de certaines variables nécessaires au bouclage, l'adaptation à des fins de diagnostic (citons entre autres : [VS87], [DF94] [Kin03]) consiste donc à comparer les sorties mesurées avec des sorties estimées. C'est l'écart entre ces valeurs qui est alors utilisé comme résidu.

Remarques sur les méthodes à base de modèles quantitatifs L'utilisation d'un modèle précis du système, ou d'une partie du système, permet un diagnostic rapide et clair. La robustesse du diagnostiqueur sera donc étroitement liée à cette précision de la modélisation. L'adaptabilité n'est néanmoins pas grande car toute modification du système nécessite obligatoirement un retour sur le calage du modèle voire sur sa structure.

L'inconvénient majeur de ces méthodes reste néanmoins la nécessité de disposer d'un modèle relativement précis du système. Remarquons que lorsque ces méthodes utilisent des modèles linéaires, la théorie est bien développée. Par contre pour les systèmes non linéaires et comme dans le cas du contrôle-commande, il est nécessaire de travailler autour d'un point de fonctionnement pour cause de contraintes de complexité de calcul ([BJL+90]). Ceci réduit ainsi grandement le champ d'application de ces techniques.

1.2.1.2 Le diagnostic à base de modèles qualitatifs

Ce sont les méthodes généralement utilisées par la communauté de l'intelligence artificielle et dont l'acronyme est DX pour *Data eXtraction*. Les modèles qualitatifs permettent d'abstraire, à un certain degré, le comportement du système à travers des modèles de type symbolique ([TMDG97]). Ces modèles décrivent d'une manière qualitative l'espace d'état continu du système et ne représentent pas la physique du système, contrairement aux modèles quantitatifs, car ils le décrivent en terme de mode de fonctionnement. Les méthodes à base de modèles qualitatifs peuvent être classifiées de deux manières :

- Soit selon le niveau d'abstraction considéré du système à diagnostiquer. Les systèmes continus où les approches ont été développées à base de graphes causaux ([BSTMD98]) et de graphes causaux temporels ([Mos01]). Les systèmes à événements discrets, dont la référence de la littérature est ([SSL$^+$95]), consistent à représenter le système par des automates à états finis. Les systèmes hybrides dynamiques où nous pouvons trouver des méthodes reposant sur des modèles hybrides tels que les automates hybrides à temps discret ([BSMB07]), les bond-graphs ([TCDTS95]) ou les réseaux de Petri hybrides ([GG96]).
- Soit selon la prise en compte, ou non, des défauts : les modèles de dysfonctionnement comme dans les techniques de propagation des défauts ou pour les graphes causaux, ou les modèles de bon fonctionnement dans le cas du diagnostic à partir des principes premiers ou par simulation qualitative.

Modèles de bon fonctionnement Le diagnostic à partir des principes premiers est une théorie logique de diagnostic proposée par [Rei87], puis reprise par [GSW92] et étendue par [DKMR92]. Le modèle décrit la structure du système à diagnostiquer ainsi que son comportement. Cette structure représente les connexions entre les composants et est donc spécifique au système en question ; par contre les connaissances comportementales ne dépendent que du domaine choisi (électronique, mécanique, etc.). Dans cette approche, il n'est pas nécessaire de connaître *a priori* les défauts potentiels du système car toute contradiction entre les observations du système et la prédiction du modèle de bon fonctionnement est obligatoirement la manifestation d'un défaut.

Une autre approche possible est le diagnostic par modélisation et simulation qualitative ([Kui86]). Il s'agit de transformer une représentation d'un système continu par des équations différentielles en un ensemble de contraintes qualitatives qui peuvent être traitées par des outils spécifiques de simulation qualitative : l'outil QSIM par exemple permet de prédire l'état qualitatif du système (en termes de stabilité, croissance ou décroissance des variables) à partir de conditions initiales données.

L'avantage d'un diagnostic à base de modèle de bon fonctionnement est de ne pas avoir besoin d'une connaissance exhaustive des défauts du système. Par rapport aux méthodes quantitatives à base de modèle qui utilisent aussi des modèles de bon fonctionnement, les méthodes qualitatives nécessitent par ailleurs moins de connaissances précises sur le système : la surface exacte d'un réservoir n'est pas utilisée pour le diagnostic d'une fuite par exemple.

Modèles de dysfonctionnement Comme nous venons de le signifier, les méthodes de diagnostic à base de modèles qualitatifs de dysfonctionnement peuvent utiliser des modèles de propagation des défauts ou les graphes causaux.

Concernant les modèles de propagation des défauts, une des techniques est l'arbre de défauts ([LP77]) qui est construit à partir de questions du type « quelles sont les causes d'un événement de haut niveau ? », et dont la réponse est en général la combinaison logique (par utilisation des opérateurs « ET » et « OU ») d'autres évènements. L'arbre est alors prolongé jusqu'aux évènements atomiques qui ne peuvent être décomposés. L'évaluation qualitative de l'arbre revient à rechercher l'ensemble minimal d'évènements nécessaires et suffisants pour provoquer un événement constaté de haut niveau. Une évaluation quantitative à base de probabilités sur les évènements primitifs peut aussi être utilisée

pour calculer la probabilité de l'évènement de haut niveau. Pour un bon diagnostic à partir d'un arbre de défauts il faut un modèle qui représente convenablement les relations causales du procédé, en particulier une bonne connaissance de la propagation des défauts.

Un graphe causal est une représentation graphique des interactions entre les variables du système en présence d'un défaut. La méthode la plus connue est Digraph ([IAOM79]) dans laquelle les nœuds représentent des variables ou des évènements et les arcs orientés les relations entre ces variables. Chaque arc est orienté du nœud « cause » vers le nœud « conséquence ». L'arc est de plus signé : le signe représente le sens de variation. Notons par ailleurs qu'un modèle Digraph peut être obtenu soit à partir d'un modèle mathématique soit à partir de relevés expérimentaux.

La principale limitation de ces méthodes est la nécessité d'un grand nombre d'hypothèses. En effet, une faible résolution entraînerait une grande incertitude dans le diagnostic et le volume de calcul est de ce fait relativement important. Ces méthodes permettent par contre de trouver tous les défauts possibles et leur raisonnement naturel assure une explication complète du défaut détecté.

1.2.2 Les méthodes de diagnostic à base de données

Contrairement aux méthodes à base de modèles, celles à base de données reposent sur un nombre important de données qui sont supposées représenter convenablement le système. Les seules informations disponibles sont les signaux issus des capteurs du système, ce qui implique que ces approches présupposent donc que ce système puisse être complètement décrit par ses observations passées et présentes. L'objectif de ces approches est alors de construire un modèle ajusté sur les données collectées, et la principale difficulté va donc être de définir non seulement la structure appropriée du modèle, mais aussi le calage approprié entre ce modèle et le système.

1.2.2.1 La reconnaissance de formes

L'objectif de la reconnaissance de formes est de classifier des objets, nommés des *formes* et qui sont représentées par des données, dans des classes prédéterminées en les comparant à des prototypes. Cette méthode repose donc sur une description complète de ces formes et de chacune des différentes classes prototypes. Un problème de diagnostic peut ainsi se définir comme un problème de reconnaissance de formes où les classes sont les modes de fonctionnement du système (nominal ou sous la présence de défauts) et les formes sont représentées par les observations du système ([Dub90]).

Un système de diagnostic conçu suivant une approche par reconnaissance de formes comporte en principe deux étapes. La première, dite étape d'apprentissage, consiste à définir les différentes observations constituant la forme et les classes connues du système, puis à construire une règle de décision précisant les frontières entre ces classes. La seconde étape, dite de décision, consiste à décider si des observations appartiennent à une des différentes classes : c'est-à-dire de rechercher les prototypes des différentes classes équivalents à ces observations.

Différentes approches peuvent être utilisées lors de l'étape d'apprentissage : les réseaux de neurones, les réseaux bayésiens, ou encore la logique floue.

1.2.2.2 Les systèmes experts

Les systèmes experts sont utilisés dans des applications où l'expertise humaine y est importante et le développement de modèles y est faible. Ce sont des systèmes à base de règles du type « `si` » « `et` » « `ou` » « `alors` » qui utilisent une information heuristique pour lier les symptômes aux défauts, établissant ainsi des associations empiriques entre effets et causes des défauts ([Far89]). Ces associations sont généralement fondées sur l'expérience de spécialistes, dits *experts*, plutôt que sur une connaissance de la structure et/ou du comportement du système. Leur fonctionnalité est de trouver la cause de ce qui a été observé en parcourant, par un raisonnement abductif, les règles préalablement établies.

L'attrait d'un système expert découle de son architecture qui sépare explicitement la connaissance du système, en utilisant un langage naturel, du mécanisme de raisonnement. Ceci rend donc possible l'évolution des connaissances du système sans avoir à agir sur le mécanisme de raisonnement. La limitation principale des systèmes experts réside dans les connaissances requises à leur élaboration, qui nécessitent d'identifier *a priori* les défauts du système et de les traduire sous forme de règles simples. Or d'une part le recensement des défauts ne peut être exhaustif, d'autre part toutes les expertises ne sont pas formalisables sous forme de règles.

1.3 La notion de diagnosticabilité

La notion de diagnosticabilité exprime un problème fondamental lors de la conception d'un système de diagnostic. Il s'agit de s'assurer qu'en fonctionnement, le diagnostiqueur sera toujours capable de diagnostiquer sans aucune ambiguïté un défaut préalablement répertorié. Cela signifie d'une part qu'il sera toujours capable de détecter un tel défaut préalablement répertorié lorsqu'il apparaît, et d'autre part qu'il sera toujours capable de l'identifier sans ambiguïté par rapport aux autres défauts préalablement répertoriés. Le premier cas revient donc à s'assurer qu'un défaut engendre toujours un comportement observable anormal, et le second revient à s'assurer que plusieurs défauts distincts n'engendrent pas un même comportement observable.

Bien que récente comparée à la notion de diagnostic, la diagnosticabilité a été largement étudiée par les deux communautés du diagnostic à base de modèles (la communauté DX, pour *Data eXtraction*, issue de l'intelligence artificielle et la communauté FDI, pour *Fault Detection and Isolation*, issue de l'automatique), que ce soit pour les systèmes discrets, les systèmes continus, ainsi que ces dernières années pour les systèmes hybrides. Les autres communautés n'ont semble-t-il pas investi ce champ de recherche, du moins n'avons-nous pas trouvé d'article significatif sur ce sujet. De notre point de vue, cela résulte du fait que comme une telle étude doit prouver le bon fonctionnement du diagnostiqueur, elle ne peut donc se réaliser qu'en phase de conception. L'utilisation de modèles de fonctionnement du système y est ainsi fortement utile car la possibilité de réaliser cette étude directement sur le système physique, ou des maquettes du système, impliquerait une augmentation du coût de conception et de développement du système.

Ces deux communautés du diagnostic à base de modèles ont donc développé deux approches, parallèles mais distinctes, d'étude de la diagnosticabilité dans le cas continu et dans le cas discret. Ces approches reprennent les formalismes utilisés pour le diagnostic : utilisation des machines à états finis pour le cas discret, et utilisation de modèles basés sur des équations algébrico-différentielles ou sur une abstraction qualitative du système pour le cas continu. Dans les deux cas, la définition classique de la diagnosticabilité est donnée par un concept de signature de défaut. L'approche du cas continu est à base d'états dans le sens où le diagnostic est accompli par une analyse de l'image instantanée des observations (i.e. : une observation à chaque instant du temps) ; alors que l'approche du cas discret est à base d'évènements et considère un suivi des états du système, ce qui signifie l'ajout d'une vision temporelle. Par la suite, un grand travail de la part de ces deux communautés a permis d'uniformiser cette notion de diagnosticabilité entre le cas discret et le cas continu (voir [CTMP06]). Le cas hybride a lui aussi été investi par ces communautés en faisant, comme nous allons le présenter un mélange entre la méthode du cas continu et celle du cas discret.

1.3.1 Diagnosticabilité dans le cas discret

Pour les systèmes à évènements discrets, la notion de diagnosticabilité fut introduite par [SSL+95] puis fut largement étudiée par la suite (voir [JHCK01] et [JK04] pour de bonnes présentations).

Ce type de système est défini par un ensemble fini d'états ainsi qu'un ensemble fini de transitions (observables ou non) entre les états. Un défaut du système est une transition non-observable spécifique.

La diagnosticabilité d'un défaut y est ainsi définie comme la non existence de deux suites finies de transitions, dont l'une passe par la transition du défaut considéré et l'autre non, et qui sont telles que les projections de ces deux suites uniquement sur les transitions observables sont identiques. L'utilisation d'un algorithme de recherche permet ensuite d'analyser ce système afin d'y rechercher de telles suites. Le problème majeur de cette approche concerne le cas des systèmes complexes où les modèles utilisés peuvent être importants et ainsi impacter la complexité en espace de l'algorithme de recherche. Différents travaux présentant des algorithmes en temps polynomiaux ont été développés afin de tenter de résoudre ce problème de complexité (comme dans [JHCK01] et [YL02]).

D'autre solutions ont par ailleurs été données. Une première approche, donnée dans [QK06] et [SP07], consiste à travailler de manière locale dans le système en définissant des diagnosticabilités locales, puis de définir, par un mécanisme de distribution, une diagnosticabilité globale suivant les résultats des diagnosticabilités locales. Une autre approche, donnée dans [CPC03] et [BDNR09], consiste à utiliser des systèmes de transitions plus expressifs en utilisant des variables symboliques et non plus numériques. La vérification de la diagnosticabilité est basée de manière formelle sur les techniques de model-checking (voir [CGP00] et [BBF$^+$01] pour une bonne introduction aux techniques de model-checking).

1.3.2 Diagnosticabilité dans le cas continu

Dans le cas des systèmes continus, un nombre important de travaux ont étudié la diagnosticabilité par une approche de placements de capteurs ([CPR00], [SRB$^+$02] et [DS03] pour la communauté DX et [Car99], [LMR97], [GLR00] et [KN02] pour la communauté FDI).

L'idée principale, explicitée dans [TMEO06], suit une approche familière à la communauté FDI qui est l'analyse structurelle. Cette idée, préalablement introduite dans [CS97]), consiste à utiliser un modèle comportemental du système et d'analyser, de manière exhaustive, les relations de redondance analytique introduites par les capteurs (réels et hypothétiques en supposant que toutes les variables du système sont mesurées) puis de construire une matrice de signature de défauts hypothétiques. L'étude de la diagnosticabilité se base ainsi sur une étude de matrice.

1.3.3 Diagnosticabilité dans le cas hybride

Dans le cas des systèmes hybrides, quelques travaux ont formalisé cette notion ([WN96], [BDTM02] et [BTM03]).

Notons que [DBDGD07a] et [DBDGD07b] traitent le cas où le système hybride est abstrait par un automate temporisé. La notion de diagnosticabilité définie correspond donc, en substance, à la notion de diagnosticabilité discrète.

Une approche clairement identifiée dans [BOTM08] consiste à coupler, en deux temps, les techniques d'étude de la diagnosticabilité dans les cas continu et discret. La première phase constiste à étudier la diagnosticabilité dans le système multimode sous-jacent (chaque état discret du système constitue un mode de fonctionnement) : il s'agit de garantir qu'il est possible d'identifier et de discriminer, sans ambiguïté, chaque mode à partir des observations. La seconde phase consiste à étudier la diagnosticabilité de chacun des sous-systèmes continus (pour chacun des modes de fonctionnement) suivant l'approche d'étude de la diagnosticabilité dans le cas continu, mais en introduisant en supplément une notion d'indicateur de consistance pour chaque autre mode du système. Finalement la notion de diagnosticabilité du système hybride est donnée en couplant les deux définitions.

1.3.4 Étude de la diagnosticabilité

S'assurer de la diagnosticabilité d'un système signifie, comme nous l'avons introduit, de s'assurer que le diagnostiqueur sera toujours capable de diagnostiquer sans aucune ambiguïté un défaut

préalablement répertorié. Ce la implique donc non seulement que le procédé d'analyse du diagnostiqueur (i.e. : ses règles d'analyse) soit assez « subtil » pour diagnostiquer les défauts mais aussi que l'information disponible du système soit assez explicite.

Bien que l'étude de la diagnosticabilité soit, comme nous venons de la présenter, parfaitement et formellement définie pour les différents types de systèmes considérés (discrets, continus ou hybrides), le passage au diagnostiqueur (i.e. : la génération du diagnostiqueur conformément à l'étude de la diagnosticabilité) n'est que rarement, voire aucunement défini et/ou développé. Cela ne signifie néanmoins pas que ce passage n'est pas possible suite à l'étude.

Il semble en effet que ces études se basent sur un cas « idéal » de diagnostiqueur ayant la capacité de connaître l'état courant du système surveillé, de calculer l'état de référence ainsi que tous les états de défauts, puis de tous les comparer. En pratique, ce type « idéal » de diagnostiqueur risque de ne pas être réalisable car il faudrait alors soit qu'il simule beaucoup de modèles en même temps, soit qu'il puisse stocker un nombre important de données. Ces études permettent ainsi de s'assurer que si un défaut n'est pas diagnosticable pour un tel diagnostiqueur « idéal », il ne le sera alors pas pour toute solution de fonctionnement d'un diagnostiqueur ayant accès qu'à une partie moins précise de l'état du système.

De notre point de vue cependant, il semble nécessaire qu'une telle étude soit fondée explicitement suivant le procédé d'analyse d'un tel diagnostiqueur « réaliste ». En effet, l'étude de la diagnosticabilité doit permettre de s'assurer que lors du fonctionnement du système, le diagnostiqueur diagnostiquera sans ambiguïté son fonctionnement (fonctionnement nominal ou fonctionnement avec un défaut). L'étude de la diagnosticabilité est donc inhérente au procédé d'analyse du diagnostiqueur, et c'est bien en partant de celui-ci que doit se conduire une étude de diagnosticabilité afin qu'elle débouche, de manière rigoureuse, sur la génération du diagnostiqueur conformément aux résultats obtenus.

1.4 Conclusion sur les méthodologies de diagnostic

Dans ce chapitre, nous venons de présenter les différentes méthodologies de diagnostic existantes ainsi que celles permettant l'étude de la diagnosticabilité d'un système. La diversité des méthodes existantes a permis de nous rendre compte de l'intérêt porté à cette thématique depuis de nombreuses années.

Nous avons pu remarquer que, bien que le diagnostic possède un socle commun de concepts fondamentaux, ce que nous avons présenté en début de ce chapitre, la majorité des méthodes présentées ont toutes une approche spécifique suivant le formalisme considéré dans la représentation du système, ou dans la représentation de la connaissance ou encore dans la technique de recherche utilisée.

Concernant les méthodes d'étude de la diagnosticabilité, qui sont quant à elles réalisées durant la phase de conception du système, nous avons ainsi pu remarquer que ce sont avant tout les méthodologies de diagnostic à base de modèles qui furent adaptées pour mener cette étude. En effet, l'utilisation de modèle se prête bien à toute étude à mener sans avoir nécessairement à disposition le système réel. Nous avons néanmoins pu nous apercevoir que le lien entre cette étude de diagnosticabilité et le passage au diagnostiqueur (i.e. : la génération d'un diagnostiqueur issu de l'étude de la diagnosticabilité) n'est souvent pas clairement établi.

Suite à ce chapitre introductif, nous allons pouvoir élaborer notre méthodologie, fondée sur un cadre théorique commun, permettant de concevoir un diagnostiqueur issu d'une étude de la diagnosticabilité de défauts préalablement identifiés.

Chapitre 2

Les systèmes technologiques pilotés

Nous allons introduire dans ce chapitre les systèmes pour lesquels nous souhaitons y intégrer un diagnostiqueur : les *systèmes technologiques pilotés*. Nous allons commencer par introduire les notions utiles à la présentation de ces systèmes, ce qui nous permettra de considérer leur architecture, puis nous présenterons comment les représenter de manière générique par les outils de modélisation. Enfin, nous présenterons un exemple applicatif de système technologique piloté qui sera utilisé dans toute la suite de ce document.

2.1 Introduction aux systèmes technologiques pilotés

Dans cette partie, nous introduisons les notions utiles permettant de présenter ce qu'est un système technologique piloté. Nous entrerons ensuite plus en détail dans l'architecture de ce type de système.

2.1.1 Des systèmes mécaniques aux systèmes technologiques pilotés

En partant d'une notion générale de système, nous nous orienterons vers les systèmes que nous souhaitons traiter : les systèmes technologiques pilotés, en faisant le lien entre les systèmes mécaniques et l'intégration de composants électroniques et informatiques permettant de définir les systèmes mécatroniques. Nous verrons que ce que nous allons considérer comme système technologique piloté n'est tout simplement qu'un système mécatronique dans lequel nous distinguons particulièrement la partie pilotage de la partie opérante.

2.1.1.1 La terminologie liée aux systèmes

D'une manière générale, un système est un regroupement d'éléments en interaction et organisés dans un environnement avec lequel il interagit pour réaliser une fonction qui lui est attribuée ([dR75]). Cette définition générale est utilisée par de nombreuses disciplines scientifiques telles que, entre autres, les sciences de l'ingénieur, les sciences physiques ou encore les sciences économiques et sociales. Dans notre approche orientée pour les systèmes habituellement étudiés en ingénierie, nous allons considérer les deux définitions suivantes :

- Selon [BJL$^+$90], un *système* est un objet physique ou un ensemble de phénomènes qui comportent des relations de cause à effet.
- Selon [Ise05], un *système mécatronique* (ou *mechatronic system* en anglais) résulte de la conception et de l'intégration simultanées de composants mécaniques, électroniques et informatiques (dans le sens traitement de l'information). Cette intégration, qui se fait entre des composants physiques (la partie « matériel », *hardware* en anglais) et des fonctions de contrôle et transmission de l'information (la partie « logiciel », *software* en anglais), est orientée vers la recherche de

l'équilibre optimal entre la structure mécanique élémentaire, l'implantation de capteurs et d'actionneurs, ainsi que le contrôle automatique de l'information. Des effets synergiques, résultant de fonctionnalités améliorées et de solutions innovantes, sont par ailleurs créés.

Ces deux définitions sont, de notre point de vue, équivalentes dans le sens où elles traitent des mêmes catégories de systèmes. Cependant, la première est moins restrictive dans le sens où les systèmes purement mécaniques (un réservoir ou une résistance de chauffe par exemple) rentrent dans le cadre de cette définition. Ainsi le point essentiel d'un système mécatronique est la combinaison appropriée de processus mécaniques, électroniques et informatiques (dans le sens technologie de l'information) s'influançant mutuellement les uns des autres. La figure 2.1 ci-dessous, inspirée de [Ise05], reprend les interactions entre ces différents domaines, avec leurs différentes disciplines potentiellement utilisées, permettant de définir la *mécatronique*.

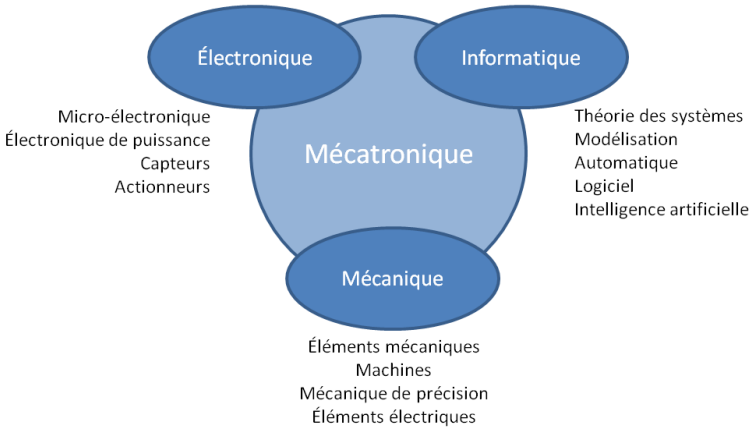

Figure 2.1 – Interaction des différents domaines de la mécatronique.

2.1.1.2 Retour sur le contrôle des systèmes mécaniques

Dans le but d'imposer le comportement d'un système mécanique pour qu'il réalise une fonction d'usage, il est nécessaire de le contrôler. C'est-à-dire de prendre le contrôle d'une ou plusieurs grandeurs physiques (vitesse, température, pression ou courant électrique par exemple) du système en les mesurant afin de vérifier leurs états, puis de déterminer à l'aide d'un traitement approprié l'action à entreprendre sur le système pour qu'elles se comportent comme souhaité ([Eti11]). D'une manière générale, nous imposons d'une part qu'une certaine grandeur physique du système ait une valeur moyenne donnée en régime permanent et ce malgré l'influence de l'environnement (i.e. : les perturbations externes), et d'autre part que cette même grandeur physique passe d'une valeur à une autre en un temps donné, voire avec un profil de variation imposé.

Le contrôle d'un tel système mécanique peut être manuel ou automatique, c'est-à-dire sans qu'aucune intervention manuelle ne soit nécessaire pour atteindre l'objectif ciblé. Remarquons que les méthodologies de contrôle-commande automatique permettent de traiter des situations impossibles à contrôler manuellement car très rapides (ayant des constantes de temps $t < 1[s]$) ou précises (avec des écarts de cible très faibles) et devant être rendues stables afin d'être utilisables.

Historiquement, le contrôle des systèmes mécaniques s'est développé par vagues successives passant d'un contrôle mécanique (utilisation de câbles, vérins) à un contrôle numérique par micro-contrôleurs [Ise05]. Remarquons donc les passages successifs du contrôle électrique (1920) à l'intégration des relais électromécaniques, des amplificateurs électriques ainsi que des contrôleurs PI (1930), puis le développement des transistors ou thyristors (1955). Suivirent ensuite l'intégration des premiers calculateurs

numériques avec leurs logiciels temps réels (1975), pour enfin arriver à la généralisation des micro-contrôleurs numériques, des ordinateurs et l'intégration directe des actionneurs et des capteurs dans le système pour former les systèmes mécatroniques.

2.1.1.3 Des systèmes mécatroniques aux systèmes technologiques pilotés

Suivant le choix des systèmes que nous souhaitons traiter, ce qu'il faut retenir de la définition d'un système mécatronique, outre qu'elle soit orientée sur les systèmes étudiés par les sciences de l'ingénieur, est qu'un tel système est un assemblage complexe et structuré de composants mécaniques, électroniques et informatiques en interaction permanente et assurant une fonction d'usage. Le terme *complexe* est ici utilisé dans le sens où les différents composants sont assemblés par des liens (matériels et/ou immatériels) nombreux, diversifiés et présentant des spécificités différentes. Un tel système est donc déterminé par la nature de ses différents composants, les interactions entre ces composants et leurs critères d'appartenance au système permettant de déterminer si un composant appartient au système ou fait au contraire partie de son environnement.

Nous en arrivons donc à la définition de la notion de *système technologique piloté*, obtenue d'un système mécatronique par considération de la représentation que nous en avons de son architecture :

Définition 2.1 *Un* système technologique piloté *est un système mécatronique pour lequel nous distinguons particulièrement la partie pilotage, que nous nommons le* système de pilotage*, de la partie opérante que nous nommons le* système opérant*.*

Remarquons bien que la différence entre un système mécatronique et un système technologique piloté se fait uniquement sur la séparation faite dans l'architecture du système : la distinction particulière entre la partie pilotage et la partie opérante. Il s'agit ainsi uniquement d'un point de vue. Par ailleurs avec l'utilisation de cette expression *système technologique piloté*, nous souhaitons mettre en avant le fait qu'un tel système permet d'accomplir ses fonctions de manière autonome suivant les consignes reçues d'un opérateur (humain ou même un autre système piloté).

2.1.2 Les systèmes technologiques pilotés

Un système technologique piloté, que nous abrégeons par *système piloté* et que nous notons classiquement S, est donc un système mécatronique pour lequel nous faisons une séparation de son architecture en deux parties : une partie pilotage, nommée *système de pilotage* et notée SP, et une partie opérante, nommée *système opérant* et notée SO, assurant une fonction d'usage. La figure 2.2 ci-dessous représente un tel système technologique piloté ayant une commande en boucle fermée.

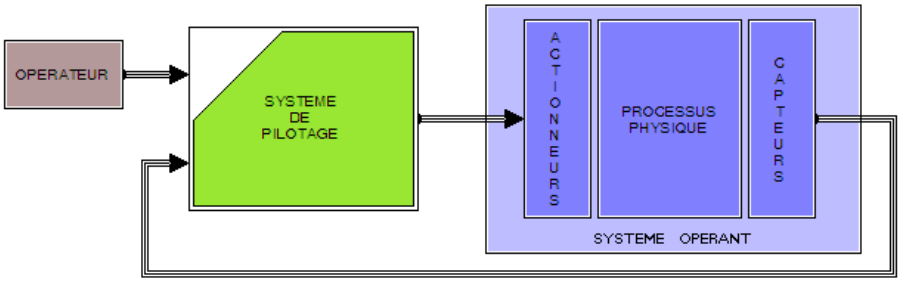

Figure 2.2 – Architecture d'un système piloté.

2.1.2.1 Le système opérant

Le système opérant SO d'un système piloté assure l'action du système, c'est-à-dire la fonction pour laquelle le système est utilisé. Cette partie est composée de trois types de composants : les actionneurs, les capteurs et divers composants physiques formant la partie que nous nommons le *processus physique*. Le rôle des capteurs est de transmettre au système de pilotage les informations sur le fonctionnement du processus physique. Il s'agit des mesures y de certaines grandeurs physiques du processus physique. Le rôle des actionneurs est de faire évoluer les grandeurs physiques du processus physique suivant les commandes u reçues du système de pilotage.

La figure 2.3 ci-dessous représente la vision architecturale du système opérant. Bien que ne faisant pas partie du système piloté en lui-même, donc du système opérant, nous avons néanmoins intégré l'environnement externe dans cette figure afin d'indiquer d'où venaient les perturbations potentielles impactant le système opérant. Il s'agit de l'architecture largement ancrée dans le monde de l'automatique (voir [Ise05] ou [Bub05], par exemple) et que nous retrouvons aussi dans les méthodologies de diagnostic issues de l'automatique ([Ise06], [BKLS03] ou [BJL+90] par exemple).

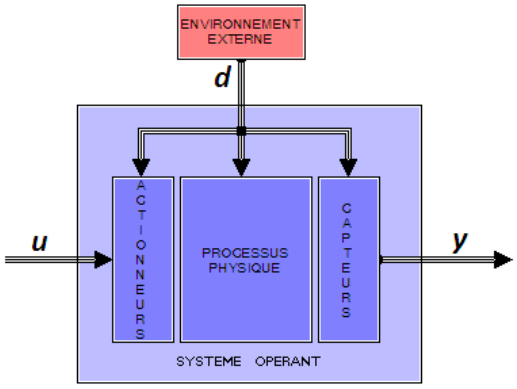

Figure 2.3 – Représentation du système opérant.

2.1.2.2 Le système de pilotage

Le système de pilotage SP d'un système piloté assure le contrôle du système opérant. Il s'agit de la logique de fonctionnement du processus physique qui doit être automatisé. Suivant d'une part la consigne c fournie par l'environnement de décision (i.e. : l'opérateur) et d'autre part suivant l'information y sur le fonctionnement du processus physique obtenue par les capteurs, le système de pilotage contrôle le système opérant en calculant les commandes u des actionneurs nécessaires au bon fonctionnement du processus physique.

Les différents niveaux de contrôle automatique Comme nous venons de l'introduire, le système de pilotage assure le contrôle automatique du système opérant. Or le contrôle automatique d'un système automatique peut être classiquement décomposé en deux niveaux représentés par la figure 2.4 suivante de la page 35 : le niveau de supervision et le niveau de régulation.

À l'intérieur même du système de pilotage se trouvent, entre autres, différents modules suivant ces différents niveaux de contrôle automatique : un module de régulation et un module de décision. Le module de régulation contient les lois de commande du système opérant ainsi que les interfaces d'entrées et sorties de et vers le système opérant (convertisseurs analogique-numérique et numérique-analogique ainsi que les filtres de traitements). Le module de décision fait le lien entre l'environnement de décision du système (l'opérateur) et la boucle de régulation : c'est-à-dire la gestion des différents modes de fonctionnement du système.

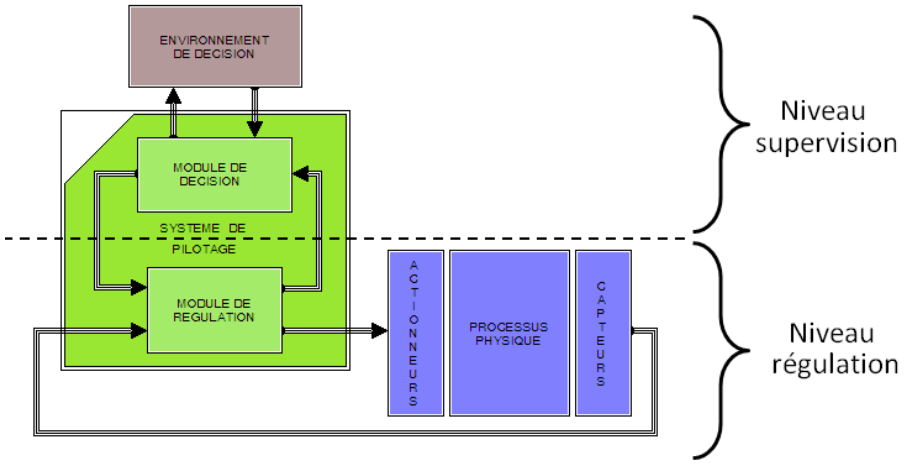

Figure 2.4 – Niveaux de contrôle des systèmes autonomes.

Dans la suite, le module de décision pourra être nommé le *superviseur* du système. L'environnement de décision, qui ne fait pas partie du système piloté, sera nommé *opérateur* ; tout en gardant à l'esprit qu'il ne s'agit pas obligatoirement d'une personne, ce peut très bien être le superviseur d'un système de plus haut niveau.

La régulation numérique Le module de régulation du système de pilotage contient les lois de commande permettant de contrôler le système opérant. Comme nous l'avons vu dans la partie précédente : depuis l'introduction des micro-contrôleurs numériques, le procédé de régulation du système opérant se fait de manière numérique. Or en régulation numérique ([Eti11]), le régulateur est réalisé sous la forme d'un algorithme de traitement (les lois de commande), donc implémenté dans un langage de programmation (en langage C par exemple), s'exécutant à intervalles réguliers $h[s]$ où h est la période d'échantillonnage. Cela signifie que la mesure $y(t)$ venant des capteurs du système opérant (i.e. : la grandeur réglée) est échantillonnée : c'est-à-dire qu'elle n'est observée qu'aux instants d'échantillonnage $0 \cdot h$, $1 \cdot h$, $2 \cdot h$, ..., $\mathrm{k} \cdot h$, ..., auxquels une conversion analogique-numérique est effectuée. L'algorithme de régulation est alors exécuté afin de délivrer une commande $u(k \cdot h)$ (i.e. : la grandeur de commande) à intervalles également réguliers h. Cette commande est ensuite convertie, par une conversion numérique-analogique, sous la forme d'un signal continu dans le temps. La figure 2.5 ci-dessous schématise ces différentes conversions entre le système opérant et le module de régulation (les lois de commande).

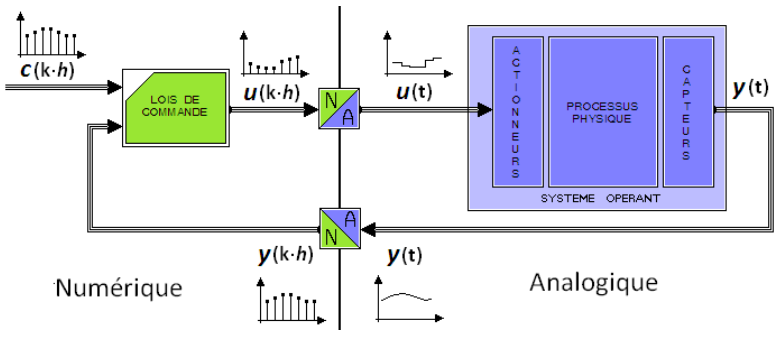

Figure 2.5 – Principe de la régulation numérique.

Les lois de commande du système opérant Nous ne rentrons pas dans le détail des différentes méthodologies de contrôle commande d'un système car la littérature est abondante dans ce domaine (citons par exemple [LR08] ou [Eti11] pour une introduction à la modélisation et à la commande de systèmes dynamiques, ainsi que [PR08] et [BR03] pour une présentation beaucoup plus complète et approfondie de ce domaine). Nous considérons juste que les lois de commande du système sont déjà élaborées. Elles peuvent être basées sur les méthodologies bien connues de la communauté de l'automatique. Citons entre autre exemple :

- la régulation PID, pour « proportionnel intégral dérivé » (voir [JM05] pour une explication détaillée), qui calcule la commande u en fonction de l'erreur entre la consigne c et la mesure y suivant trois objectifs. Qu'elle soit proportionnelle à l'erreur. Qu'elle soit aussi calculée en intégrant l'erreur (i.e. : la commande augmente en permanence si l'erreur est constante en espérant ainsi que l'erreur finisse par décroître). Qu'elle soit enfin proportionnelle à la dérivée de l'erreur (i.e. : une accélération de la correction est ainsi créée dans le cas où l'erreur s'accroît brutalement).
- la régulation prédictive à base de modèles (voir [Ric97] pour une courte introduction et [RO09] pour une explication détaillée avec des exemples applicatifs) se base sur l'utilisation d'un modèle dynamique du système pour anticiper son comportement futur et calculer ainsi la commande u optimale au fonctionnement du système suivant la consigne c ciblée par l'opérateur et la mesure y.

Comme l'indique [Ric97], la régulation PID est largement utilisée dans le milieu industriel de par son efficacité remarquable et son rapport prix/performance très avantageux lorsqu'il est possible de l'appliquer. Elle ne couvre néanmoins pas tous les besoins et ses performances atteignent leurs limites dans plusieurs cas : lorsque les processus sont « difficiles » à contrôler (non linéaires, instables, non stationnaires, à grand retard pur ou encore multi-variables par exemple) ou lorsque les performances exigées par l'utilisateur sont complexes (forte atténuation des perturbations ou réponse en temps minimal par exemple) ce qui amène à fonctionner sur des contraintes qui affectent soit les variables de commande u, soit des variables internes x du processus.

Notons que le cas d'étude du document, que nous allons présenter dans la suite de ce chapitre, utilise une commande prédictive à base de modèles dont nous en ferons une courte présentation.

Architecture du module de régulation L'architecture du module de régulation du système de pilotage est présenté par la figure 2.6 ci-dessous. Ce module reçoit en entrées la consigne de fonctionnement du module de décision ainsi que les mesures venant des capteurs du processus physique. Ces mesures sont d'abord échantillonnées et converties en format numérique par le convertisseur analogique-numérique, puis ensuite filtrées par le module de filtrage. Le module de régulation émet en sortie les commandes des capteurs ainsi que différentes informations sur le système opérant (par exemple ces commandes ainsi que ces mesures filtrées). Ces commandes sont ensuite converties, par un convertisseur numérique-analogique, sous la forme d'un signal continu dans le temps.

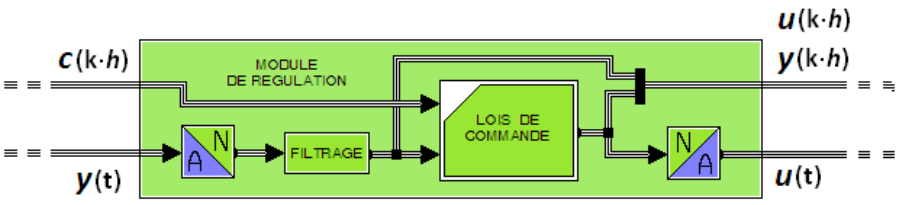

Figure 2.6 – Architecture du module de régulation du système de pilotage.

Hypothèse de régulation du système opérant Bien que nous ayons abordé les notions liées au contrôle-commande des systèmes, nous n'avons pas traité les vérifications préalablement nécessaires et classiquement réalisées en automatique : étude de la stabilité d'un système, de son observabilité ou de sa commandabilité par exemples. Nous supposons pour la suite que ces critères (pas nécessairement tous) ont été étudiés pour le système opérant et ainsi qu'une régulation a été élaborée.

Nous pouvons donc introduire la notion de *dynamique de réponse* du système. Il s'agit du temps $\beta \in \mathbb{T}_\iota$ que met le système pour atteindre n'importe quelle consigne donnée par l'opérateur. Pour la suite, nous la nommerons aussi *dynamique* du système

2.1.2.3 Architecture d'un système piloté

Différentes architectures du système piloté peuvent être considérées suivant le type d'informations prises en compte du processus physiques ([Bub05]) :

- Système en boucle ouverte sans mesure des perturbations externes.
- Système en boucle ouverte avec mesure de certaines perturbations externes.
- Système en boucle fermée sans mesure des perturbations externes.
- Système en boucle fermée avec mesure de certaines perturbations externes.

Le sens de « *mesure de certaines perturbations externes* » signifie que tout ou une partie des perturbations externes sont mesurées.

La figure 2.7 ci-dessous reprend ces différentes architectures possibles d'un système piloté. En haut à gauche est représenté le système en boucle ouverte sans mesure de perturbations alors qu'à droite est représenté le système en boucle ouverte avec mesure de perturbations. En bas à gauche est représenté le système en boucle fermée sans mesure de perturbations alors qu'à droite est représenté le système en boucle fermée avec mesure de perturbations.

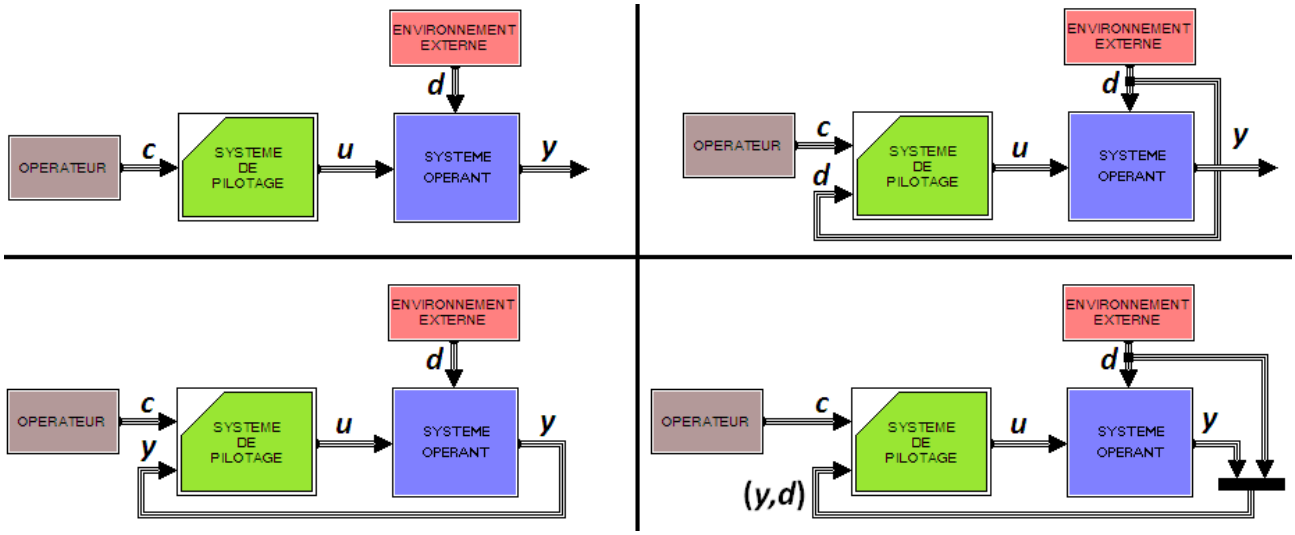

Figure 2.7 – Les différentes architectures possibles d'un système piloté.

Pour la suite du document, nous faisons la supposition que le système est en boucle fermée avec potentiellement des mesures de certaines perturbations externes. Bien que les mesures du système opérant et de certaines perturbations externes soient différenciées sur la figure précédente, nous pouvons, pour éviter d'alourdir les figures, supposer que les mesures y du processus physique prennent aussi en compte les mesures de certaines perturbations externes d. Par ailleurs, bien que la méthodologie présentée par la suite sera décrite en prenant en considération un système technologique piloté en boucle fermée avec de potentielles mesures de certaines perturbations externes, cette méthodologie sera tout à fait applicable aux autres architectures que nous venons de présenter.

La figure 2.8 ci-dessous représente l'architecture d'un tel système piloté. Nous observons bien la distinction entre la partie pilotage de la partie opérante constituée des actionneurs, du processus physique et des capteurs. Bien que ne faisant pas partie du système piloté en lui-même, nous avons intégré l'opérateur dans cette figure afin d'indiquer d'où vient la consigne fournie au système. Les perturbations externes, quant à elles, s'appliquent à toutes les parties du système opérant : les actionneurs, le processus physique et les capteurs.

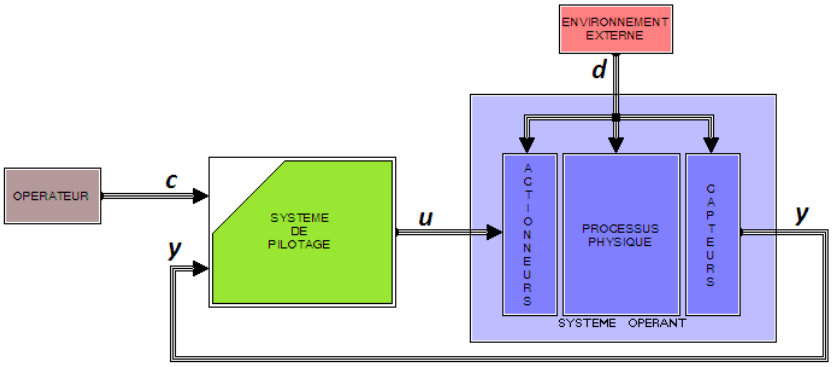

Figure 2.8 – Architecture d'un système piloté.

2.2 Représentation d'un système piloté

Dans cette partie, nous allons décrire la représentation d'un système piloté : c'est-à-dire la manière dont nous allons le modéliser. Nous allons de ce fait considérer l'architecture du système piloté selon la « vision » qu'en aura le diagnostiqueur.

Nous ferons d'abord dans quelques préliminaires un bref retour sur le fonctionnement supposé d'un diagnostiqueur (de notre point de vue), qui nous permettra de ce fait de considérer l'architecture du système piloté selon la « vision » qu'en aura le diagnostiqueur. Nous passerons ensuite en revue les différentes techniques possibles de modélisation d'un système, pour enfin définir l'ensemble temporel propre à l'exécution du système. Suite à ces préliminaires, nous pourrons modéliser un système piloté. Nous considérerons en premier lieu l'architecture d'un système piloté tel que nous venons de la présenter (i.e. : avec ses parties pilotage et opérante), puis nous modéliserons celle-ci que nous nommerons simplement « système simple ». Ensuite, et dans le but d'avoir une approche de diagnostic à base de modèles, nous introduirons donc l'architecture adéquate constituée du système piloté entier (i.e. : sa partie opérante et sa partie pilotage) et d'un modèle lui aussi entier à des fins de diagnostiques. Nous la modéliserons et la nommerons « système complet ».

2.2.1 Préliminaires

Ces préliminaires vont nous permettre, suite à un bref retour sur le fonctionnement du futur diagnostiqueur, de distinguer les différentes architectures possibles ainsi que les modélisations possibles d'un système piloté. Nous introduirons ensuite l'ensemble temporel propre à l'exécution du système.

2.2.1.1 Retour sur le fonctionnement du diagnostiqueur

Comme nous l'avons vu, un diagnostiqueur peut être implanté de deux manières : une implantation embarquée ou une implantation débarquée. Nous faisons le choix d'une implantation embarquée pour laquelle le diagnostiqueur est donc intégré au système. Il surveillera celui-ci en temps réel, à travers l'analyse de son comportement réellement observé, afin de déterminer s'il fonctionne normalement.

Dans le cas contraire, il devra déterminer la cause du fonctionnement anormal (le ou les défauts apparus) afin d'informer le module de décision qui choisira la suite de fonctionnement du système.

Le comportement réellement observé du système est fourni par l'évolution dans le temps des valeurs de ses variables observables. Selon la méthode de diagnostic considérée, ces variables observables désignent les mesures des capteurs ainsi que potentiellement les commandes du système de pilotage aux actionneurs et les consignes venant de l'opérateur.

La surveillance du comportement réellement observé est effectuée en vérifiant en temps réel que ce comportement valide ou invalide certaines propriétés du bon et des mauvais fonctionnements. Ces propriétés, définies suivant un formalisme spécifique établi en fonction de la méthodologie de diagnostic appliquée, doivent donc refléter le fonctionnement du système dans le cas normal et les cas anormaux (i.e. : lors de la présence de défaut(s)). Elles représentent donc les règles d'analyse du diagnostiqueur. Notre objectif étant d'apprendre ces différentes propriétés par simulation du modèle, nous avons de ce fait besoin de modèles décrivant le système en bon et mauvais fonctionnement.

2.2.1.2 Les différentes architectures considérées

La première architecture que nous considérerons, que nous nommerons le « système simple », sera la boucle composée du système de pilotage et du système opérant. Nous justifierons le fait de considérer cette boucle par le fait que les systèmes pilotés sont généralement contrôlés en boucle fermée. Néanmoins, comme nous considérons que suite au diagnostic d'un défaut, le diagnostiqueur informe le module de décision qui choisira la suite de fonctionnement du système, alors nous considérons que ce module de décision n'est pas surveillé par le diagnostiqueur, et nous supposerons que la consigne venant de l'opérateur est sans défaut. Cela signifie que nous réduisons l'architecture considérée du système de pilotage uniquement au module de régulation, qui contient les lois de commande du système opérant ainsi que les interfaces d'entrées et sorties de et vers le système opérant. Par ailleurs, en n'intégrant pas le fonctionnement du module de décision qui adopte les différents modes de fonctionnement du système, nous supposons donc que le système se trouve en mode de fonctionnement nominal. Cette architecture du système simple tel que la « verra » le diagnostiqueur est représentée par la figure 2.9 ci-dessous.

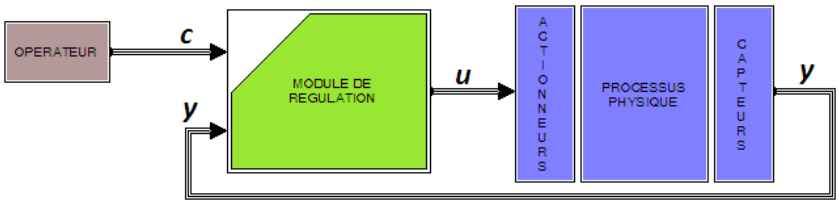

Figure 2.9 – Vision du système simple par le diagnostiqueur.

Par ailleurs, nous avons vu au chapitre 1 d'introduction au diagnostic que les méthodologies de diagnostic à base de modèles embarquent un modèle du système. En fonctionnement, ce modèle est simulé afin de comparer le fonctionnement réel du système, issu des variables observables, au comportement prédit obtenu par la simulation du modèle embarqué. De ce fait, nous pouvons définir une architecture utilisant un modèle du système. C'est ce que nous nommerons le « système complet » et qui comportera non seulement le modèle du système opérant mais aussi un modèle du système de pilotage (une duplication du module de régulation) qui commandera donc ce modèle. Cela signifie que le système et le modèle auront leurs propres systèmes de pilotage indépendants et alimentés par la même consigne venant de l'opérateur. La figure 2.10 de la page de la page 40 représente cette architecture du système complet tel que la « verra » le diagnostiqueur.

Enfin et comme indiqué dans [VRYK03], bien que les systèmes opérants soient des processus continus, un diagnostiqueur utilise pour sa part des données échantillonnées dans le temps. Pour nos

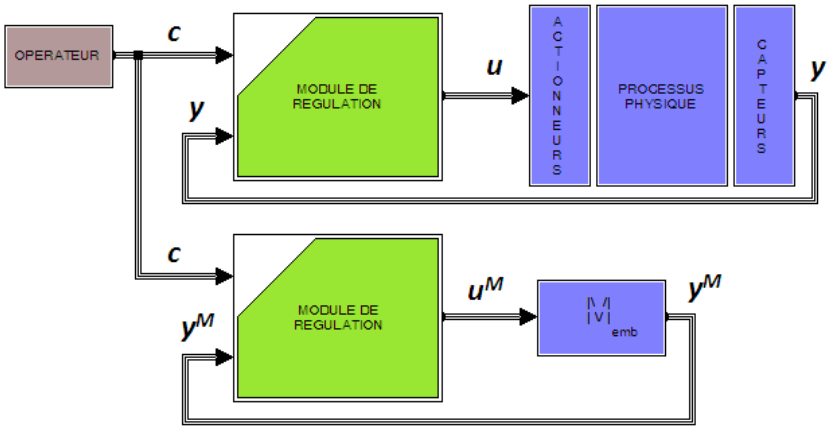

Figure 2.10 – Vision du système complet par le diagnostiqueur.

phases d'apprentissage, il est par conséquent nécessaire d'avoir des modèles discrets dans la variable temporelle de fonctionnement. Cela est d'autant plus important pour le modèle embarqué car il sera intégré dans un calculateur, qui lui ne fonctionne qu'avec un pas temporel échantillonné. Nous ferons donc l'hypothèse de toujours avoir à notre disposition des modèles évoluant dans un ensemble temporel discret et allons d'ailleurs décrire formellement, dans la suite, cet ensemble temporel spécifique à l'exécution du système.

2.2.1.3 Les différents types de modélisation de systèmes

La représentation du fonctionnement dynamique d'un système, par le biais d'un modèle mathématique, est une étape fondamentale dans de nombreuses disciplines scientifiques telles que les sciences de l'ingénieur, les sciences physiques ou encore les sciences économiques et sociales. Concernant les sciences de l'ingénieur, le modèle mathématique joue un rôle important tant dans la conception (le dimensionnement des composants), que l'étude (analyse des mécanismes inhérents au comportement du système), ainsi que la conduite (optimisation du comportement, commande et surveillance) d'un système existant ou devant être construit.

En reprenant [BJL+90], un modèle d'un système est une représentation abstraite du système, réalisée à partir des outils mathématiques à disposition et pour certaines conditions d'utilisation du système. Il s'agit donc d'une formalisation mathématique de la réalité du système qui se présente généralement sous la forme d'un ensemble de variables et de relations de contraintes (fonctions et/ou équations) liant ces variables. Les modèles peuvent être de nature et de complexité différentes.

Abstractions possibles de modélisation Selon [Der09], il y a trois abstractions possibles pour modéliser la dynamique de fonctionnement d'un système : les systèmes continus, les systèmes à événements discrets et les systèmes dynamiques hybrides. Le choix entre ces abstractions se faisant suivant l'objectif considéré.

Les systèmes continus sont constitués de différentes grandeurs physiques (vitesse, température, pression ou courant électrique par exemple) qui peuvent prendre des valeurs réelles lorsque le temps évolue. Certaines de ces grandeurs physiques peuvent être mesurées. La représentation de ces systèmes fait appel à des outils mathématiques aptes à la représentation de la dynamique continue : les équations différentielles. Les systèmes continus sont perçus par l'automaticien à travers une représentation reposant le plus souvent sur des variables d'état continues et une variable temporelle continue ou discrète (comme le montre [LR08]).

Les Systèmes à Événements Discrets (généralement notés SED) sont des systèmes constitués d'un ensemble fini d'états potentiels et dont les transitions entre ces différents états sont associées à l'oc-

currence d'événements discrets (dans le sens où le temps n'a pas nécessairement d'importance). Citons comme exemple de modèles mathématiques de tels systèmes les automates à états finis, tels que les structures de Kripke bien connues dans le domaine du model-checking (voir [CGP00] et [BBF+01]), et les Réseaux de Petri. Remarquons par ailleurs que certains modèles de SED, les automates temporisés par exemple ([AD94]), utilisent des variables temporelles de nature symbolique, où le temps sert principalement à définir une chronologie entre les événements.

Les Systèmes Dynamiques Hybrides couvrent simultanément les deux aspects continu et discret par évolution dans le temps et combinaison des variables continues et des variables discrètes (voir [Bra98] ainsi que [Gir04] pour des présentations complètes agrémentées d'exemples). Un état discret du système peut être vu comme un système continu avec des variables continues reliées par des contraintes restreignant l'état en question. La transition du système d'un état à un autre fait ainsi changer son mode de fonctionnement en lui faisant subir d'autres lois continues propres au nouvel état. De nombreux exemples sont disponibles dans [Bra98], citons entre autres : une balles rebondissante, le contrôle de la température d'une pièce ou du niveau d'un réservoir, ou encore le contrôle électronique du système de transmission d'un véhicule.

Critères principaux de choix de la modélisation adéquate Selon [BKLS03], trois critères principaux permettent de choisir la modélisation adéquate d'un système : le type des variables à considérer et leurs ensembles de valeurs, le temps d'exécution du système et les contraintes liant les différentes variables.

Concernant les variables, la première interrogation concerne les variables à considérer : c'est-à-dire celles qui ont le plus d'intérêt dans la description du fonctionnement du système. Le système opérant introduit généralement des variables de puissance et d'énergie, alors que le système de pilotage introduit quant à lui des signaux de contrôle et d'information. Ainsi les variables à considérer sont toutes les grandeurs mises en relations dans les différents composants du système. Suite au choix des variables du système à considérer, la seconde interrogation concerne le choix de l'ensemble des valeurs pouvant être assignées aux variables. Les variables dites *quantitatives* prennent leurs valeurs dans un sous-ensemble de l'ensemble $\mathbb{R}$ des nombres réels, tandis que les variables dites *qualitatives* prennent leurs valeurs dans un ensemble fini de symboles non nécessairement ordonnés.

Concernant le temps d'exécution du système, la variable le décrivant peut prendre ses valeurs dans deux types d'ensembles. L'ensemble $\mathbb{R}^+$ des nombres réels positifs permet de décrire un fonctionnement en temps continu. L'ensemble $\mathbb{N}$ des nombres entiers positifs (ou tout ensemble isomorphe à $\mathbb{N}$), permet de décrire un fonctionnement échantillonné dans le temps. Le temps d'exécution du système est dit *synchrone* tant que la période d'échantillonnage du système est fournie par une horloge, il s'agit d'un échantillonnement à pas fixe. Dans le cas contraire lorsqu'il est dirigé par les changements d'évènements du système, il est dit *asynchrone* et considéré à chaque occurrence d'évènements.

Concernant les contraintes liant les différentes variables du système, elles doivent décrire l'évolution temporelle de celles-ci dans leurs ensembles de valeurs. Elles peuvent être classées suivant ce qu'elles représentent et la forme qu'elles prennent. Au plus bas niveau du système, chaque composant est décrit par ses propres contraintes et le système global, formé par le regroupement de ses différents composants le constituant, est ainsi décrit par la concaténation de toutes les différentes contraintes. Les différentes formes de contraintes sont regroupées suivant les différents types des variables et du temps. L'évolution des variables continues (i.e. : celles évoluant dans un sous-ensemble des nombres réels) peut être décrite suivant une variable temporelle continue en utilisant les équations algébriques et différentielles ainsi que les fonctions de transfert (transformée de Laplace). Elle peut aussi être décrite suivant une variable temporelle discrète, lorsque le système de pilotage est considéré avec un échantillonnage à pas fixe, en utilisant les équations algébriques, les relations de récurrence ou encore les fonctions de transfert.

2.2.1.4 Définition et propriétés de l'ensemble du temps d'exécution du système

Avant de modéliser les systèmes simple et complet, nous introduisons au préalable la structure de l'espace temporel spécifique à l'exécution du système. En effet, nous allons décrire l'évolution du système (son fonctionnement) dans le temps, il convient donc en premier lieu de décrire formellement cet ensemble temporel. Pour la majorité des notions mathématiques, notamment les notions ensemblistes, nous ferons référence à [CL03].

L'ensemble du temps Le diagnostiqueur sera un outil implanté dans un calculateur fonctionnant dans un domaine temporel discret, c'est-à-dire échantillonné à pas fixe au moyen d'une horloge. De ce fait, l'ensemble $\mathbb{T}$ du temps spécifique à l'exécution du système est considéré discret, $\mathbb{T}$ est donc isomorphe à l'ensemble $\mathbb{N}$ des entiers naturels. Cette considération est aussi justifiée par le fait que le système de pilotage est un calculateur fonctionnant lui aussi dans un domaine temporel discret, pas nécessairement identique à celui du diagnostiqueur. Cet ensemble $\mathbb{T}$ peut être vu comme un cadencement à intervalle régulier ι, où ι représentant la période d'échantillonnage.

Définition 2.2 *Pour une période d'échantillonnage $\iota \in \mathbb{D}$ (i.e. : $\iota = a \cdot 10^{\kappa}$ où $a \in \mathbb{N}^*$ et $\kappa \in \mathbb{Z}$), l'ensemble $\mathbb{T}_\iota$ du temps est de la forme $\mathbb{T}_\iota = \{\iota \cdot n / n \in \mathbb{N}\}$.*

Pour un instant $\mathrm{t} \in \mathbb{T}_\iota$, il existe donc un entier $n \in \mathbb{N}$ tel que $\mathrm{t} = \iota \cdot n$. Par exemple pour $a = 1$ et $\kappa = -1$ alors $\iota = 0.1$ et $\mathbb{T}_{0.1} = \{0; 0.1; 0.2; \ldots; 0.9; 1; 1.1; 1.2; \ldots; 1.9; 2; \ldots\}$, pour $a = 7$ et $\kappa = 0$ alors $\iota = 7$ et $\mathbb{T}_7 = \{0; 7; 14; 21; 28; \ldots\}$.

Opérations arithmétiques sur l'ensemble du temps Les opérations arithmétiques sur $\mathbb{T}_\iota$ sont définies par :

- Addition : pour deux instants $\mathrm{t}, \mathrm{t}' \in \mathbb{T}_\iota$ de la forme $\mathrm{t} = \iota \cdot n$ et $\mathrm{t}' = \iota \cdot m$, l'addition de t et t′ est définie par $\mathrm{t} + \mathrm{t}' = \iota \cdot (n + m)$.
- Soustraction : pour deux instants $\mathrm{t}, \mathrm{t}' \in \mathbb{T}_\iota$ de la forme $\mathrm{t} = \iota \cdot n$ et $\mathrm{t}' = \iota \cdot m$, la soustraction de t et t′ est définie uniquement si $n \geq m$ par $\mathrm{t} - \mathrm{t}' = \iota \cdot (n - m)$.
- Successeur : pour un instant $\mathrm{t} \in \mathbb{T}_\iota$ de la forme $\mathrm{t} = \iota \cdot n$, le *successeur* de t dans $\mathbb{T}_\iota$ est noté t^+ et se définit par $\mathrm{t}^+ = \iota \cdot (n + 1)$.
- Prédécesseur : pour un instant $\mathrm{t} \in \mathbb{T}_\iota$ de la forme $\mathrm{t} = \iota \cdot n$, le *prédécesseur* de t dans $\mathbb{T}_\iota$ est noté t^- et se définit uniquement si $n > 0$ par $\mathrm{t}^- = \iota \cdot (n - 1)$.
- Multiplication par un scalaire : pour un instant $\mathrm{t} \in \mathbb{T}_\iota$ de la forme $\mathrm{t} = \iota \cdot n$ et un scalaire $\lambda \in \mathbb{N}^*$, la multiplication de t par λ est définie par $\lambda \cdot \mathrm{t} = (\lambda \cdot \iota) \cdot n$.

L'addition d'un instant $\mathrm{t} \in \mathbb{T}_\iota$, de la forme $\mathrm{t} = \iota \cdot n$, par un scalaire $\lambda \in \mathbb{N}^*$ est facilement obtenue par itération de la fonction successeur ; de même que la soustraction avec la fonction prédécesseur en faisant néanmoins attention que $\lambda \leq n$:

$$\mathrm{t} + \lambda = (\cdots(\mathrm{t}^{\overbrace{+}^{\lambda \text{ fois}}})\cdots)^+ = \iota \cdot (n + \lambda) \qquad \mathrm{t} - \lambda = (\cdots(\mathrm{t}^{\overbrace{-}^{\lambda \text{ fois}}})\cdots)^- = \iota \cdot (n - \lambda)$$

Relation d'ordre dans l'ensemble du temps L'ensemble $\mathbb{T}_\iota$ est muni de la relation d'ordre $\leq$ usuelle sur les nombres. Pour deux instants $\mathrm{t}, \mathrm{t}' \in \mathbb{T}_\iota$, de la forme $\mathrm{t} = \iota \cdot n$ et $\mathrm{t}' = \iota \cdot m$, alors $\mathrm{t} \leq \mathrm{t}'$ si et seulement si $n \leq m$. Il s'agit bien sûr d'un ordre total.

Distance dans l'ensemble du temps L'ensemble $\mathbb{T}_\iota$ est muni de la distance $dist_{\mathbb{T}_\iota}$ définie par :

$$dist_{\mathbb{T}_\iota} : \begin{array}{ccc} \mathbb{T}_\iota \times \mathbb{T}_\iota & \longrightarrow & \mathbb{R}_+ \\ (t, t') & \longmapsto & \begin{cases} \frac{t-t'}{\iota} & \text{si } t \geq t' \\ \frac{t'-t}{\iota} & \text{sinon} \end{cases} \end{array}$$

Il s'agit bien d'une distance car les axiomes de symétrie, séparation et inégalité triangulaire sont bien vérifiés.

Intervalles dans l'ensemble du temps Un intervalle de $\mathbb{T}_\iota$ est l'ensemble de tous les éléments de $\mathbb{T}_\iota$ compris entre deux éléments quelconques de $\mathbb{T}_\iota$. Par construction de cet ensemble $\mathbb{T}_\iota$, les seuls intervalles de $\mathbb{T}_\iota$ pouvant être considérés (or singleton du type $[t_1; t_1]$ avec $t_1 \in \mathbb{T}_\iota$) sont les fermés et bornés de la forme $[t_1; t_2] = \{t \in \mathbb{T}_\iota / t_1 \leq t \wedge t \leq t_2\}$, avec $t_1, t_2 \in \mathbb{T}_\iota$. En effet, considérer un intervalle ouvert revient à considérer l'intervalle fermé avec le successeur ou le prédécesseur de la borne où il est ouvert :

- un intervalle du type $]t_1; t_2[$, avec $t_1, t_2 \in \mathbb{T}_\iota$, s'écrit simplement sous la forme $[{t_1}^+; {t_2}^-]$;
- un intervalle du type $]t_1; t_2]$, avec $t_1, t_2 \in \mathbb{T}_\iota$, s'écrit simplement sous la forme $[{t_1}^+; t_2]$;
- un intervalle du type $[t_1; t_2[$, avec $t_1, t_2 \in \mathbb{T}_\iota$, s'écrit simplement sous la forme $[t_1; {t_2}^-]$;

Pour un intervalle $I \subseteq \mathbb{T}_\iota$ de la forme $I = [t_1; t_2]$, sa longueur est $long(I) = \frac{t_2 - t_1}{\iota}$ (i.e. : la distance entre ses deux bornes) et son cardinal est $card(I) = \frac{t_2 - t_1}{\iota} + 1$.

2.2.2 Modélisation du système piloté simple

Nous allons modéliser le système piloté simple : c'est-à-dire constitué uniquement du système opérant et de son système de pilotage. Bien que d'autres modélisations peuvent être utilisées, nous utiliserons une modélisation par espace d'état. Nous avons privilégié ce choix de modélisation pour sa simplicité à nous permettre, par la suite, d'intégrer les défauts potentiels du système : intégrer directement les équations des défauts dans les équations du modèle de bon fonctionnement du système.

2.2.2.1 Modélisation du système opérant

Un système opérant *SO* est constitué de trois parties : la partie actionneurs, la partie processus physique et la partie capteurs. La figure 2.11 ci-dessous, incluant les perturbations externes potentielles, représente cette architecture. Nous avons vu que les outils adéquats pour sa modélisation sont les équations algébriques et différentielles décrivant son évolution dans le temps.

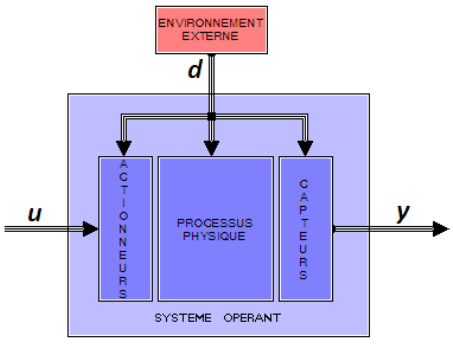

Figure 2.11 – Architecture du système opérant.

La modélisation d'un tel système opérant *SO* est classiquement donnée par espace d'état suivant une représentation continue en temps discret $\mathbb{T}_\iota$ ([LR08]). Les équations le décrivant sont données par l'ensemble 2.1 suivant :

$$SO : \begin{cases} x(\mathrm{t}^+) & = & f(x(\mathrm{t}), \theta, u(\mathrm{t}), d(\mathrm{t})) \\ y(\mathrm{t}) & = & g(x(\mathrm{t})) \\ x(0) & = & x_{init} \end{cases} \tag{2.1}$$

où $x = (x_1, \ldots, x_n) \in X$, $\theta = (\theta_1, \ldots, \theta_m) \in \mathbb{R}^m$, $u = (u_1, \ldots, u_p) \in U$, $d = (d_1, \ldots, d_t) \in D$ et $y = (y_1, \ldots, y_q) \in Y$ représentent respectivement les vecteurs des états, des paramètres, des commandes fournies par le système de pilotage, des perturbations venant de l'environnement extérieur et des mesures du système. Les fonctions f et g sont supposées au moins continues par morceaux.

2.2.2.2 Intégration du système de pilotage dans la modélisation

Dans la majeure partie des travaux de la littérature ([BJL+90], [PM01], [VRYK03], [Ise06] ou encore [BKLS03]), les méthodes mises au point pour le diagnostic se basent sur une représentation du système en boucle ouverte. Or, dans la plupart des applications industrielles, le système est inséré dans une boucle de régulation ou de commande, piloté par un contrôleur afin d'accroître ses performances et de les maintenir en dépit des entrées inconnues pouvant l'affecter.

Le diagnostic des défauts est dans ce contexte plus délicat du fait des objectifs contradictoires entre la commande et le diagnostic. En effet, l'objectif de la commande est de minimiser, voire d'annuler, les effets des perturbations et des défauts ; alors que l'objectif du diagnostic est, quant à lui, justement de mettre en évidence ces défauts. La solution envisagée est de considérer la boucle complète du système : c'est-à-dire le système opérant bouclé avec son système de pilotage et qui forment la *boucle réelle*.

La modélisation d'un tel système S passe ainsi par la modélisation du système opérant SO, que nous venons de présenter, ainsi que par la modélisation du système de pilotage SP : c'est-à-dire, car nous avons réduit ce système de pilotage au module de régulation, l'élaboration des lois de commande ainsi que les différentes interfaces d'entrées et sorties de et vers le système opérant. Nous supposons donc, pour un système donné, avoir un modèle de bon fonctionnement du système opérant SO ainsi que son module de régulation (i.e. : ses lois de commande et interfaces) comme le représente la figure 2.12 ci-dessous.

Le modèle de bon fonctionnement du système opérant SO est nommé *modèle parfait* et noté M_{parf}. La « perfection » de ce modèle M_{parf}, et donc l'emploi du terme *parfait*, signifie que ce modèle représente le plus fidèlement possible le système opérant SO réel. C'est-à-dire qu'il intègre tous les multiples phénomènes physiques évoluant dans le système réel, y compris les non-linéarités qui sont des contraintes pour une utilisation en embarqué. Il pourra de ce fait avoir une représentation « fidèle » du système physique réel et la seule limite potentielle de fidélité de modélisation reposera dans l'outil utilisé pour la modélisation et la simulation.

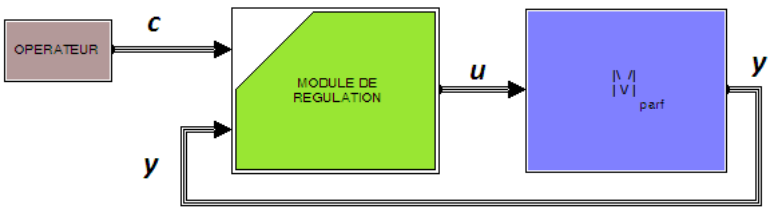

Figure 2.12 – Architecture de modélisation du système simple.

Nous considérons une modélisation continue en temps discret $\mathbb{T}_\iota$, avec une représentation par espace d'état, représentée par l'ensemble 2.2 suivant d'équations :

$$S : \left\{ \begin{array}{l} SO : \left\{ \begin{array}{lcl} x(\mathrm{t}^+) & = & f(x(\mathrm{t}), \theta, u(\mathrm{t}), d(\mathrm{t})) \\ y(\mathrm{t}) & = & g(x(\mathrm{t})) \\ x(0) & = & x_{init} \end{array} \right. \\ SP : \left\{ \begin{array}{lcl} a(\mathrm{t}^+) & = & h(c(\mathrm{t}), a(\mathrm{t}), y(\mathrm{t})) \\ u(\mathrm{t}) & = & k(a(\mathrm{t})) \\ a(0) & = & a_{init} \end{array} \right. \end{array} \right. \tag{2.2}$$

où $a = (a_1, \ldots, a_s) \in A$ et $c = (c_1, \ldots, c_r) \in C$ représentent respectivement les vecteurs des états et des consignes (venant de l'opérateur) du système de pilotage. Les fonctions h et k sont supposées au moins continues par morceaux.

2.2.2.3 Remarques concernant l'émission et l'acquisition des signaux

Un point important qu'il est nécessaire d'aborder dès à présent et qui impacte la pertinence du modèle parfait vis-à-vis du système réel, concerne l'émission et l'acquisition des différents signaux du système.

En premier lieu, signalons comme nous l'avons déjà indiqué que nous supposons que le procédé de régulation du système opérant se fait de manière numérique. Ainsi, et bien que nous aurions pu être plus rigoureux dans la description des équations du modèle, en décrivant le système opérant avec une représentation en temps continu ainsi que le système de pilotage avec cette représentation en temps discret et un passage entre les deux se faisant par représentation des convertisseurs analogique-numérique et numérique-analogique, nous avons préféré garder cette représentation globale en temps discret comme cela se fait de manière usuelle dans la littérature.

En second lieu, les différents signaux mesurés du système sont généralement de différentes natures : électriques, analogiques ou numériques. Concernant les signaux électriques ou analogiques, ils sont par ailleurs généralement fortement bruités. Ils est par conséquent nécessaire d'en effectuer un pré-traitement pour les rendre utilisables par l'algorithme de commande du système (les lois de commande du système de pilotage). Les outils propres au traitement du signal, les filtres bien connus et maitrisés par la communauté de l'automatique, sont des solutions adéquates permettant ce pré-traitement des signaux mesurés bruités. Dans notre cas, toute cette phase d'acquisition des différents signaux est supposée intégrée dans la modélisation du système S. En effet lorsque nous disons que le diagnostiqueur a accès aux variables mesurées, nous supposons implicitement qu'il y a accès au niveau du système de pilotage et que celles-ci ont ainsi, au préalable, été pré-traitées au niveau de ce système de pilotage.

2.2.3 Modélisation du système piloté complet

2.2.3.1 Rajout d'informations grâce à un modèle

Les méthodologies de diagnostic à base de modèles, que nous avons présentées au chapitre 1 précédent, consistent à comparer le fonctionnement réellement observé du système à son fonctionnement prédit ceci afin d'en détecter des divergences. Ce fonctionnement prédit est obtenu par simulation d'un modèle embarqué de bon fonctionnement du système.

Il faut donc disposer d'un modèle « embarqué » M_{emb} de bon fonctionnement du système. Ici encore, nous supposons disposer de ce modèle embarqué M_{emb}. Nous obtenons donc le système réel, alimenté par la consigne venant de l'opérateur, ainsi que son modèle alimenté par la même commande venant du système de pilotage du système réel.

2.2.3.2 Modélisation du modèle embarqué

Comme dans le cas du système simple S, le modèle embarqué M_{emb} est représenté par l'ensemble 2.3 suivant d'équations :

$$\mathrm{M}_{emb} : \begin{cases} x^M(\mathrm{t}^+) &= f^M(x^M(\mathrm{t}), \theta^M, u(\mathrm{t}), d^M(\mathrm{t})) \\ y^M(\mathrm{t}) &= g^M(x^M(\mathrm{t})) \\ x^M(0) &= x^M_{init} \end{cases} \tag{2.3}$$

où $x^M = (x^M_1, \ldots, x^M_n) \in X^M$, $\theta^M = (\theta^M_1, \ldots, \theta^M_m) \in \mathbb{R}^m$, $d^M = (d^M_1, \ldots, d^M_t) \in D^M$ et $y^M = (y^M_1, \ldots, y^M_q) \in Y^M$ représentent respectivement les vecteurs des états, des paramètres, des perturbations et des mesures du modèle embarqué M_{emb}. Les fonctions f^M et g^M sont supposées au moins continues par morceaux.

2.2.3.3 Rajout d'un système de pilotage pour le modèle embarqué

Au vu de l'architecture considérée, le système bouclé avec sa commande, il faut donc disposer d'un modèle de l'ensemble complet. Pour la partie opérante, nous disposons du modèle « embarqué » M_{emb} de bon fonctionnement du système. Pour la partie commande, une simple duplication du module de régulation permet de l'obtenir. Remarquons néanmoins que ce module de régulation peut être allégé des différentes interfaces d'entrées et de sorties, et ainsi être uniquement constitué des lois de commande. Comme le représente la figure 2.13 ci-dessous, nous obtenons donc deux boucles de fonctionnement alimentées toutes les deux par la même consigne venant de l'opérateur.

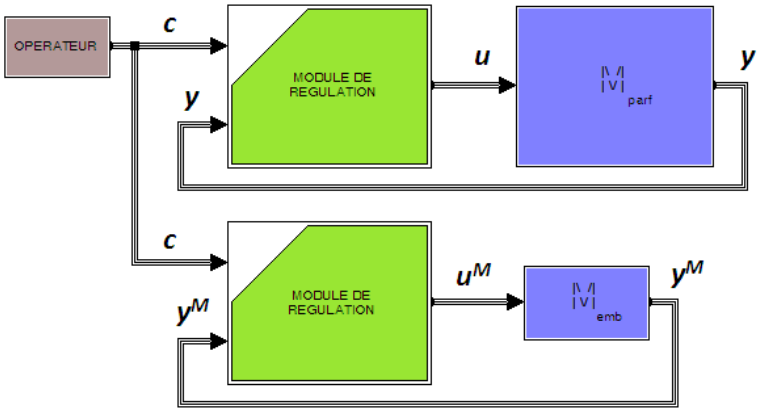

Figure 2.13 – Architecture de modélisation du système complet.

2.2.3.4 Modélisation du système complet

Comme dans le cas où nous ne considérions que le système simple S, le système complet $\overline{S}$ est représenté par l'ensemble 2.4 suivant d'équations :

$$\overline{S}: \begin{cases} S: & \begin{cases} x(\mathrm{t}^+) = f(x(\mathrm{t}), \theta, u(\mathrm{t}), d(\mathrm{t})) \\ y(\mathrm{t}) = g(x(\mathrm{t})) \\ a(\mathrm{t}^+) = h(a(\mathrm{t})) \\ u(\mathrm{t}) = k(c(\mathrm{t}), a(\mathrm{t}), y(\mathrm{t})) \\ x(0) = x_{init} \\ a(0) = a_{init} \end{cases} \\ S^M: & \begin{cases} x^M(\mathrm{t}^+) = f^M(x^M(\mathrm{t}), \theta^M, u^M(\mathrm{t}), d^M(\mathrm{t})) \\ y^M(\mathrm{t}) = g^M(x^M(\mathrm{t})) \\ a^M(\mathrm{t}^+) = h(c(\mathrm{t}), a^M(\mathrm{t}), y^M(\mathrm{t})) \\ u^M(\mathrm{t}) = k(a^M(\mathrm{t})) \\ x^M(0) = x^M_{init} \\ a^M(0) = a^M_{init} \end{cases} \end{cases} \tag{2.4}$$

où S représente le modèle parfait M_{parf} avec sa partie commande, suivant les mêmes hypothèses que précédemment sur les vecteurs et fonctions, et S^M représente le modèle embarqué M_{emb} avec sa partie commande dupliquée. $a^M = (a_1^M, \ldots, a_s^M) \in A^M$ représente le vecteur des états du système de pilotage dupliqué.

Remarquons que, autant pour le modèle parfait que pour le modèle embarqué, nous ne considérons pas de potentielles dérives temporelles des vecteurs θ et θ^M des paramètres.

2.2.3.5 Remarques et notations

Un point important à signaler dès à présent concerne la dynamique de réponse du système. Rappelons qu'il s'agit du temps que met le système pour atteindre n'importe quelle consigne donnée par l'opérateur. Or en considérant deux boucles, nous introduisons automatiquement deux dynamiques, la dynamique β_{parf} liée au modèle parfait et la dynamique β_{emb} liée au modèle embarqué, qui peuvent très bien ne pas être égales. Comme le plus important est le fonctionnement du système réel, donné par le modèle parfait, nous supposerons donc que la dynamique β du système complet est la dynamique β_{parf} liée au modèle parfait.

Pour la suite du document, nous posons d'une part $\overline{\mathrm{V}} = \{c; a; u; x; y; d; a^M; u^M; x^M; y^M; d^M\}$ l'ensemble des variables du système ainsi que $\overline{\mathrm{VD}}$ le produit de tous les domaines des variables du système complet $\overline{S}$ (i.e. : $\overline{\mathrm{VD}} = C \times A \times U \times X \times Y \times D \times A^M \times U^M \times X^M \times Y^M \times D^M$). D'autre part $\overline{\mathrm{V}}_{obs} = \{c; u; y; u^M; y^M\}$ désigne l'ensemble des variables observables du système et $\overline{\mathrm{VD}}_{obs}$ désigne le produit de tous les domaines des variables observables du système complet $\overline{S}$ (i.e. : $\overline{\mathrm{VD}}_{obs} = C \times U \times Y \times U^M \times Y^M$).

2.3 Le cas d'étude

Le but du document est de présenter une démarche de conception d'un outil de diagnostic appliqué aux systèmes technologiques pilotés. Nous allons de ce fait appliquer, tout au long du document, toutes les notions et démarches introduites sur un cas d'étude spécifique. Ce cas d'étude a été choisi dans l'objectif qu'il soit d'une part bien connu et maîtrisé et d'autre part qu'il représente un exemple concret de systèmes industriels.

Ce cas d'étude est un générateur électrique à pile à combustible que nous nommerons simplement par la suite un *système pile à combustible*.

2.3.1 Une pile à combustible

Une pile à combustible est une pile où la génération de l'électricité se fait grâce à l'oxydation sur une électrode d'un combustible réducteur (de l'hydrogène dans notre cas d'étude) couplée à la réduction sur l'autre électrode d'un oxydant (l'oxygène de l'air toujours dans notre cas d'étude). Sans rentrer dans les détails de fonctionnement, nous laissons le soin au lecteur intéressé de consulter [SNCH$^+$00] pour une présentation complète et détaillée du principe de fonctionnement ainsi que des différentes technologies de piles à combustible.

Différentes familles de piles à combustible ont été développées pour différents champs d'application allant de l'embarqué (engins spatiaux, navires, sous-marins, bus, automobiles, appareils nomades, etc.) au stationnaire (usines de fabrications, hôpitaux, bâtiments de bureaux, maisons individuelles, etc.). Le niveau de maturation des piles est maintenant bon, par suite des efforts de plusieurs grands groupes industriels et de constructeurs automobiles, avec l'avantage d'avoir une bonne compacité, de bonnes perspectives de réduction de coût et des durées de vie suffisantes (40 000 heures environ).

2.3.2 Le système pile à combustible étudié

Le projet FISYPAC ([RGPC09]) avait pour objectifs de réaliser et de fiabiliser un système pile à combustible d'une puissance nominale de 17 kW, puis de l'intégrer et de le tester à bord d'un véhicule électrique. Le véhicule conçu est du type « range extender » permettant de corriger le problème d'autonomie inhérent aux véhicules électriques (équipés généralement d'une batterie de puissance) en y intégrant une génératrice d'énergie électrique : le système pile à combustible. La puissance du véhicule provient donc de l'hybridation du système pile à combustible avec une batterie de puissance. Le véhicule servant de support au projet fut une 307CC, initialement thermique, transformée en véhicule électrique dont les performances ciblées étaient :

- vitesse maximale de 130 km/h ;
- autonomie minimale de 400 km ;
- consommation d'hydrogène proche de 1kg d'hydrogène au 100 km.

Le rôle du système pile à combustible est de fournir de l'énergie électrique à un convertisseur électrique haute tension pour entraîner une machine électrique de traction ou pour recharger une batterie Li-ion. Le système est autonome dans la gestion de ses alimentations en carburant, en comburant et en fluide caloporteur pour assurer les consignes de demande de puissance électrique provenant du superviseur véhicule.

La pile à combustible fut initialement conçue lors du projet GENEPAC ([AGRPC07]) pour une utilisation dans l'automobile, en tenant compte de plusieurs objectifs tels que la taille réduite, la modularité, l'efficacité et la compatibilité avec les contraintes de production en grand nombre et d'intégration dans un véhicule ([PCR06]. Il s'agit d'une technologie de pile à combustible à membrane polymère (PEMFC pour *Proton Exchange Membrane Fuel Cell*) permettant le transfert des protons de l'anode vers la cathode par combustion de l'hydrogène dans l'oxygène de l'air. Elle produit donc de l'électricité, de l'eau et de la chaleur. Pour un fonctionnement correct, les réactions chimiques de production doivent être effectuées à une température, une pression et avec un taux d'humidité appropriés.

2.3.2.1 Architecture du système pile à combustible étudié

Le système pile à combustible est composé de la pile en elle-même, de ses deux circuits d'alimentation en oxygène et en hydrogène, et de son circuit de refroidissement permettant d'assurer l'homogénéité de la température dans la pile. Les deux principaux sous-systèmes sont donc ce circuit de refroidissement et les deux circuits d'alimentation qui ont des fonctions disjointes : refroidissement et fourniture de la puissance. Une autre justification de la séparation du circuit de refroidissement est le fait que la gestion thermique peut être une gestion globale véhicule du et non pas dédiée aux seuls

composants du système pile (ce qui est le cas dans le projet FISYPAC). Une description complète de l'architecture de ce système pile à combustible est fournie dans [RGPC09] et [BDF+10]. L'architecture complète du système pile à combustible étudié est représentée par la figure 2.14 suivante de la page 49.

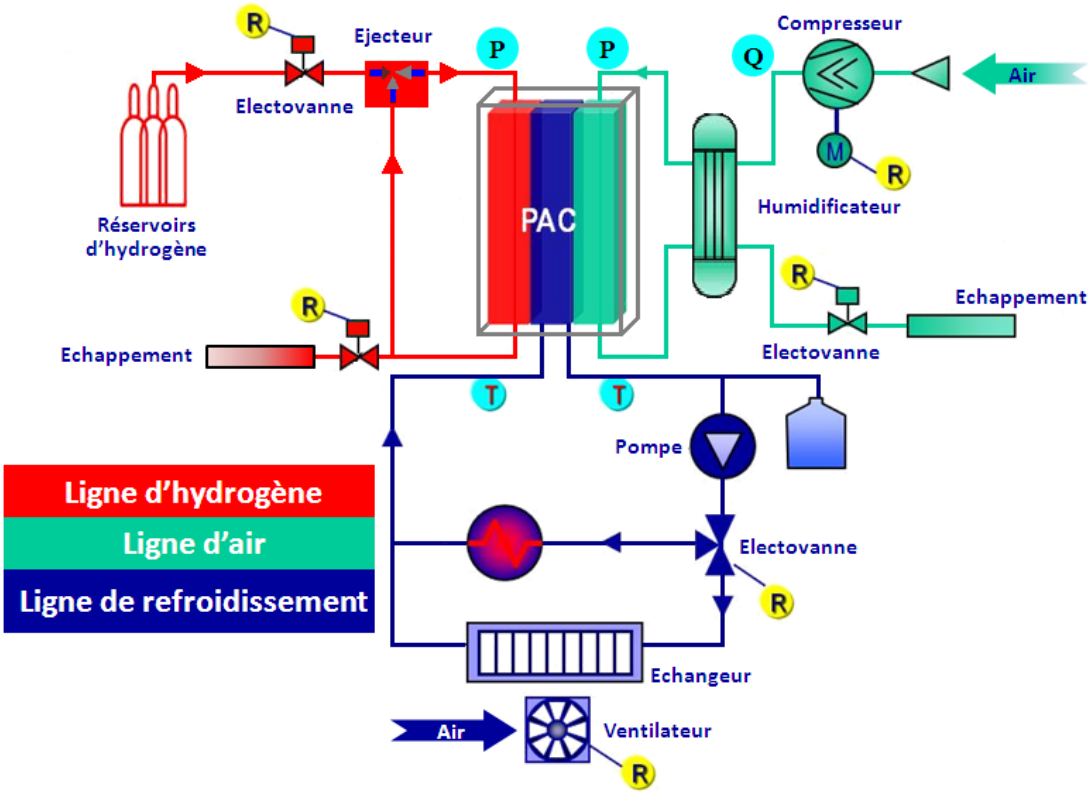

Figure 2.14 – Architecture du système pile à combustible.

La pile à combustible est constituée de deux modules, chacun composés de 120 cellules électriques, montés en série et liés entre eux par une culasse de distribution des fluides assurant l'alimentation en air et en hydrogène ainsi que le refroidissement des modules.

La ligne d'alimentation en air fournit la pile en oxygène. Un compresseur contrôle le débit d'air nécessaire et une électrovanne contrôle la pression de l'air dans la ligne ainsi que son rejet à l'échappement. Le taux d'humidité est régulé par un composant physique spécifique non contrôlé (qui n'est donc pas un actionneur) : un humidificateur qui fonctionne comme une éponge. Deux capteurs sont présents sur cette ligne : un capteur de débit situé en sortie du compresseur et un capteur de pression situé en entrée de la pile.

La ligne d'alimentation en hydrogène fournit la pile en hydrogène. Cet hydrogène est stocké sous haute pression (700 bars) dans des réservoirs *ad-hoc* intégrés au véhicule. Une électrovanne contrôle l'admission de l'hydrogène dans la ligne à la pression appropriée. Un éjecteur, dimensionné spécifiquement pour assurer la stœchiométrie ainsi que le taux d'humidité appropriés de l'hydrogène, est utilisé pour permettre la re-circulation de l'hydrogène dans la ligne. Une électrovanne contrôle le rejet d'hydrogène à l'échappement. Un capteur de pression est situé sur la ligne en entrée de la pile.

La ligne de refroidissement permet, pour un fonctionnement correct, d'assurer l'homogénéité de la température dans le cœur de la pile. Le liquide caloporteur circule à travers cette ligne grâce à une pompe de re-circulation. Une électrovanne à trois voies permet d'orienter le liquide vers un échangeur thermique, refroidi par un ventilateur afin de le rafraîchir, ou vers un radiateur de chauffe lors des démarrages à froid car la pile nécessite une certaine température interne afin de pouvoir se mettre en

fonctionnement. Deux capteurs de température sont situés sur cette ligne : un en entrée de la pile et l'autre en sortie.

2.3.2.2 Le système de pilotage du système pile à combustible

Le système de pilotage du système pile à combustible (complètement décrit dans [BDF+10]) fut conçu et développé avec une approche à base de modèles. Cette approche utilise une commande prédictive à base de modèles qui simule un modèle interne du système pour prédire le comportement futur du système afin de calculer les commandes appropriées. La conception de ce système de pilotage fut réalisée en utilisant un modèle complet du système pile à combustible (en plus du modèle interne) afin de simuler ses comportements statique et dynamique. Sa structure reproduit la décomposition hiérarchique du système en le considérant comme un dual : un sous-système de pilotage pour chacune des lignes (air, hydrogène et refroidissement) et un système de pilotage global pour le système pile à combustible global.

Le système de pilotage global a pour rôle de réguler les différentes lignes pour permettre à la pile de fournir la puissance électrique demandée par le superviseur véhicule. Suivant donc une consigne de puissance électrique provenant du superviseur véhicule, le système de pilotage global calcule ainsi les commandes adéquates de débit d'air, de pression d'air et d'hydrogène pour répondre à cette demande, tout ceci en assurant la température du cœur de la pile. Le débit d'air est calculé suivant cette puissance électrique demandée ainsi qu'en accord avec l'exigence de stœchiométrie d'air (limitée à 1,5). La pression d'air est déduite de ce débit d'air en accord avec les exigences de pression dans la pile (entre 1,3 et 1,5 bar). La pression d'hydrogène est calculée en suivant directement la mesure de la pression d'air et en accord avec l'exigence d'intégrité de la pile : la différence de pression entre l'anode et la cathode doit être inférieure à 300 millibars.

Le système de pilotage de la ligne d'air calcule les commandes du compresseur et de l'électrovanne suivant les consignes de débit et pression d'air reçues du système de pilotage global. Le système de pilotage de la ligne d'hydrogène calcule la commande de l'électrovanne d'admission d'hydrogène suivant le besoin en hydrogène. Enfin, le système de pilotage du circuit de refroidissement assure principalement la gestion thermique du système pile avec en supplément le préchauffage et le post-refroidissement.

Il y a une rétro-action entre tous les niveaux des systèmes de pilotage : le superviseur véhicule, le système de pilotage global et les systèmes de pilotage des deux lignes d'alimentation (le système de pilotage du circuit de refroidissement ne rentre que peu en compte dans cette procédure, sauf lors d'une température anormale du cœur de la pile où le système complet est mis à l'arrêt). Quand le superviseur véhicule demande une puissance électrique au système de pilotage global du système pile à combustible, celui-ci calcule le débit d'air et les pressions d'air et d'hydrogène nécessaires puis en fait la demande aux systèmes de pilotage des deux lignes d'alimentation. Ensuite, celles-ci régulent leur propre ligne et informent le système de pilotage global du débit et des pressions qui sont mesurés. Celui-ci estime alors la puissance électrique produite et en informe le superviseur véhicule.

2.3.2.3 Le modèle du système pile à combustible

Le modèle du système pile à combustible fut conçu suivant différents travaux issus de la littérature. [MAH+00], [FG03] et [LLA98] pour les modèles électrochimiques décrivant les relations entre l'intensité et la tension des cellules. [PPS02], [WNS05], [PSP02] et [GPH+03] pour les modèles dynamiques de systèmes pile à combustible prenant en compte tous les principaux phénomènes thermodynamiques, hydrauliques et électriques. Ce modèle fut validé avec le système réel lors des tests expérimentaux.

Il est développé sous l'environnement de simulation MATLAB/Simulink© en utilisant une approche à base de composants provenant des outils PhiGraph© et PhiSim© ([BFY05]). PhiGraph©, adapté du concept Bond-Graph, est un outil permettant de générer des modèles ou des librairies de composants de systèmes physiques en utilisant un environnement de schémas-blocs. PhiSim© est un outil étendant

PhiGraph© par l'ajout du concept multi-port : un unique lien entre différents composants permet de représenter l'ensemble des relations d'interaction entre ces composants.

Ce modèle du système pile à combustible représente les comportements statique et dynamique du système et est composé, comme le système réel, des quatre parties : la pile, les deux lignes d'alimentation et la ligne de refroidissement. Il est constitué de l'assemblage de différents composants élémentaires (compresseur, électrovannes, humidificateur, tuyauterie, etc.) utilisés comme pièces de base du modèle suivant les phénomènes physiques à représenter (débit, pression, taux d'humidité, stœchiometrie, température, tension et intensité électrique, etc.).

La figure 2.15 suivante montre une comparaison entre les données du système réel et celles du modèle, réalisée durant des tests expérimentaux. Le premier graphique concerne la puissance électrique : en rouge est représentée celle demandée par le superviseur du véhicule au système de pilotage global, et en bleu est représentée celle mesurée par les capteurs du système réel. Les autres graphiques montrent respectivement les valeurs de la température de la pile, la pression et le débit d'air : en bleu sont représentées celles mesurées par les capteurs du système réel et en rouge sont représentées celles venant du modèle.

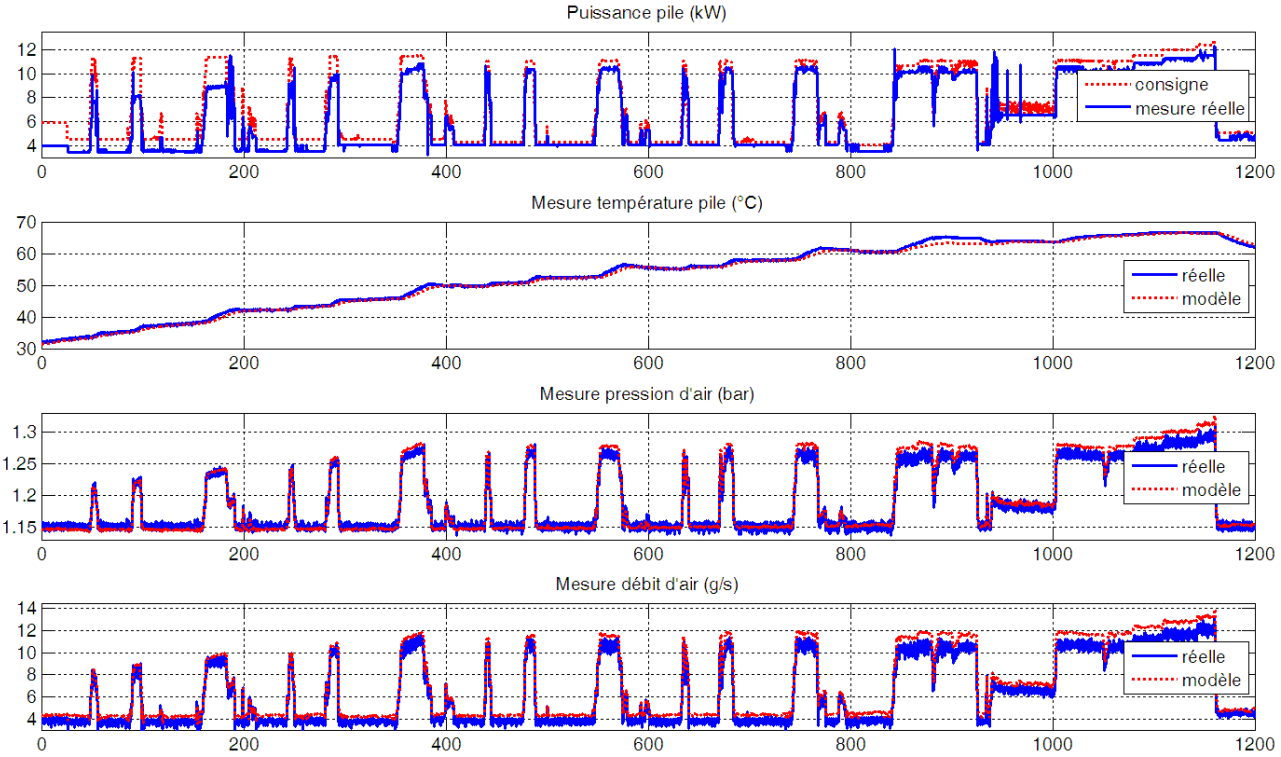

Figure 2.15 – Comparaison entre les données du système réel et celles du modèle.

2.3.3 La ligne d'air du système pile à combustible

Pour la suite, nous allons nous intéresser à une sous-partie du système pile à combustible : la ligne d'alimentation en air que nous nommerons simplement la *ligne d'air*. Cette sous-partie présente l'avantage de ne pas être trop complexe et, bien qu'étant intégrée dans un système piloté, est elle-même un système piloté à part entière composé de sa partie opérante et de sa partie pilotage. En effet, comme nous l'avons précédemment indiqué, la structure du système de pilotage du système pile à combustible reproduit la décomposition hiérarchique du système avec un système de pilotage global pour le système pile à combustible global et un sous-système de pilotage pour chacune des lignes. La

ligne d'air possède donc son propre système de pilotage recevant sa consigne du système de pilotage global.

Cette ligne d'air a le rôle de fournir la pile en air (dont l'oxygène en est consommé) à un débit et une pression ciblés tout en respectant les contraintes de taux d'humidité et de stœchiométrie. Un compresseur contrôle le débit d'air nécessaire et une électrovanne contrôle la pression de l'air dans la ligne. Le taux d'humidité est régulé par un humidificateur qui est, comme nous l'avons indiqué, un organe « passif » fonctionnant comme une éponge. L'architecture complète de la partie opérante de cette ligne d'air est présentée par la figure 2.16 ci-dessous. Les différents composants intervenant dans cette structure sont décrits dans le tableau 2.1 suivant, en prenant une décomposition hiérarchique par actionneurs, composants physiques et capteurs.

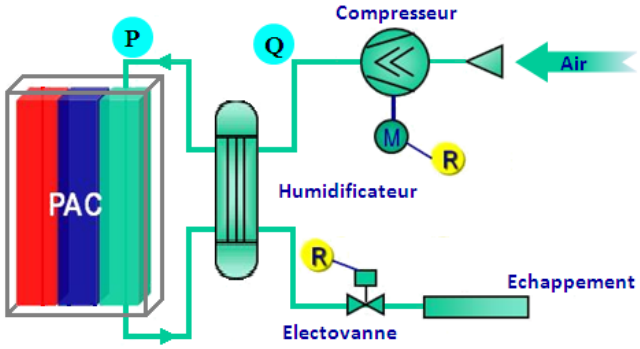

Figure 2.16 – Architecture de la ligne d'air du système pile à combustible.

Composants		Fonctions
Actionneurs	Compresseur	Alimente en air la ligne avec un débit cible.
	Électrovanne	Régule la pression dans la ligne et permet d'évacuer l'air consommé.
Composants physiques	Humidificateur	Régule le taux d'humidité de l'air fourni à la pile.
	Pile	
	Raccord d'admission	Raccord entre l'extérieur et l'entrée du compresseur.
	Raccord compresseur-humidificateur	Tuyau de raccordement entre le compresseur et l'humidificateur.
	Raccords humidificateur-pile	Tuyaux de raccordement entre l'humidificateur et la pile.
	Raccord humidificateur-électrovanne	Tuyaux de raccordement entre l'humidificateur et l'électrovanne.
	Raccord d'échappement	Raccord entre l'électrovanne et l'extérieur.
Capteurs	Débit	Situé en sortie du compresseur.
	Pression	Situé en entrée de la pile.

Tableau 2.1 – Composants de la partie opérante de la ligne d'air.

2.3.3.1 Le modèle parfait de la ligne d'air

Comme nous l'avons signalé, un modèle de simulation du système pile à combustible, donc de la ligne d'air, a été développé sous l'environnement de simulation MATLAB/Simulink©. Il correspond au modèle « parfait » nécessaire au diagnostic suivant l'architecture du système complet que nous avons précédemment présentée. Ce modèle est fabriqué par l'assemblage de composants élémentaires (compresseur, électrovannes, humidificateur, tuyauterie, etc.) utilisés comme pièces de base du modèle

suivant les phénomènes physiques à représenter. Dans le cas de la ligne d'air, il s'agit du débit, de la pression, du taux d'humidité, et de la température de l'air.

La figure 2.17 suivante de la page 53 représente la partie opérante du modèle de simulation de la ligne d'air. Il s'agit d'un modèle du type « thermo-fluide » pour lequel nous retrouvons la décomposition du système en blocs élémentaires ainsi que le concept de représentation Bond-Graph. Outre les blocs *compresseur*, *électrovanne* et *humidificateur* clairement identifiés, les autres blocs élémentaires sont :

- deux blocs *source de fluide* représentant l'admission et l'échappement d'air avec leurs raccords respectifs.
- cinq blocs *volume de fluide* : quatre représentant les différentes tuyauteries entre les composants et un spécifique représentant la pile.
- deux blocs *résistance hydraulique* permettant d'obtenir les pertes de charge en amont et aval de la pile.

Remarquons que du point de vue thermo-fluide, la pile n'est représentée que par un volume (un bloc *volume de fluide*) avec consommation de matière (blocs *résistance hydraulique*).

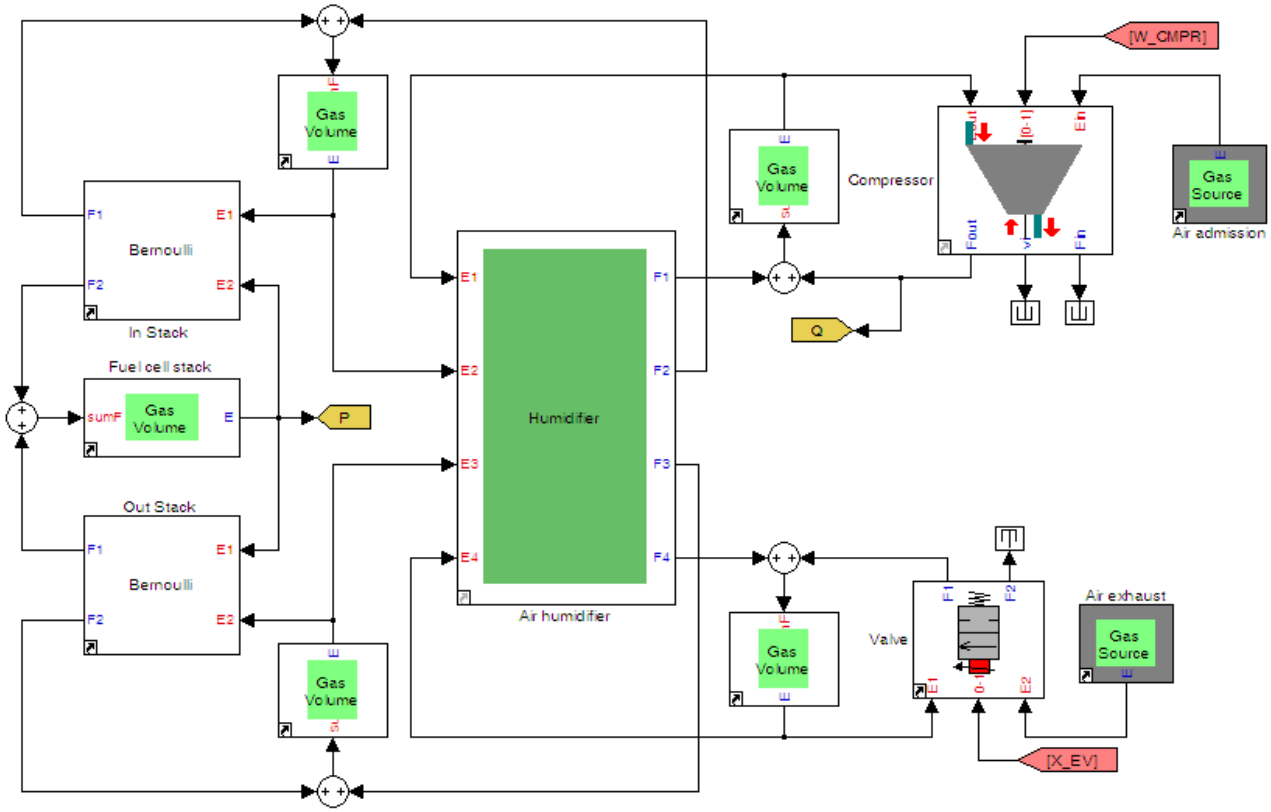

Figure 2.17 – Partie opérante du modèle de simulation de la ligne d'air.

Les différents blocs du modèle de la ligne d'air Nous n'allons pas donner les équations de ce modèle car d'une part cela serait trop volumineux, et cela n'aurait d'autre part que peu d'intérêt pour la suite du document dans le sens où, sauf pour y intégrer les défauts potentiels, nous allons utiliser ce modèle comme une boîte noire. Nous allons tout de même présenter les principales caractéristiques de fonctionnement des différents blocs.

Le bloc *compresseur* représente une transformation (compression) de la pression d'air (exprimée en bar) entre l'entrée et la sortie du compresseur et ainsi une augmentation ou diminution du débit d'air (exprimé en gramme par seconde) en sortie du compresseur. Son fonctionnement est régi par une

équation non-linéaire où le débit de sortie est exprimé en fonction de la pression de sortie et du régime (exprimé en rotations par seconde) du compresseur.

Le bloc *électrovanne* représente une modification de la quantité d'air envoyé vers l'extérieur. Son fonctionnement est représenté par l'ouverture d'un orifice modélisé par une perte de charge régie par les équations de Barré Saint-Venant.

Le bloc *humidificateur* représente l'humidification d'un flux d'air sec à partir d'un flux d'air humide. La modélisation est basée sur la méthode NUT, pour *Nombre d'Unité de transfert*. Il s'agit d'un nombre défini par le rapport entre la conductance thermique, représentant à la fois la taille et la conductivité de l'échangeur, et la capacité de transport de chaleur minimale. Cette modélisation prend aussi en compte les pertes de charge, le transfert d'humidité et le transfert thermique.

Le bloc *source de fluide* (noté *Gas Source*) représente une source d'air donnée par un mélange des gaz d'oxygène (O_2), d'azote (N_2) et de vapeur d'eau (H_2O) à une pression (exprimée en bar) et température (exprimée en degré Celsius) fixées. Les compositions massiques de chacun des gaz (exprimées en pourcentage dans une mole du mélange des trois gaz), la pression ainsi que la température du fluide sont fixées comme paramètres du modèle. Le bloc fournit donc un fluide suivant une certaine pression, température et composition.

Le bloc *volume de fluide* (noté *Gas Volume*) représente un contenant de transit du fluide entre deux composants (une portion de tuyauterie ou la pile en elle-même) dont le fonctionnement est régi par l'équation des gaz parfaits. En fonction du paramètre de volume (en litre) du contenant, le bloc exprime donc l'évolution des variables de débit, pression et température dans le tuyau.

Le bloc *résistance hydraulique* (noté *Bernoulli*) permet d'obtenir une perte de charge dans un volume. Son fonctionnement est régi par l'équation de Bernoulli : le débit massique est donné par une fonction non-linéaire prenant en compte la section du volume considéré, la masse volumique de chacun des gaz et la différence entre la pression en entrée et en sortie du volume.

2.3.3.2 Le système de pilotage de la ligne d'air

Le système de pilotage de la ligne d'air doit contrôler le débit et la pression d'air requis par le système de pilotage global du système pile à combustible. Comme nous l'avons vu, ce débit d'air est calculé suivant d'une part la demande de puissance électrique requise par le superviseur du véhicule et d'autre part en accord avec l'exigence de stœchiometrie. La pression d'air est ensuite déduite de ce débit et en accord avec l'exigence de pression maximale autorisée dans la pile et la ligne. Pour ce faire, le système de pilotage de la ligne d'air calcule la vitesse de rotation du compresseur et l'angle d'ouverture de l'électrovanne suivant ces consignes de débit et pression d'air.

Comme nous l'avons dit, le système de pilotage de la ligne d'air utilise une approche de commande prédictive à base de modèles. Un modèle interne simplifié a donc été développé et intégré à cette partie pilotage. Par regroupement, la ligne d'air a été réduite à un seul volume (la pile) et deux pertes de charge : les pertes de charge en amont de la pile sont regroupées avec le compresseur, celles en aval sont regroupées avec l'électrovanne. Ceci revient donc à supposer que d'une part le débit en entrée de la pile est égal au débit de sortie du compresseur et d'autre part que le débit en sortie de la pile est égal au débit de la vanne de contre pression.

2.3.3.3 Le modèle embarqué de la ligne d'air

La ligne d'air que nous venons de présenter est un système piloté simple. Afin de le considérer comme un système piloté complet, nous devons obtenir un modèle embarqué de cette ligne. Nous dupliquerons ensuite les lois de commande pour former la boucle modèle.

Au paragraphe précédent, nous venons d'indiquer que le système de pilotage utilise une approche de commande prédictive à base de modèles et qu'un modèle interne simplifié a donc été développé et intégré à cette partie pilotage. Il s'agit du modèle que nous allons considérer pour le cas d'étude. Nous

allons donc dupliquer ce modèle interne de la commande ainsi que dupliquer cette commande afin de créer les deux boucles, dont l'architecture est représentée par la figure 2.18 suivante de la page 55 :

- une boucle réelle composée du modèle parfait M_{parf} de la ligne d'air bouclée avec les lois de commande, basées sur une approche prédictive à base de modèles et utilisant un modèle interne M_{emb} ;
- une boucle modèle composée d'une duplication du modèle interne M_{emb} utilisée pour la commande ainsi qu'une duplication des lois de commande utilisant elles aussi un modèle interne M_{emb}.

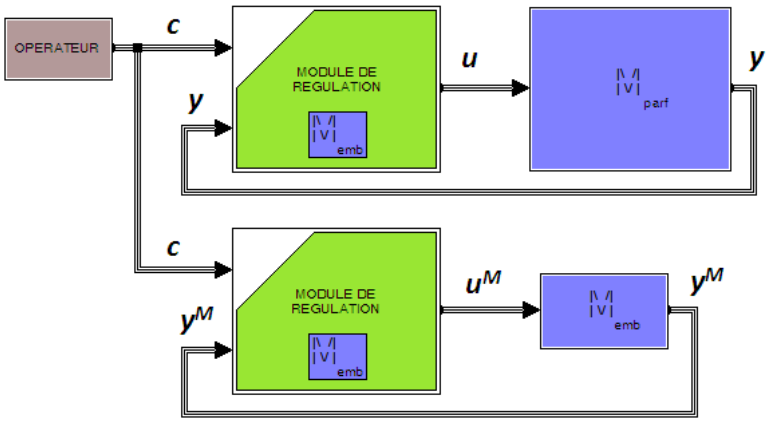

Figure 2.18 – Architecture de la ligne d'air complète.

2.3.3.4 Ensemble des variables observables de la ligne d'air complète

Les variables « observables » (terme utilisé dans le sens de variables mesurées) de la boucle réelle sont les mesures y_Q de débit et y_P de pression, ainsi que les commandes u_ω du compresseur et u_x de l'électrovanne. Les variables observables de la boucle modèle sont les mesures y_Q^M de débit et y_P^M de pression, ainsi que les commandes u_ω^M du compresseur et u_x^M de l'électrovanne. En ajoutant les consignes c_Q de débit et c_P de pression, nous obtenons l'ensemble $\overline{\mathrm{V}}_{obs}$ des variables observables de cette ligne d'air complète :

$$\overline{\mathrm{V}}_{obs} = \{c_Q; c_P; u_\omega; u_x; y_Q; y_P; u_\omega^M; u_x^M; y_Q^M; y_P^M\}$$

Le produit $\overline{\mathrm{VD}}_{obs}$ de tous les domaines des variables observables de cette ligne d'air complète est, pour sa part, donné par :

$$\overline{\mathrm{VD}}_{obs} = C_Q \times C_P \times U_\omega \times U_x \times Y_Q \times Y_P \times U_\omega^M \times U_x^M \times Y_Q^M \times Y_P^M$$

2.3.3.5 Simulation de la ligne d'air

Lors de sa conception, la période d'échantillonnage ι de cette ligne d'air fut fixée à 0.01 seconde ; l'ensemble du temps d'exécution de la ligne d'air est donc $\mathbb{T}_{0.01} = \{0; 0.01; 0.02; \ldots; 0.09; 0.1; 0.11; \ldots\}$.

La figure 2.19 suivante de la page 56 représente une simulation de cette ligne d'air suivant un profil de consigne de débit donné et durant une fenêtre temporelle de longueur 50 secondes. La consigne de débit de la ligne d'air suit la consigne de puissance électrique requise par le superviseur véhicule, qui est liée directement à l'action du conducteur sur la pédale d'accélération du véhicule. Ce profil de consigne reflète donc une fonction aléatoire comme lors d'une utilisation urbaine où le conducteur accélère et décélère en permanence avec des niveaux et des périodes plus ou moins élevés.

Le premier graphique représente l'évolution du débit d'air dans la ligne. Le trait discontinu rouge représente le débit requis par le contrôleur global du système pile à combustible (c'est-à-dire la consigne de débit). Le trait continu bleu représente la mesure du débit du modèle parfait (représentant le système réel) et le trait continu vert représente la mesure du débit du modèle embarqué.

Le deuxième graphique représente l'évolution de la pression d'air dans la ligne. Le trait discontinu rouge représente la consigne de pression requise, le trait continu bleu représente la mesure de la pression du modèle parfait et le trait continu vert représente celle du modèle embarqué. Remarquons que cette consigne de pression est corrélée à la consigne de débit d'air car comme nous l'avons signalé, elle est déduite de ce débit en accord avec les exigences de pression dans la pile.

Les troisième et dernier graphiques représentent les évolutions des commandes des actionneurs (le compresseur pour le troisième graphique et l'électrovanne pour le dernier) fournies par le contrôleur de la ligne d'air. Les traits bleus concernent les commandes fournies au modèle parfait et les traits verts celles fournies au modèle embarqué.

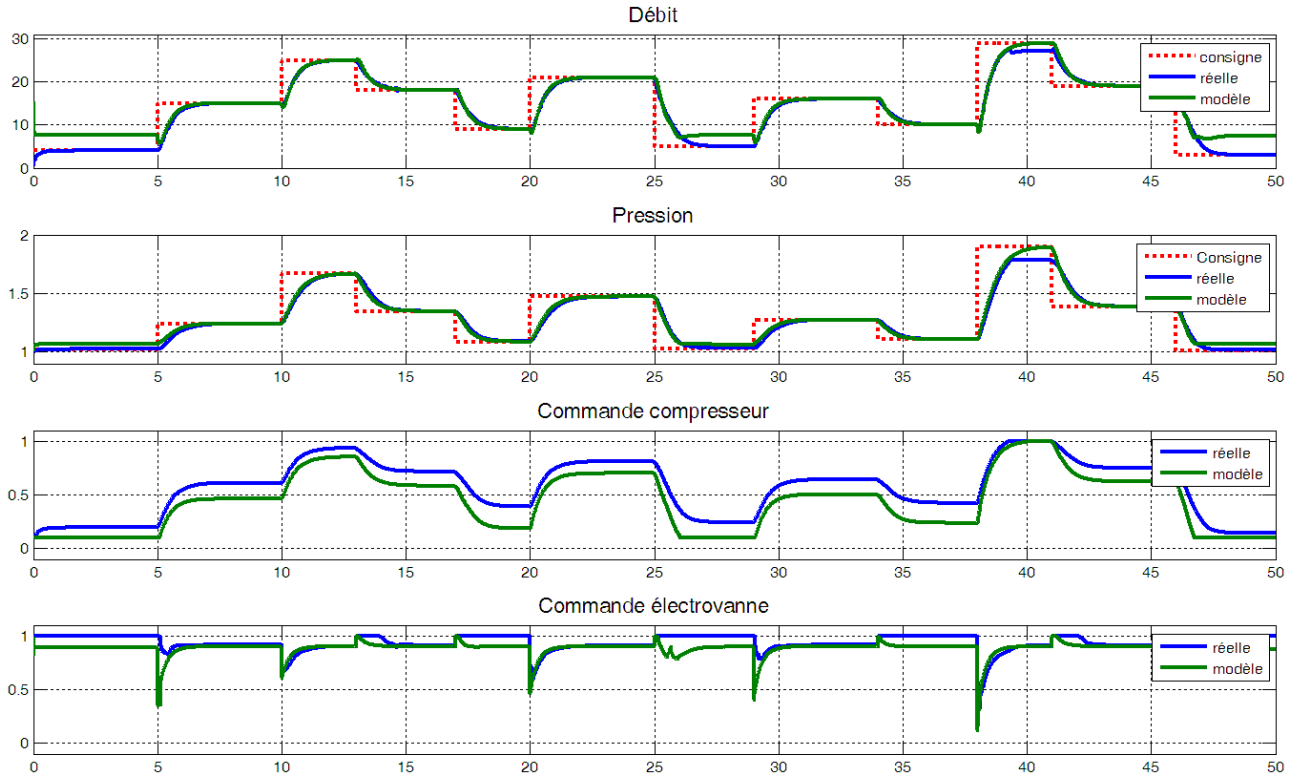

Figure 2.19 – Simulation de la ligne d'air.

2.4 Conclusion sur les systèmes technologiques pilotés

Nous venons, dans ce chapitre, de présenter les systèmes technologiques pilotés pour lesquels nous souhaitons y intégrer un diagnostiqueur. Nous les avons introduits en les considérant comme des systèmes mécatroniques pour lesquels nous distinguons particulièrement la partie pilotage de la partie opérante.

Nous avons ensuite introduit le formalisme permettant de les modéliser, en accord avec l'utilisation d'outils de simulation. Nous avons de ce fait considéré une modélisation continue par espace d'état en temps discret. Ceci se justifie car la modélisation continue s'adapte bien à la représentation de la partie opérante et le temps discret est une condition nécessaire pour une utilisation en simulation, tant pour l'étude de la diagnosticabilité que pour le diagnostiqueur.

Enfin, nous avons présenté un exemple applicatif de système technologique piloté qui sera utilisé dans la suite de ce document. Il s'agit de la ligne d'air d'une pile à combustible qui a l'avantage de ne pas être trop complexe pour pouvoir facilement mettre en application les concepts présentés par la suite.

Suite à ce chapitre, il nous faut maintenant déterminer les défauts potentiels du système, que nous pourrons intégrer dans le modèle que nous venons de décrire. Nous pourrons ensuite en étudier la diagnosticabilité avant de générer le diagnostiqueur associé à cette étude de diagnosticabilité.

Chapitre 3

La typologie des défauts

La modélisation pertinente d'un défaut potentiel du système, c'est-à-dire son intégration au plus juste dans le modèle de bon fonctionnement, est une phase importante pour un fonctionnement parfait du futur diagnostiqueur ([Ise06]). Elle nécessite de bien comprendre le lien entre le défaut physique réel et ses effets sur le système réel ; lien qui ne peut être réellement fourni que par un examen du système réel ainsi qu'une compréhension des lois physiques et une analyse des symptômes du défaut. Plusieurs raisons peuvent mener à des défauts d'un système, découlant entre autres exemples :
- d'une mauvaise conception ;
- d'une mauvaise fabrication ou d'un mauvais assemblage ;
- d'un mauvais fonctionnement ;
- d'une absence ou d'un oubli de maintenance ;
- d'un vieillissement, d'une corrosion ou d'une usure durant l'exploitation du système.

Nous observons bien que des défauts peuvent être causés par de multiples facteurs et qu'ainsi leur étude préalable est nécessaire, si ce n'est obligatoire, pour un bon fonctionnement du futur diagnostiqueur. Par conséquent et avant d'étudier la diagnosticabilité des défauts potentiels d'un système piloté, il convient au préalable de les inventorier puis de les caractériser pour permettre ensuite de les intégrer au plus juste dans le modèle de bon fonctionnement du système. Ceci nous permettra par la suite d'une part d'étudier l'impact de ces défauts répertoriés sur le fonctionnement du système (i.e. : l'étude de la diagnosticabilité), puis d'autre part de générer le diagnostiqueur associé à cette étude de diagnosticabilité.

Ce chapitre va donc définir une typologie de défauts des systèmes pilotés qui doit permettre de caractériser tous les défauts potentiels d'un tel système. Nous ferons d'abord une étape préliminaire permettant un bref retour sur les notions liées aux défauts, sur l'intégration du diagnostic dans le cycle de vie d'un système ainsi que sur la différence entre défauts logiciels et défauts matériels. Nous identifierons ensuite les défauts potentiels d'un système piloté au travers d'une étude de sûreté de fonctionnement, puis nous déterminerons les principales caractéristiques des défauts qui nous permettront de les intégrer dans le modèle de bon fonctionnement du système. Nous obtiendrons alors les modèles de défauts. Nous présenterons une implémentation en MATLAB/Simulink© de cette typologie qui va nous permettre d'intégrer les défauts directement dans des modèles de simulation de systèmes. Nous mettrons enfin en œuvre toutes ces notions abordées sur le cas d'étude.

3.1 Préliminaires

3.1.1 Notions de défaut, dysfonctionnement et panne

Malgré l'abondante littérature traitant du diagnostic de défauts, très peu de documents s'attachent à définir formellement les notions de défaut, dysfonctionnement et panne. Un effort de définition est accompli dans [Ise06] qui reprend les différents travaux effectués dans [Omd88], [IFI83] et [IB97] pour

tenter de définir ces notions de manière standard. Nous donnons ici ces notions.

Définition 3.1 *Un défaut* (*ou faute ou défaillance*) *est une dérive non-permise d'au moins une propriété caractéristique du système par rapport aux conditions standard et acceptables de fonctionnement du système.*

Un défaut est un état anormal de fonctionnement du système pouvant causer une réduction, voire une perte de la capacité de l'unité fonctionnelle à exécuter sa fonction requise. Un défaut est indépendant du fait que le système soit opérationnel ou non et peut très bien ne pas affecter le fonctionnement normal du système. Un défaut peut initier un dysfonctionnement ou une panne du système.

Définition 3.2 *Un dysfonctionnement est une irrégularité intermittente dans la réalisation d'une fonction désirée du système.*

Un dysfonctionnement est donc une interruption temporaire de la fonction du système. Il s'agit d'un évènement résultant d'un ou plusieurs défauts.

Définition 3.3 *Une panne est une interruption permanente de la capacité du système à exécuter une fonction requise sous des conditions opérationnelles spécifiées.*

Comme pour un dysfonctionnement, une panne est un évènement résultant d'un ou plusieurs défauts. Différents types de pannes peuvent être distingués suivant leurs nombres (panne simple ou pannes multiples) et leurs prévisions (panne aléatoire donc non prévisible, panne déterministe donc prévisible sous certaines conditions, panne systématique ou causale dépendant de conditions connues).

3.1.2 Les différents niveaux de modélisation de défauts

En automatique, des modèles dits « de bon fonctionnement » sont classiquement utilisés ; nous avons d'ailleurs vu au chapitre 2 précédent, présentant les systèmes technologiques pilotés, comment les obtenir. Ils caractérisent le comportement normal du système : c'est-à-dire lorsqu'aucun défaut n'est présent. En surveillance, il est par contre nécessaire de compléter le modèle afin de caractériser le comportement en défaut du système. L'enrichissement du modèle de bon fonctionnement, qui permet de produire les modèles de défauts, peut être élaboré suivant trois niveaux de connaissance ([Coc04]) :

- Le niveau 1 est le niveau de connaissance le plus élémentaire qui consiste à indiquer les équations décrivant le (ou les) composant(s) qui est (sont) directement affecté(s) par le défaut : les équations du modèle qui ne seront probablement plus valides en cas de défaut.
- Le niveau 2 de connaissance consiste à décrire, grâce à des variables supplémentaires appelées variables de défaut, comment sont modifiées les équations de fonctionnement normal du système lorsqu'un défaut survient. Les défauts peuvent être additifs ou multiplicatifs suivant la manière dont les variables de défaut influencent les équations du modèle.
- Le niveau 3 de connaissance consiste à modéliser l'évolution dynamique du défaut. Des équations supplémentaires liant les variables de défaut sont ajoutées au modèle de bon fonctionnement. L'obtention de ce type de modèles nécessite néanmoins soit une connaissance fine des phénomènes physiques, soit des données expérimentales du processus défectueux.

3.1.3 Le diagnostic dans le cycle de vie d'un système

Le cycle de vie d'un système est marqué par plusieurs étapes dans lesquelles nous allons intégrer les problématiques de diagnostic : c'est-à-dire quand peuvent apparaître les défauts dans ce cycle de vie ainsi que comment y relier le diagnostic.

Selon [AFI09] et comme le représente la figure 3.1 de la page 61, le cycle de vie d'un système est décomposé en huit grandes phases classées chronologiquement dans le temps :

- la spécification du système (i.e. : la définition du cahier des charges),
- la conception du système,
- la réalisation des différents constituants,
- l'intégration du système (i.e. : l'assemblage entre eux des différents constituants pour former le système),
- la calibration, la vérification et la validation du système,
- le transfert vers l'exploitation du système,
- l'exploitation et le maintien en condition opérationnelle du système,
- le retrait de service du système.

Figure 3.1 – Phases du cycle de vie d'un système.

Remarquons que durant la phase d'exploitation, le système connaît différentes situations de vie. Il peut à un moment soit être en utilisation normale ou dégradée, soit être hors utilisation (en stockage, en transport ou encore en maintenance par exemple). Le profil de vie d'une automobile, par exemple, comporte différentes situations de vie durant la phase d'exploitation : en stationnement, en utilisation, en transport, en maintenance (maintenance préventive, périodique ou dépannage), etc.

3.1.3.1 Les défauts dans le cycle de vie du système

Plusieurs raisons peuvent provoquer des défauts d'un système. En analysant le cycle de vie d'un système, nous pouvons observer que certaines phases sont propices à l'introduction de défauts :

- Lors de la phase de conception, le non-respect d'exigences du cahier des charges du système peut avoir pour conséquence un mauvais fonctionnement nominal voire même un impact sur l'intégrité du système ou son environnement. Citons par exemple des erreurs logicielles (non respect du seuil maximal de contrôle de la pression d'air dans l'élaboration des lois de commande) ou des erreurs de dimensionnement de composants (définition de la section d'une tuyauterie trop petite par rapport au seuil minimal de fonctionnement).
- Lors de la phase de réalisation du système, l'utilisation de composants défectueux ou un mauvais assemblage du système peut aussi avoir pour conséquence un mauvais fonctionnement nominal ainsi qu'un impact potentiel sur l'intégrité du système ou son environnement. Citons par exemple lors de la fabrication de la ligne d'hydrogène, l'utilisation d'une vanne ayant son clapet de fermeture mal ébavuré engendrant ainsi une fuite interne (composant défectueux) ou une mauvaise soudure de tuyaux (mauvais assemblage) pouvant provoquer une fuite d'hydrogène occasionnant une explosion (nous reviendrons plus loin dans ce chapitre sur cette conséquence catastrophique).
- Lors de la phase d'exploitation du système, des défauts peuvent être dus à de mauvaises utilisations du système, à des maintenances non réalisées, ou encore à des usures liées au fonctionnement

normal du système (i.e. : vieillissement ou corrosion du système).

Remarquons que suivant la phase considérée, les défauts ne seront pas identiques. Les erreurs de conception ou de réalisation provoquent des défauts dits « systématiques » dans le sens où ils se produiront inévitablement (lorsque bien sûr les conditions permettant leurs occurrences seront réunies). À l'inverse des défauts d'exploitation (ruptures ou usures de composants matériels dues à de mauvaises utilisations ou à des maintenances non réalisées ou encore au vieillissement normal) qui eux provoquent des défauts dits « aléatoires » dans le sens où l'occurrence du défaut ne peut être prédite.

3.1.3.2 Intégration de problématiques de diagnostic dans le cycle de vie du système

Les problématiques de diagnostic apparaissent à plusieurs étapes du cycle de vie du système. Durant l'étape de spécification du système et au niveau des exigences non-fonctionnelles, il faut définir les pannes ou mauvais fonctionnements potentiels du système afin d'en faire une liste typologique qui servira lors de la phase de conception. Pour chacun des défauts de cette liste, il doit y avoir un niveau de criticité du défaut, qui permettra de le classer dans les défauts à diagnostiquer soit par un diagnostiqueur embarqué, soit par un diagnostiqueur débarqué. Il faut de plus définir la méthodologie de diagnostic, c'est-à-dire les méthodes et moyens de détection et d'identification des défauts. Le suivi de ces différents défauts ainsi que leurs modes de fonctionnement dégradés associés sont aussi à définir lors de cette étape. Nous approfondirons cela dans la partie suivante d'identification des défauts avec la présentation de la sûreté de fonctionnement.

L'étape de conception du système incorpore aussi la conception des diagnostiqueurs (i.e. : le diagnsotiqueur embarqué et le débarqué) suivant la méthodologie de diagnostic choisie à l'étape de spécification. Après conception et modélisation du système, que ce soit le système opérant mais aussi le système de pilotage avec notamment les lois de commande, il faut modéliser les différents défauts répertoriés puis en étudier leur diagnosticabilité. Nous verrons au chapitre 4 d'étude de la diagnosticabilité comment réaliser une telle étude suivant ce qu'il est possible d'« observer » du système. Si un défaut n'est pas diagnosticable, il faut alors revenir à la phase de conception du système pour rajouter ou ajuster les « observateurs » (i.e. : placement de nouveaux capteurs par exemple). La conception des diagnostiqueurs se réalise après l'étude de la diagnosticabilité des défauts et suivant les résultats de cette étude. Le diagnostiqueur embarqué sera couplé à la partie contrôle commande du système et le diagnostiqueur débarqué sera conçu pour se connecter au système lors des phases de maintenance et réparation. Enfin il faut concevoir la procédure de suivi des défauts qui permettra d'ajuster les diagnostiqueurs, lors de la phase d'exploitation du système, pour de nouveaux défauts non-répertoriés lors de la phase de conceptualisation, soit par oubli, soit par évolution du système et donc des défauts potentiels.

Durant l'étape d'intégration, vérification et validation, la vérification va examiner et tester le système. C'est-à-dire que non seulement le système sera testé (i.e. : vérification de son fonctionnement, c'est ce qui se passe habituellement), mais les diagnostiqueurs vont, eux aussi, être vérifiés par simulations de défauts lors des tests.

L'étape d'exploitation et de maintien en condition opérationnelle va appliquer toutes les procédures définies à l'étape de conception pour les diagnostiqueurs. Le diagnostiqueur embarqué va fonctionner en temps réel, suivant la stratégie élaborée, afin de surveiller que le comportement du système, obtenu des différentes mesures, est normal. Dans le cas contraire, il va chercher à quel comportement de défaut correspond ce comportement anormal, c'est-à-dire qu'il va rechercher le ou les défauts apparus. Le système passera alors en modes dégradés associés aux défauts diagnostiqués. Lors des phases de maintenance ou de réparation, les procédures de diagnostic débarqué sont appliquées, les défauts qui ne peuvent être traités lors de ces phases sont renvoyés au service de conception pour étude, via des fiches de défauts par exemple.

3.1.4 Défauts logiciels et défauts matériels

Du fait de la variété des composants constituant le système piloté, ainsi que leurs interactions permanentes les uns avec les autres, les défauts y apparaissant peuvent être de différentes natures. Le système de pilotage, de nature numérique, sera donc affecté par des défauts logiciels, liés à des erreurs de spécification, de conception ou de programmation ; à l'inverse du système opérant, de nature mécanique, qui sera affecté par des défauts matériels, tels des usures ou ruptures d'organes par exemple. Ainsi, la manière de définir, de modéliser et d'étudier les différents défauts du système ne va donc pas être identique dans les deux cas. Pour les défauts logiciels, les méthodologies de validation et vérification seront adéquates, alors que ce sont les méthodologies de diagnostic qui vont l'être pour les défauts matériels. Nous avons donc fait un choix sur la nature des défauts qui vont être traités : nous ne nous intéressons qu'aux défauts affectant le système opérant. Nous faisons ainsi l'hypothèse que le système de pilotage est sans défaut.

Nous venons de signaler que dans le cas de défauts logiciels, les méthodologies de validation et de vérification sont adéquates pour traiter ces défauts. Les méthodes de validation permettent de s'assurer que le logiciel réalise les fonctions attendues, et celles de vérification permettent de s'assurer que le logiciel fonctionne correctement. Citons par exemple le test statique consistant à l'examen de code, de spécifications ou de documents de conception (le cahier des charges), le test dynamique consistant à exécuter le code pour s'assurer de son fonctionnement correct, ou encore la vérification formelle grâce aux méthodes de preuve formelle ou de model-checking d'un modèle formel.

3.2 Identification des défauts

Comme nous l'avons signalé, il convient avant toute intégration et étude des défauts potentiels, de déterminer quels seront les défauts à prendre en compte par une solution de diagnostic : c'est-à-dire identifier les défauts devant être diagnostiqués par un diagnostiqueur. L'approche adéquate permettant cette identification est l'étude de sûreté de fonctionnement d'un système, qui est définie dans [VCL92] ou [Vil97] comme la science des défauts incluant leur connaissance, leur évaluation, leur prévention, leur mesure et leur maîtrise.

Dans cette partie, nous allons brièvement présenter la démarche d'étude de sûreté de fonctionnement. Il ne s'agit en aucun cas de faire un état de l'art de la sûreté de fonctionnement qui est hors de propos de ce document. Néanmoins, nous allons voir qu'une étude de sûreté de fonctionnement est la « porte d'entrée » du diagnostic car c'est à partir d'elle que sont identifiés les défauts potentiels à traiter par un diagnostiqueur. Les thématiques de diagnostic sont d'ailleurs fortement en relation avec les thématiques de sûreté de fonctionnement. Nous nous référerons par conséquent uniquement aux références suivantes : [LAB+96] et [MGT+04], ainsi qu'à [Ise06] définissant par ailleurs beaucoup de notions utiles en sûreté de fonctionnement. Nous pouvons aussi nous tourner vers [Zwi99] ou [Zwi09] pour une introduction des concepts fondamentaux, ou vers [Sta09] pour une présentation beaucoup plus complète et approfondie. Enfin nous garderons l'usage du terme « défaillance » à la place de « défaut », comme cela est habituellement fait dans ce domaine de la sûreté de fonctionnement.

3.2.1 La sûreté de fonctionnement

Une étude de sûreté de fonctionnement permet de garantir la fourniture des performances fonctionnelles d'un système au moment voulu, dans les conditions données, pendant la durée de vie prévue et sans dommage pour lui-même et son environnement. Elle regroupe les activités d'évaluation de la Fiabilité, la Maintenabilité, la Disponibilité et la Sécurité du système, dont l'acronyme est FMDS qui est l'équivalent anglais de RAMS pour *Reliability*, *Availability*, *Maintainability* and *Safety*. Rappelons ces définitions.

– La *fiabilité* est l'aptitude d'un système à accomplir une fonction requise, dans des conditions données et pendant une durée temporelle donnée.
– La *maintenabilité* est l'aptitude d'un système à être maintenu ou rétabli dans un état dans lequel il peut accomplir une fonction requise, lorsque la maintenance est effectuée dans des conditions données avec des procédures et des moyens prescrits.
– La *disponibilité* est l'aptitude d'un système à être en état d'accomplir une fonction requise dans des conditions données et à un instant temporel donné.
– La *sécurité* est l'aptitude d'un système à éviter de faire apparaître, dans des conditions données, des évènements critiques.

La démarche d'étude de sûreté de fonctionnement met en œuvre les méthodes adaptées pour prévoir les défaillances, les hiérarchiser en fonction de leur gravité et de leur probabilité d'apparition, détecter leurs causes, définir leurs effets et proposer des solutions techniques permettant de les supprimer ou au moins d'en diminuer la gravité. En partant d'une analyse fonctionnelle d'un système, les principales méthodes permettant de mener une étude de sûreté de fonctionnement, classées chronologiquement et représentées dans la figure 3.2 ci-dessous, sont l'analyse préliminaire des risques (dont l'acronyme est l'APR), l'analyse des modes de défaillance, de leurs effets et de leur criticité (dont l'acronyme est l'AMDEC) et l'étude des arbres de défaillances (que nous n'aborderons pas dans ce document). Remarquons que, bien que l'analyse fonctionnelle ne fasse pas partie intégrante d'une étude de sûreté de fonctionnement, elle y est néanmoins nécessaire, c'est pourquoi nous la présentons ici.

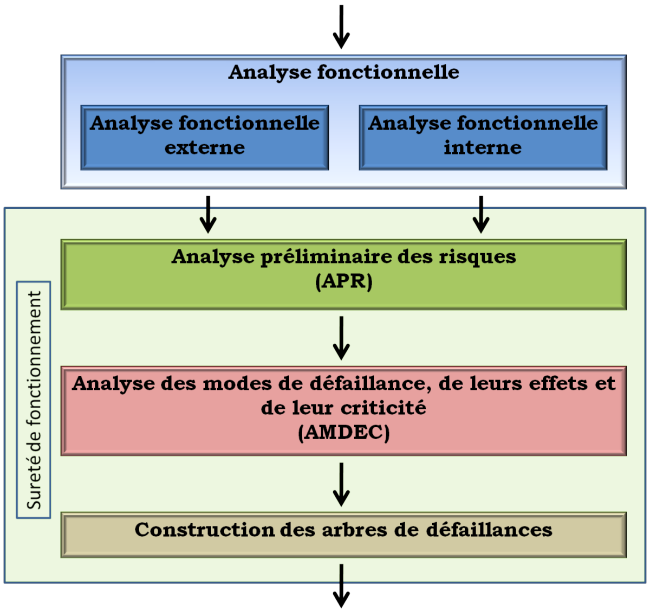

Figure 3.2 – Chronologie des principales méthodes d'une étude de la sûreté de fonctionnement.

3.2.1.1 Analyse fonctionnelle

L'analyse fonctionnelle définit le système avec ses différentes fonctions ainsi que l'ensemble des milieux extérieurs avec lesquels il est en relation. Cette analyse est un préalable à l'étude de sûreté de fonctionnement et se décompose en deux sous-parties : l'analyse fonctionnelle externe et l'analyse fonctionnelle interne.

– L'analyse fonctionnelle externe définit le système et son principe de fonctionnement, ses situations de vie ainsi que ses différents milieux extérieurs associés. Elle a pour objectif de permettre la rédaction du cahier des charges fonctionnel du système avec pour principe de le considérer

comme une boîte noire en faisant abstraction, si possible, de solutions techniques.
– L'analyse fonctionnelle interne décrit le système par les éléments qui le constituent et le fonctionnement interne entre ses différents composants. Il s'agit de la première description technique du système qui fait suite à l'analyse fonctionnelle externe. Le système n'est donc plus traité comme une boîte noire et les éléments et liens entre éléments qui le composent sont identifiés et décrits. Cette identification doit permettre de retrouver l'ensemble des fonctions données dans l'analyse fonctionnelle externe, ceci afin de pouvoir les décomposer.

3.2.1.2 Analyse préliminaire des risques

L'analyse préliminaire des risques, dont l'acronyme est l'APR, permet d'identifier les évènements redoutés liés à un système. Un événement redouté est la conséquence d'une défaillance du système qui provoque par exemple et selon les cas : une gêne, un risque avéré pour le matériel ou les personnes, un danger pour l'environnement, un risque financier, une perte d'image, etc. . L'identification se fait par construction de scénarios menant à ces événements redoutés. Ces scénarios peuvent avoir comme origines des défaillances des fonctions du système, des agressions potentielles de l'environnement sur le système, ou encore des menaces du système sur l'environnement. Nous parlons alors dans le premier cas d'analyse préliminaire des risques fonctionnels, dans le second cas d'analyse préliminaire des risques d'agression et dans le troisième cas d'analyse préliminaire des risques de menace. Dans le cas des risques fonctionnels par exemple, les modes de défaillance d'une fonction peuvent être déclinés en :
– perte : la fonction était requise et assurée, puis cesse d'être assurée à un moment donné alors qu'elle est toujours requise.
– absence : la fonction n'était pas requise et au moment où elle le fut, elle n'a pas été assurée.
– dégradation : la fonction est assurée, mais à un moment donné elle est assurée de manière incomplète ou erronée, elle est par exemple excessive, insuffisante, inversée, retardée ou instable.
– intempestif : sans être requise, la fonction est assurée à un moment donné alors qu'elle ne devrait pas l'être.

Un évènement redouté est classé suivant son indice de gravité défini suivant l'impact de la défaillance d'origine. Les niveaux d'indice de gravité proviennent généralement de normes : pour les plus connues, CEI 61508 sécurité fonctionnelle des systèmes électriques/électroniques et électroniques programmables, ainsi que ses adaptations CEI 61511 pour les procédés industriels, CEI 61513 pour le secteur du nucléaire, CEI 62061 pour la sécurité des machines, EN 50126, EN 50128 et EN 50129 pour le secteur du ferroviaire, et ISO 26262 pour le secteur de l'automobile. Par exemple, les différents niveaux suivants peuvent être considérés :
– Niveau 1 : défaillance minime, peu ou pas détectée par les utilisateurs du système, et telle que les prestations du système restent globalement assurées.
– Niveau 2 : défaillance provoquant une gêne avérée et telle que le système est encore utilisable, mais nécessite une intervention à court ou moyen terme.
– Niveau 3 : défaillance provoquant un arrêt du système et nécessitant une intervention avant remise en service.
– Niveau 4 : défaillance entraînant un risque corporel, mortel ou non, sur l'homme.

Une telle analyse préliminaire des risques peut se faire en remplissant le tableau 3.1 suivant :

Situation de vie	Fonction	Mode de défaillance	Scénario	Événement redouté		
				Système	Macro-système	Gravité

Tableau 3.1 – Tableau d'analyse préliminaire des risques fonctionnels.

3.2.1.3 Analyse des modes de défaillance, de leurs effets et de leur criticité

L'analyse des modes de défaillance, de leurs effets et de leur criticité, dont l'acronyme est l'AMDEC, est l'étude des causes des défaillances du système. Il s'agit d'une méthode d'analyse préventive de la fiabilité d'un système permettant de hiérarchiser les défaillances redoutées du système afin d'en définir des actions correctives pertinentes et efficaces. Elle définit pour chaque défaillance un indice de criticité prenant en compte ses effets (i.e. : les événements redoutés) avec leurs gravités issues de l'analyse préliminaire des risques, ses causes ainsi que la possibilité de la détecter ou non suivant ses causes. Suivant la valeur de l'indice de criticité, des plans d'actions peuvent être mis en place (prise en compte par un diagnostiqueur par exemple) afin éventuellement de diminuer cette criticité.

Nous avons vu que l'analyse préliminaire des risques liste les effets des défaillances avec des indices de gravité. Cette analyse n'est, en elle-même, néanmoins pas suffisante pour estimer complètement le risque associé. L'AMDEC permet donc de classer ces risques en y associant un indice C de criticité (aussi appelé indice de risque). Il s'agit du produit de trois facteurs, chacun noté de 0 à 10, représenté dans le tableau 3.2 ci-dessous : la gravité G (ou sévérité) de la défaillance, la fréquence F d'apparition de la défaillance et la probabilité D de non-détection de la défaillance. Cet indice de criticité permet de hiérarchiser les défaillances potentielles et de définir ainsi celles ayant besoin de plans d'actions à engager. Dans le cas où un plan d'action est défini, il faut alors par la suite ré-estimer l'indice en ré-évaluant les indices G de gravité, F de fréquence d'apparition et D de probabilité de non-détection.

Note	Gravité G	Fréquence d'apparition F	Probabilité de non-détection D
10	Mort d'homme	Permanent	Aucune probabilité de détection
5	Conséquences financières et/ou matérielles	Fréquent	Un système de détection est en place mais n'est pas infaillible
1	Pas grave	Rare	Le système de détection est infaillible

Tableau 3.2 – Facteurs de criticité des défaillances.

L'analyse préliminaire des risques et l'AMDEC peuvent être intégrées dans un unique tableau rempli et mis à jour progressivement lors de la phase de conception du système. Nous obtenons le tableau complet en ajoutant les colonnes du tableau 3.3 ci-dessous de l'AMDEC au tableau 3.1 de la page 65 de l'APR fonctionnels.

Cause		Détection		Criticité initiale	Plan d'action	Ré-estimation			Criticité ré-estimée
Intitulé	F	Moyen	D			G	F	D	

Tableau 3.3 – Tableau d'analyse des modes de défaillance, de leurs effets et de leur criticité.

La construction de l'AMDEC n'est pas la fin de la sûreté de fonctionnement car elle ne reste qu'un support à faire vivre pour l'analyse du système (qui reste indispensable une fois l'AMDEC construite). Une fois établie, elle n'est donc pas figée car elle évolue au fur et à mesure qu'avance la conception et jusqu'à la mise en service du système. Elle doit être revue en particulier quand les conditions d'utilisation changent, quand la conception du produit évolue, quand les règles d'utilisation changent, ou encore en fonction des retours que font les utilisateurs.

3.2.2 Défauts potentiels identifiés

Nous voyons là que, suite à la construction de l'AMDEC, l'ensemble des défauts potentiels qui devront être pris en compte par un outil de diagnostic est défini. Nous considérons pour la suite un ensemble $\Gamma = \{F_1; \ldots; F_k\}$ de défauts identifiés. De plus, pour simplifier la présentation, le cas sans

défaut sera désigné par convention par le défaut F_0 et sera un élément de cet ensemble de défauts : $\Gamma = \{F_0; F_1; \ldots; F_k\}$.

3.3 Caractéristiques des défauts

Une typologie des défauts consiste à définir un certain nombre de traits caractéristiques des défauts afin d'en faciliter la classification et l'étude. Cette partie va d'abord introduire différentes caractéristiques de défauts issues de la littérature, puis résumer ces caractéristiques afin d'obtenir une définition complète et correcte d'un défaut.

De nombreuses classifications de défauts peuvent être trouvées dans la littérature. Nous avons néanmoins observé ([BFDR10]) que toutes font une distinction entre le comportement du défaut, sa localisation et son effet sur le système. Considérons par exemple les deux couples suivants de défauts :

- la rupture d'une courroie d'entraînement d'un ventilateur et un court circuit d'une carte de puissance ;
- la rupture de l'axe de transmission d'un compresseur et l'encrassement d'un compresseur.

Nous remarquons que les deux défauts du premier couple ne sont pas localisés sur les mêmes composants d'un système, mais qu'ils ont cependant les mêmes caractéristiques comportementales : ils apparaissent brusquement et sont permanents. À l'inverse, les deux défauts du second couple sont localisés sur le même composant, mais n'ont pas les mêmes caractéristiques comportementales. Ceci amène donc naturellement à distinguer les défauts non seulement en fonction de leurs localisations et effets sur le système, mais aussi en fonction de leurs comportements.

3.3.1 La localisation des défauts

La partie opérante d'un système piloté est un ensemble de composants interagissant entre eux afin d'accomplir une fonction requise. Il est donc naturel de penser qu'un défaut puisse affecter soit un composant du système, soit un ensemble de composants.

3.3.1.1 Localisation classique selon la littérature

Selon [Bas99], trois localisations de défauts peuvent être définies sur un système : les défauts f_i d'actionneurs, les défauts f_o de capteurs et les défauts f_θ de composants ou du processus. En considérant les sorties observables y d'un système S d'état x, paramétré par θ et ayant les entrées u, qui s'écrit en négligeant la dynamique par l'équation 3.1 suivante :

$$y = S(x, \theta, u) \tag{3.1}$$

alors un défaut f_i perturbe la variable u des entrées du système, un défaut f_o perturbe la variable y des sorties du système et un défaut f_θ perturbe les paramètres θ du système.

Selon [BKLS03], [BJL+90] ou [Ise06], un défaut peut être localisé dans les différents types de composants du système :

- Dans un capteur, le défaut se caractérise par un écart entre la valeur réelle de la grandeur et sa mesure. Ce défaut se classe en fonction de son type : biais, dérive, modification du gain de mesure, valeurs aberrantes, blocage du capteur à une valeur atteinte ou à une coupure électrique du capteur.
- Dans le processus physique, le défaut est dû à des modifications de la structure (par exemple une fuite ou une rupture d'un organe) ou des paramètres du modèle (par exemple un encrassement d'un tuyau ou un bouchage partiel d'une conduite).

- Dans un actionneur, le défaut se traduit par une incohérence entre les commandes et la sortie du système (par exemple une pompe délivrant un débit incohérent avec sa caractéristique hydraulique).
- Dans l'unité de contrôle commande, le défaut se caractérise par un écart entre la valeur réelle de la sortie du contrôleur, selon l'algorithme implémenté, et sa mesure.

Enfin, [VRYK03] caractérise trois cas de localisation de défauts sur le système :
- Les changements grossiers de paramètres du modèle : dans toute modélisation, il existe des processus se produisant en dessous du niveau de détails du modèle ; ces processus non-modélisés sont typiquement rassemblés comme paramètres. Les défauts de paramètres surviennent quand il y a une perturbation de l'environnement entrant dans le processus, à travers une ou plusieurs variables. Par exemple, le changement du coefficient de transfert de chaleur dû à l'encrassement dans un échangeur thermique.
- Les changements structurels : en référence aux changements dans le processus lui-même, ils se produisent à cause de graves pannes dans le système et résultent de changements dans le flot d'informations entre les différentes variables. Par exemple, les pannes de contrôleur, les blocages de valves ou encore les fuites dans les tuyauteries.
- Les mauvais fonctionnements de capteurs ou d'actionneurs : une panne dans un des instruments de contrôle (i.e. : un capteur ou un actionneur) peut causer une déviation des variables d'état du système au-delà des limites acceptables.

3.3.1.2 Localisation suivant les exigences

Les exigences d'un système décrivent ce que le système doit être et doit faire. Ces exigences peuvent être fonctionnelles : c'est-à-dire qu'elles décrivent les caractéristiques du système ou des processus que le système doit exécuter. Elles peuvent aussi être non-fonctionnelles et décrire ainsi les propriétés que le système doit avoir. Les exigences sont par conséquent définies sur tout ou partie du système.

Par ailleurs, un défaut peut aussi être vu comme la non-vérification d'une exigence du système ; ce qui fait qu'un défaut peut être localisé soit dans un des composants du système ou soit dans un ensemble de composants suivant que l'exigence est donnée sur tout ou partie du système. Par exemple, pour un système en boucle fermée, l'usure ou un état non-prévu du processus physique peut amener à ce que le système de pilotage (au moins le module de régulation) pilote mal ce processus. Le défaut n'est donc pas seulement dû uniquement à ce système de pilotage ou au processus physique, mais à l'ensemble bouclé du système.

3.3.2 Le comportement des défauts

Comme nous avons pu le remarquer dans les deux précédents couples d'exemples, les défauts peuvent être regroupés suivant des caractéristiques similaires. Toutes les ruptures ou casses de composants, par exemple, sont des défauts dont l'occurrence se fait brusquement. À l'inverse, les usures sont des défauts apparaissant progressivement.

3.3.2.1 Les comportements suivant la littérature

Comportements suivant le type de composants Dans [Ise06], les principaux comportements de défauts sont donnés suivant le type de composant concerné. Deux catégories comportementales sont établies : la formation et le comportement temporel. Comme l'indique le tableau 3.4 de la page 69, la caractéristique « formation » distingue les défauts systématiques des défauts aléatoires, alors que la caractéristique « comportement temporel » distingue les défauts permanents, les défauts transitoires, les défauts intermittents, les défauts de bruit et les défauts de dérive.

Défauts		Type de composants			
		Composants mécaniques	Composants électriques	Matériel électronique	Logiciel
Formation	Systématique	✓		✓	✓
	Aléatoire		✓	✓	✓
Comportement temporel	Permanent			✓	✓
	Transitoire	✓	✓	✓	✓
	Intermittent		✓	✓	✓
	Bruité		✓	✓	
	Dérive	✓	✓	✓	

Tableau 3.4 – Comportements suivant le type concerné de composants.

- Un défaut systématique est un défaut dont l'occurrence est toujours au même moment. Elle peut résulter d'une usure ou du vieillissement du système, ou de la conséquence d'un fonctionnement spécifique.
- Un défaut aléatoire est un défaut dont l'occurrence se produit à n'importe quel moment, il n'y a aucune certitude sur celle-ci.
- Un défaut permanent est un défaut qui reste persistant après son occurrence.
- Un défaut transitoire ne persiste pas après son occurrence : c'est-à-dire qu'il apparaît puis disparaît sans qu'il n'y ait nécessairement eu d'actions correctrices.
- Un défaut intermittent est un défaut qui apparaît puis disparaît puis réapparaît puis disparaît de nouveau, et ainsi de suite sans qu'il n'y ait nécessairement eu d'actions correctrices.
- Un défaut bruité est un défaut qui apparaît comme un bruit permanent sans être nécessairement très important et sans nécessairement évoluer.
- Un défaut de dérive est un défaut qui évolue de manière progressive.

Comportement suivant la dépendance au temps [Ise97] spécifie la « dépendance au temps » (*time dependency*) d'un défaut qui distingue, comme l'indique le tableau 3.5 ci-dessous, les défauts brusques, les défauts progressifs et les défauts intermittents. La figure 3.3 suivante montre le comportement des défauts en fonction de la dépendance au temps.

Dépendance au temps	Explications
Défaut brusque	défaut dont l'apparition est brutale, la durée entre le moment où le système fonctionne normalement et le moment où il est en défaut est quasiment nulle. Généralement caractéristique des ruptures ou casses de composants.
Défaut progressif	défaut dont l'apparition est graduelle avec le temps. Généralement caractéristique d'usures de composants.
Défaut intermittent	défaut qui apparaît puis disparaît puis réapparaît puis disparaît de nouveau, et ainsi de suite sans nécessairement d'actions correctrices.

Tableau 3.5 – Comportement suivant la dépendance au temps.

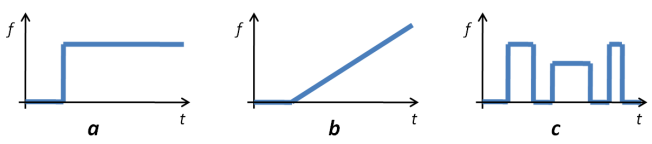

Figure 3.3 – Dépendance au temps des défauts.

Progression suivant la sévérité des défauts [BJL+90] détermine une progression dans la sévérité des défauts. Il faut distinguer les défauts naissants, les défauts passagers, les défauts permanents et les défauts catastrophiques.

- Un défaut naissant se caractérise par le fait que la dérive de la propriété caractéristique du système commence à apparaître. Cela représente les premiers instants du défaut.
- Un défaut passager est un défaut pour lequel la propriété caractéristique du système a dérivé puis est revenue en état normal. Le défaut ne reste pas présent après son apparition, il ne persiste pas.
- Un défaut permanent caractérise le fait que la propriété caractéristique a dérivé et ne revient plus à un état normal. Le défaut reste persistant après son apparition.
- Un défaut catastrophique est un défaut qui donne lieu à une situation accidentelle aux lourdes conséquences sur le système ou son environnement (pertes matérielles, financières ou humaines).

Autres caractéristiques Suivant divers documents de la littérature, différentes caractéristiques de défauts peuvent être définies, en mélangeant autant ceux caractérisant un système en entier que ceux n'en caractérisant que certaines parties. Remarquons que certaines de ces caractéristiques ont en partie déjà été présentées.

- Défaut soudain : défaut imprévisible pouvant intervenir à n'importe quel moment et cela même lorsqu'une maintenance préventive du système est respectée. Il peut être soit inhérent à des faiblesses de l'équipement, ou peut être dû à des évènements extérieurs au système. Il se caractérise par un saut brusque de son comportement nominal à un moment aléatoire.
- Défaut progressif : défaut faisant suite à l'usure du système ou de l'un de ses composants. Il s'agit d'une dérive progressive et de plus en plus importante du fonctionnement nominal du système.
- Défaut partiel : défaut survenant suite à un dysfonctionnement d'un composant, ou d'un ensemble de composants, du système. Seule une partie du système est en défaut et cela ne conduit pas nécessairement à un défaut global du système.
- Défaut par dégradation : défaut combinant les caractéristiques d'un défaut progressif et d'un défaut partiel. Il s'agit donc d'un dysfonctionnement touchant un composant, ou un ensemble de composants du système, mais dont la caractéristique est une dérive progressive de son fonctionnement nominal.
- Défaut complet : défaut caractérisant le fait que la fonction attendue du système disparaît complètement et de manière permanente. Le système soit ne fonctionne plus, c'est-à-dire qu'il est en arrêt et ne peut plus être mis en marche, soit fonctionne mais ne fournit aucune action dans son mode de fonctionnement nominal et ceci pour n'importe quel ordre.
- Défaut cataleptique : combinaison d'un défaut à la fois soudain et complet. La fonction du système disparaît totalement et soudainement.
- Défaut intermittent : caractérise le fait que le système est défaillant durant une période de temps limitée, puis retrouve son aptitude à accomplir la fonction attendue sans intervention corrective.

3.3.3 Conclusion sur les caractéristiques des défauts

Cette partie nous a permis de faire un point sur les différentes caractéristiques des défauts pouvant apparaître dans un système. Nous avons ainsi vu qu'un défaut se définit suivant différentes caractéristiques qui déterminent autant son comportement que sa localisation ou encore ses conséquences. Il est possible de résumer cela en trois principales caractéristiques :

- le comportement du défaut,
- l'effet du défaut,
- les conséquences du défaut.

Comportement d'un défaut Le comportement d'un défaut détermine son instant d'occurrence dans le temps, sa force d'apparition ainsi que sa durée de présence. L'instant d'occurrence d'un défaut, que nous réduirons à *occurrence* d'un défaut, peut être aléatoire, systématique ou dépendant d'un évènement interne ou externe au système. La force d'apparition d'un défaut peut être brusque ou progressive. Enfin, la durée de présence d'un défaut peut être permanente, transitoire ou intermittente.

Effet d'un défaut L'effet d'un défaut détermine sa prise en compte dans le système. Il s'agit de déterminer sa localisation dans le système ainsi que la ou les perturbations induites.

Un défaut peut être localisé soit sur une partie bien déterminée du système, soit dans un ensemble de composants du système, ou même sur le système complet.

Comme le représente la figure 3.4 ci-dessous, une partie bien déterminée du système est un capteur, un (ou des) composant(s) du processus physique, un actionneur, l'unité de contrôle commande ou encore un lien entre différents composants. Dans un capteur, un défaut perturbe les sorties du système. Dans un (ou des) composant(s) du processus physique, un défaut perturbe les états et/ou les paramètres du système. Dans un actionneur, un défaut perturbe les entrées du système. Dans l'unité de contrôle commande, un défaut perturbe l'algorithme de contrôle commande du système. Enfin, dans un lien entre différents composants, un défaut perturbe les différentes données (physiques ou informatiques) transitant par ce lien.

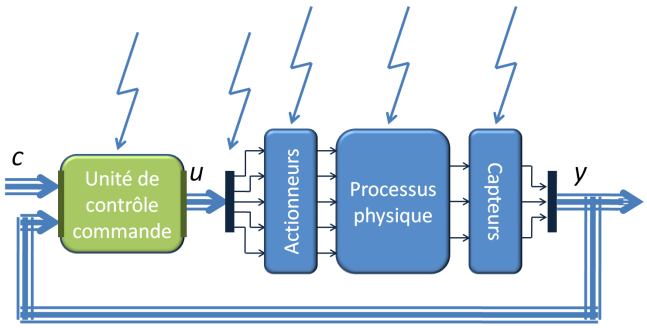

Figure 3.4 – Localisation des défauts selon la littérature.

Conséquences d'un défaut Les conséquences engendrées par un défaut, sur le système lui-même et/ou son environnement, sont à déterminer suivant les pertes potentielles (matérielles et/ou humaines) qu'il peut générer. Cette caractéristique est déterminée lors de l'étude de sûreté de fonctionnement du système.

3.4 Construction du modèle de défauts

La partie précédente nous a permis de mettre en évidence les principales caractéristiques d'un défaut qui sont son comportement, son effet sur le système ainsi que ses conséquences sur le système et son environnement. Dans le but d'intégrer les défauts dans le modèle de bon fonctionnement du système, il est important de distinguer quelles sont les caractéristiques nécessaires à cette intégration.

Rappelons que notre but est d'étudier la diagnosticabilité des défauts afin de générer le diagnostiqueur du système associé à cette étude. La caractéristique des conséquences du défaut sur le système et son environnement n'est par conséquent pas prise en compte car ces conséquences sont supposées être déterminées lors de l'étude de sûreté de fonctionnement du système. Lors des phases de modélisation et d'étude de la diagnosticabilité des défauts, ces conséquences sont censées être connues et maîtrisées.

Pour caractériser un défaut et comme le représente la figure 3.5 de la page 72, nous considérons donc uniquement son comportement et son effet sur le système. Pour intégrer un défaut dans le modèle

de bon fonctionnement du système, il convient donc de bien définir ces deux caractéristiques. Cela permet de construire le modèle de défaut du système : le système sous la présence du défaut.

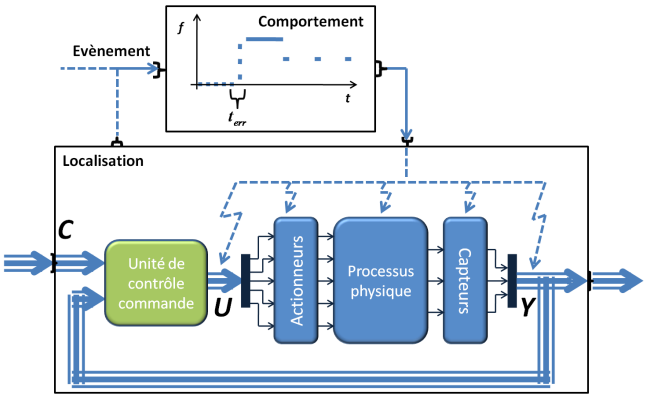

Figure 3.5 – Construction des défauts.

3.4.1 Modélisation du comportement d'un défaut

Le comportement d'un défaut détermine son *occurrence*, sa *force d'apparition* ainsi que sa *durée de présence*. L'occurrence d'un défaut, qui désigne l'instant du temps où il apparaît, peut être aléatoire, systématique ou même dépendre d'un évènement interne ou externe au système. Sa force d'apparition peut être brusque ou progressive. Enfin sa durée de présence peut être permanente, transitoire ou intermittente.

Nous allons décrire le comportement d'un défaut par un signal temporel évoluant entre 0 et 1. La valeur 0 représente l'absence du défaut et la valeur 1 représente sa présence « complète ». Que l'occurrence d'un défaut soit aléatoire, systématique ou dépendant d'un évènement, remarquons que le comportement du défaut sera toujours décrit suivant un instant $\mathrm{t}_n \in \mathbb{T}_\iota$ du temps correspondant à cette occurrence ; qu'il soit donc contrôlé ou non.

Le comportement d'un défaut permanent et brusque est décrit par l'équation suivante :

$$dft_{(F,\mathrm{t}_n)}(\mathrm{t}) = \begin{cases} 0 & \text{si } \mathrm{t} \in [0;\mathrm{t}_n[\\ 1 & \text{si } \mathrm{t} \in [\mathrm{t}_n;\infty[\end{cases}$$

Le comportement d'un défaut permanent et progressif est décrit par l'équation suivante :

$$dft_{(F,\mathrm{t}_n)}(\mathrm{t}) = \begin{cases} 0 & \text{si } \mathrm{t} \in [0;\mathrm{t}_n[\\ min\{1; a \cdot (\mathrm{t} - \mathrm{t}_n)\} & \text{si } \mathrm{t} \in [\mathrm{t}_n;\infty[\end{cases}$$

où a représente la pente de progression. Le comportement d'un défaut transitoire et brusque est décrit par l'équation suivante :

$$dft_{(F,\mathrm{t}_n)}(\mathrm{t}) = \begin{cases} 0 & \text{si } \mathrm{t} \in [0;\mathrm{t}_n[\cup[\mathrm{t}_n+\delta;\infty[\\ 1 & \text{si } \mathrm{t} \in [\mathrm{t}_n;\mathrm{t}_n+\delta[\end{cases}$$

où δ représente la durée de transition. Le comportement d'un défaut transitoire et progressif est décrit par l'équation suivante :

$$dft_{(F,\mathrm{t}_n)}(\mathrm{t}) = \begin{cases} 0 & \text{si } \mathrm{t} \in [0;\mathrm{t}_n[\\ min\{1; a \cdot (\mathrm{t} - \mathrm{t}_n)\} & \text{si } \mathrm{t} \in [\mathrm{t}_n;\mathrm{t}_n+\delta[\\ max\{0; min\{1; a \cdot \delta\} - a \cdot (\mathrm{t} - (\mathrm{t}_n+\delta))\} & \text{si } \mathrm{t} \in [\mathrm{t}_n+\delta;\infty[\end{cases}$$

où a représente la pente de progression et δ représente la durée de transition. Le comportement d'un défaut intermittent et brusque est décrit par l'équation suivante :

$$dft_{(F,\mathrm{t}_n)}(\mathrm{t}) = \begin{cases} 0 & \text{si } \mathrm{t} \in [0;\mathrm{t}_n[\cup \bigcup_{m\in\mathbb{N}}[\mathrm{t}_n + (m+r)\cdot p; \mathrm{t}_n + (m+1)\cdot p[\\ 1 & \text{si } \mathrm{t} \in \bigcup_{m\in\mathbb{N}}[\mathrm{t}_n + m\cdot p; \mathrm{t}_n + (m+r)\cdot p[\end{cases}$$

où p est la période entre deux apparitions du défaut et $r \in]0;1[$ est le pourcentage de présence du défaut sur la période p. Enfin, le comportement d'un défaut intermittent et progressif est décrit par la fonction suivante donnée par récurrence :

$$\begin{array}{rccc} dft_{(-1)(F,\mathrm{t}_n)} : & [0;\mathrm{t}_n[& \longrightarrow & [0;1] \\ & \mathrm{t} & \longmapsto & 0 \end{array}$$

$$\begin{array}{rccc} dft^{+}_{(0)(F,\mathrm{t}_n)} : &]\mathrm{t}_n;\mathrm{t}_n + p\cdot r] & \longrightarrow & [0;1] \\ & \mathrm{t} & \longmapsto & min\{1; a(\mathrm{t}-\mathrm{t}_n)\} \end{array}$$

$$\begin{array}{rccc} dft^{-}_{(0)(F,\mathrm{t}_n)} : &]\mathrm{t}_n + p\cdot r;\mathrm{t}_n + p] & \longrightarrow & [0;1] \\ & \mathrm{t} & \longmapsto & max\{0; dft^{+}_{(0)(F,\mathrm{t}_n)}(\mathrm{t}_n + p\cdot r) - a(\mathrm{t}-(\mathrm{t}_n + p\cdot r))\} \end{array}$$

et pour $m \in \mathbb{N}$, pour $\mathrm{t} \in]\mathrm{t}_n + (m+1)\cdot p; \mathrm{t}_n + (m+1)\cdot p + (p\cdot r)]$:

$$dft^{+}_{(m+1)(F,\mathrm{t}_n)}(\mathrm{t}) = min\{1; dft^{-}_{(m)(F,\mathrm{t}_n)}(\mathrm{t}_n + (m+1)\cdot p) + a(\mathrm{t} - (\mathrm{t}_n + (m+1)\cdot p))\}$$

pour $\mathrm{t} \in]\mathrm{t}_n + (m+1)\cdot p + (p\cdot r); \mathrm{t}_n + (m+2)\cdot p]$:

$$dft^{-}_{(m+1)(F,\mathrm{t}_n)}(\mathrm{t}) = max\{0; dft^{+}_{(m+1)(F,\mathrm{t}_n)}(\mathrm{t}_n + (m+1)\cdot p + (p\cdot r)) - a(\mathrm{t} - (\mathrm{t}_n + (m+1)\cdot p + (p\cdot r)))\}$$

où p est la période entre deux apparitions du défaut, $r \in]0;1[$ est le pourcentage de présence du défaut sur la période p et a est la pente de progression du défaut.

3.4.2 Modélisation de l'effet d'un défaut

L'effet d'un défaut détermine sa prise en compte dans le système : c'est-à-dire la ou les perturbations induites ainsi que sa localisation dans le système.

Pour un défaut $F \in \Gamma \setminus \{F_0\}$ et une occurrence $\mathrm{t}_n \in \mathbb{T}_\iota$, le modèle $\overline{S_F}$ du système complet $\overline{S}$, sous la présence du défaut F apparaissant à l'occurrence t_n, est défini en considérant l'ensemble des équations du modèle de bon fonctionnement (que nous notons $\overline{S_{F_0}}$) où la variable perturbée considérée v (une commande u, une mesure y, un état x ou un paramètre θ) est remplacée par sa perturbation v_F, décrite par l'équation suivante avec $dft_{(F,\mathrm{t}_n)}$ représentant le comportement du défaut :

$$v_F(\mathrm{t}) = pert_F(\mathrm{t}, v(\mathrm{t}), dft_{(F,\mathrm{t}_n)}(\mathrm{t}))$$

3.4.2.1 Modélisation des perturbations induites

Une perturbation peut être additive, multiplicative, sinusoïdale, limitative ou même stopper le signal à sa dernière valeur. Une perturbation additive est décrite par l'équation suivante :

$$v_F(\mathrm{t}) = v(\mathrm{t}) + a\cdot dft_{(F,\mathrm{t}_n)}(\mathrm{t})$$

où $dft_{(F,\mathrm{t}_n)}$ représente le comportement du défaut et a le paramètre additif. Une perturbation multiplicative est décrite par l'équation suivante :

$$v_F(\mathrm{t}) = \begin{cases} v(\mathrm{t}) & \text{si } dft_{(F,\mathrm{t}_n)}(\mathrm{t}) = 0 \\ v(\mathrm{t}) \cdot m \cdot dft_{(F,\mathrm{t}_n)}(\mathrm{t}) & \text{si } dft_{(F,\mathrm{t}_n)}(\mathrm{t}) > 0 \end{cases}$$

où $dft_{(F,\mathrm{t}_n)}$ représente le comportement du défaut et m le paramètre multiplicatif. Une perturbation sinusoïdale est décrite par l'équation suivante :

$$v_F(\mathrm{t}) = v(\mathrm{t}) + dft_{(F,\mathrm{t}_n)}(\mathrm{t}) \cdot (a \cdot sin(f \cdot \mathrm{t}))$$

où $dft_{(F,\mathrm{t}_n)}$ représente le comportement du défaut, a le paramètre d'amplitude et f le paramètre de fréquence (en radian par unité de temps). Une perturbation limitative est décrite par l'équation suivante :

$$v_F(\mathrm{t}) = \begin{cases} v_{max} & \text{si } v(\mathrm{t}) > v_{max} \text{ et } dft_{(F,\mathrm{t}_n)}(\mathrm{t}) > 0 \\ v_{min} & \text{si } v(\mathrm{t}) < v_{min} \text{ et } dft_{(F,\mathrm{t}_n)}(\mathrm{t}) > 0 \\ v(\mathrm{t}) & \text{si } (v_{min} \leq v(\mathrm{t}) \leq v_{max} \text{ et } dft_{(F,\mathrm{t}_n)}(\mathrm{t}) > 0) \text{ ou } dft_{(F,\mathrm{t}_n)}(\mathrm{t}) = 0 \end{cases}$$

où $dft_{(F,\mathrm{t}_n)}$ représente le comportement du défaut, v_{max} le paramètre de valeur maximum et v_{min} le paramètre de valeur minimum. Enfin une perturbation stoppant le signal à sa dernière valeur est décrite par l'équation suivante :

$$v_F(\mathrm{t}) = \begin{cases} v(\mathrm{t}) & \text{si } dft_{(F,\mathrm{t}_n)}(\mathrm{t}) = 0 \\ v({\mathrm{t}_n}^-) & \text{si } dft_{(F,\mathrm{t}_n)}(\mathrm{t}) > 0 \end{cases}$$

où $dft_{(F,\mathrm{t}_n)}$ représente le comportement du défaut.

3.4.2.2 Modélisation des localisations

Perturbations de mesures Un défaut $F_y \in \Gamma$ de capteur se modélise par une perturbation des mesures y et se représente par $y_{F_y}(\mathrm{t}) = Pert_{F_y}(\mathrm{t}, y(\mathrm{t}), dft_{(F_y,\mathrm{t}_n)}(\mathrm{t}))$, où $dft_{(F_y,\mathrm{t}_n)}$ désigne le comportement du défaut et $\mathrm{t}_n \in \mathbb{T}_\iota$ son occurrence. Les équations de fonctionnement du système complet $\overline{S}$ avec un défaut $F_y \in \Gamma$ de capteur sont données par l'ensemble 3.2 suivant :

$$\overline{S_{F_y}} : \begin{cases} S_{F_y} : \begin{cases} x(\mathrm{t}^+) &= f(x(\mathrm{t}), \theta, u(\mathrm{t}), d(\mathrm{t})) \\ y(\mathrm{t}) &= Pert_{F_y}(\mathrm{t}, g(x(\mathrm{t})), dft_{(F_y,\mathrm{t}_n)}(\mathrm{t})) \\ a(\mathrm{t}^+) &= h(c(\mathrm{t}), a(\mathrm{t}), y(\mathrm{t})) \\ u(\mathrm{t}) &= k(a(\mathrm{t})) \\ x(0) &= x_{init} \\ a(0) &= a_{init} \end{cases} \\ S^M : \begin{cases} x^M(\mathrm{t}^+) &= f^M(x^M(\mathrm{t}), \theta^M, u^M(\mathrm{t}), d^M(\mathrm{t})) \\ y^M(\mathrm{t}) &= g^M(x^M(\mathrm{t})) \\ a^M(\mathrm{t}^+) &= h(c(\mathrm{t}), a^M(\mathrm{t}), y^M(\mathrm{t})) \\ u^M(\mathrm{t}) &= k(a^M(\mathrm{t})) \\ x^M(0) &= x^M_{init} \\ a^M(0) &= a^M_{init} \end{cases} \end{cases} \tag{3.2}$$

Perturbations d'états Un défaut $F_x \in \Gamma$ dans le processus physique, dû à des modifications de la structure du système, se modélise par une perturbation des états x et se représente par $x_{F_x}(\mathrm{t}) = Pert_{F_x}(\mathrm{t}, x(\mathrm{t}), dft_{(F_x,\mathrm{t}_n)}(\mathrm{t}))$, où $dft_{(F_x,\mathrm{t}_n)}$ désigne le comportement du défaut et $\mathrm{t}_n \in \mathbb{T}_\iota$ son occurrence. Les équations de fonctionnement du système $\overline{S}$ avec un défaut $F_x \in \Gamma$ dans le processus physique, dû à des modifications de la structure, sont données par l'ensemble 3.3 suivant :

$$\overline{S_{F_x}} : \begin{cases} S_{F_x} : \begin{cases} x(\mathrm{t}^+) &= Pert_{F_x}(\mathrm{t}, f(x(\mathrm{t}), \theta, u(\mathrm{t}), d(\mathrm{t})), dft_{(F_x,\mathrm{t}_n)}(\mathrm{t})) \\ y(\mathrm{t}) &= g(x(\mathrm{t})) \\ a(\mathrm{t}^+) &= h(c(\mathrm{t}), a(\mathrm{t}), y(\mathrm{t})) \\ u(\mathrm{t}) &= k(a(\mathrm{t})) \\ x(0) &= Pert_{F_x}(0, x_{init}, dft_{(F_x,\mathrm{t}_n)}(0)) \\ a(0) &= a_{init} \end{cases} \\ S^M : \begin{cases} x^M(\mathrm{t}^+) &= f^M(x^M(\mathrm{t}), \theta^M, u^M(\mathrm{t}), d^M(\mathrm{t})) \\ y^M(\mathrm{t}) &= g^M(x^M(\mathrm{t})) \\ a^M(\mathrm{t}^+) &= h(c(\mathrm{t}), a^M(\mathrm{t}), y^M(\mathrm{t})) \\ u^M(\mathrm{t}) &= k(a^M(\mathrm{t})) \\ x^M(0) &= x^M_{init} \\ a^M(0) &= a^M_{init} \end{cases} \end{cases} \tag{3.3}$$

Perturbations de paramètres Un défaut $F_\theta \in \Gamma$ dans le processus physique, dû à des modifications des paramètres du système, se modélise par une perturbation des paramètres θ et se représente par $\theta_{F_\theta}(\mathrm{t}) = Pert_{F_\theta}(\mathrm{t}, \theta, dft_{(F_\theta,\mathrm{t}_n)}(\mathrm{t}))$, où $dft_{(F_\theta,\mathrm{t}_n)}$ désigne le comportement du défaut et $\mathrm{t}_n \in \mathbb{T}_\iota$ son occurrence. Les équations de fonctionnement du système $\overline{S}$ avec un défaut $F_\theta \in \Gamma$ dans le processus physique, dû à des modifications des paramètres, sont données par l'ensemble 3.4 suivant :

$$\overline{S_{F_\theta}} : \begin{cases} S_{F_\theta} : \begin{cases} x(\mathrm{t}^+) &= f(x(\mathrm{t}), Pert_{F_\theta}(\mathrm{t}, \theta, dft_{(F_\theta,\mathrm{t}_n)}(\mathrm{t})), u(\mathrm{t}), d(\mathrm{t})) \\ y(\mathrm{t}) &= g(x(\mathrm{t})) \\ a(\mathrm{t}^+) &= h(c(\mathrm{t}), a(\mathrm{t}), y(\mathrm{t})) \\ u(\mathrm{t}) &= k(a(\mathrm{t})) \\ x(0) &= x_{init} \\ a(0) &= a_{init} \end{cases} \\ S^M : \begin{cases} x^M(\mathrm{t}^+) &= f^M(x^M(\mathrm{t}), \theta^M, u^M(\mathrm{t}), d^M(\mathrm{t})) \\ y^M(\mathrm{t}) &= g^M(x^M(\mathrm{t})) \\ a^M(\mathrm{t}^+) &= h(c(\mathrm{t}), a^M(\mathrm{t}), y^M(\mathrm{t})) \\ u^M(\mathrm{t}) &= k(a^M(\mathrm{t})) \\ x^M(0) &= x^M_{init} \\ a^M(0) &= a^M_{init} \end{cases} \end{cases} \tag{3.4}$$

Perturbations de commandes Un défaut $F_u \in \Gamma$ d'actionneur se modélise par une perturbation des commandes u et se représente par $u_{F_u}(\mathrm{t}) = Pert_{F_u}(\mathrm{t}, u(\mathrm{t}), dft_{(F_u,\mathrm{t}_n)}(\mathrm{t}))$, où $dft_{(F_u,\mathrm{t}_n)}$ désigne le comportement du défaut et $\mathrm{t}_n \in \mathbb{T}_\iota$ son occurrence. Les équations de fonctionnement du système $\overline{S}$ avec un défaut $F_u \in \Gamma$ d'actionneur sont données par l'ensemble 3.5 suivant :

$$\overline{S_{F_u}} : \begin{cases} S_{F_u} : \begin{cases} x(\mathrm{t}^+) &= f(x(\mathrm{t}), \theta, u(\mathrm{t}), d(\mathrm{t})) \\ y(\mathrm{t}) &= g(x(\mathrm{t})) \\ a(\mathrm{t}^+) &= h(c(\mathrm{t}), a(\mathrm{t}), y(\mathrm{t})) \\ u(\mathrm{t}) &= Pert_{F_u}(\mathrm{t}, k(a(\mathrm{t})), dft_{(F_u,\mathrm{t}_n)}(\mathrm{t})) \\ x(0) &= x_{init} \\ a(0) &= a_{init} \end{cases} \\ S^M : \begin{cases} x^M(\mathrm{t}^+) &= f^M(x^M(\mathrm{t}), \theta^M, u^M(\mathrm{t}), d^M(\mathrm{t})) \\ y^M(\mathrm{t}) &= g^M(x^M(\mathrm{t})) \\ a^M(\mathrm{t}^+) &= h(c(\mathrm{t}), a^M(\mathrm{t}), y^M(\mathrm{t})) \\ u^M(\mathrm{t}) &= k(a^M(\mathrm{t})) \\ x^M(0) &= x^M_{init} \\ a^M(0) &= a^M_{init} \end{cases} \end{cases} \tag{3.5}$$

3.5 Bibliothèque de défauts

Nous avons développé une bibliothèque MATLAB/Simulink© de défauts permettant d'intégrer directement dans un modèle de simulation MATLAB/Simulink© des défauts potentiels, afin de simuler le comportement du système sous la présence de ces défauts intégrés. Celle-ci nous a permis, comme nous allons le voir dans la partie suivante, d'intégrer directement les défauts potentiels de la ligne d'air du système pile à combustible.

3.5.1 Composants de la librairie

Cette librairie, représentée par la figure 3.6 ci-dessous, a été construite en tenant compte de la séparation que nous venons de donner : le comportement du défaut et son effet sur le système. Cette librairie est donc composée de deux blocs distincts : un bloc « *signal de défaut* » et un bloc « *perturbation* ».

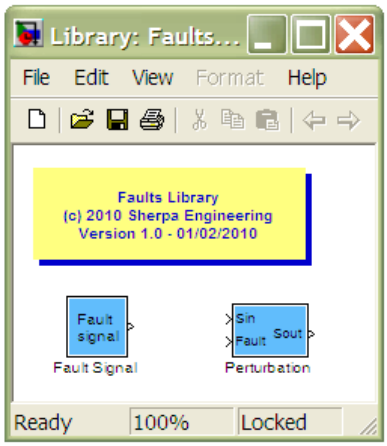

Figure 3.6 – Librairie de défauts développée sous MATLAB/Simulink©.

Le bloc *signal de défaut* permet de représenter le comportement d'un défaut. Il émet un signal évoluant entre 0 et 1 et paramétré suivant le comportement désiré : occurrence aléatoire ou à un instant donné ou dépendant d'un évènement, force d'apparition brusque ou progressive, ainsi que durée de présence permanente ou transitoire ou intermittente. La valeur 0, du signal émis par le bloc, représente l'absence du défaut (le système fonctionne normalement), alors que la valeur 1 représente sa présence « complète ».

Le bloc *perturbation* permet de représenter l'effet d'un défaut. Ce bloc perturbe les signaux quelconques des composants affectés par le défaut. Il est paramétrable suivant la perturbation désirée : additive, multiplicative, sinusoïdale, limitative ou stop. Les signaux quelconques des composants affectés par le défaut peuvent être des variables d'entrées u, des variables de sorties y, des variables d'état x, ou des constantes de paramètres θ. Ce bloc *perturbation* est contrôlé par un bloc *signal de défaut* qui lui est relié en entrée afin de fonctionner suivant le comportement du défaut, comme nous l'avons décrit dans les équations de perturbation d'une variable.

3.5.2 Intégrations types des composants de la librairie

Comme nous l'avons présenté dans [BFDR10], cette librairie est parfaitement bien adaptée pour les systèmes modélisés par l'assemblage de plusieurs composants génériques. D'une part, même si un défaut perturbe plusieurs composants, plusieurs blocs *perturbation* seront alors nécessaires mais un unique bloc *signal de défaut* sera alors nécessaire pour caractériser son comportement et ainsi contrôler ces blocs *perturbation*. D'autre part, tous les défauts potentiels d'un même composant peuvent être intégrés

en ajoutant les blocs *perturbation* adéquats directement à l'intérieur du composant ; un nouveau port d'entrée, de défaut, est alors ajouté au composant pour contrôler l'un de ces blocs *perturbation* par un bloc *signal de défaut*. Ceci est particulièrement intéressant lorsque les composants sont réutilisés pour d'autres systèmes.

Cette librairie a été utilisée pour intégrer les défauts potentiels dans le modèle MATLAB/Simulink© de la ligne d'air. D'une manière générique et comme nous l'avons présenté dans [BDRF11b], nous avons établi trois types d'intégration d'un défaut : l'intégration externe, l'intégration interne et l'intégration par ajout de composants.

3.5.2.1 Intégration externe

Pour ce type d'intégration, représentée par la figure 3.7 suivante, les deux blocs *signal de défaut* et *perturbation* de la librairie sont intégrés directement sur un ou plusieurs liens reliant le ou les composants affectés. Pour un composant affecté, le bloc *perturbation* est donc inséré sur le lien d'entrée ou de sortie du composant suivant le type de défaut. Le bloc *signal de défaut* lui est ensuite relié.

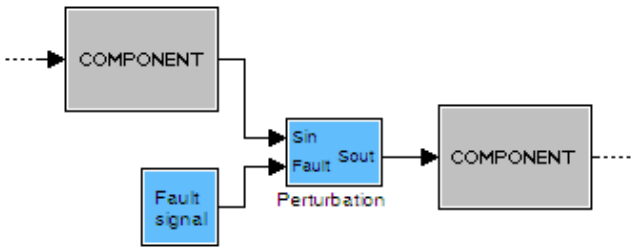

Figure 3.7 – Intégration externe d'un défaut.

3.5.2.2 Intégration interne

Pour ce type d'intégration, représentée par la figure 3.8 suivante, il faut entrer à l'intérieur du composant. Un bloc *perturbation* est inséré sur un lien sortant soit d'une constante, soit d'un bloc fonctionnel (opérateurs arithmétiques, intégrateurs, etc). Un nouveau port d'entrée de défaut est ajouté au composant pour contrôler ce bloc *perturbation* par un bloc *signal de défaut*.

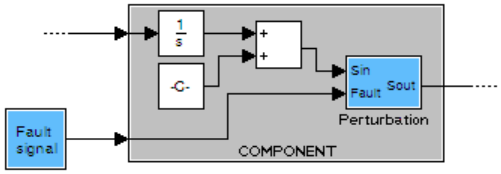

Figure 3.8 – Intégration interne d'un défaut.

3.5.2.3 Intégration par ajout de composants

Pour ce type d'intégration, représenté par la figure 3.9 suivante de la page 78, de nouveaux composants doivent être ajoutés. Ces nouveaux composants représentent de nouveaux actionneurs, contrôlés par un bloc *signal de défaut*, ainsi que de nouveaux constituants physiques.

Une fuite dans une tuyauterie par exemple est représentée par un trou dans le tuyau. Cela peut être modélisé par une nouvelle vanne reliée vers l'extérieur et contrôlée de manière à être ouverte lorsque la fuite est souhaitée présente.

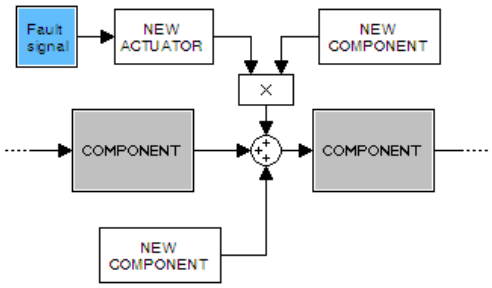

Figure 3.9 – Intégration d'un défaut par ajout de composants.

3.6 Application sur le cas d'étude

Nous allons mettre en pratique sur le cas d'étude, la ligne d'air du système pile à combustible, la typologie des défauts que nous venons de présenter dans ce chapitre. Nous n'allons pas faire une étude de sûreté de fonctionnement de cette ligne d'air, car cela est hors de portée du document, mais nous allons néanmoins présenter les principales idées ayant conduit aux défauts considérés. Nous présenterons ensuite les caractéristiques des différents défauts puis montrerons comment nous les avons intégrés dans le modèle de simulation de cette ligne d'air.

3.6.1 Identification des défauts

Comme nous l'avons vu au chapitre 2 lors la présentation du système pile à combustible, ce système est composé d'une multitude de composants variés en interaction permanente et combinant de multiples phénomènes physiques : thermodynamique, hydraulique et électrique. De ce fait, beaucoup de défauts peuvent potentiellement apparaître. Leurs conséquences peuvent être des pertes de performance du système, dans les meilleurs des cas. Dans le pire des cas, elles peuvent être des pannes ou dysfonctionnements de composants impliquant de sérieux dégâts autant pour l'intégrité du système en lui-même que pour son environnement. Par exemple, [ASC01] montre que les caractéristiques chimiques de l'hydrogène (i.e. : une petite molécule ayant une forte propension à s'échapper par de petites ouvertures) le rendent approprié aux fuites qui peuvent provoquer une explosion ayant un impact catastrophique sur l'environnement du système. Remarquons que les problématiques de fuites d'hydrogène de systèmes pile à combustible sont étudiées dans [ISM08], [AGPV04] et [PSP05] par des approches de diagnostic à base de modèles.

Durant le projet FISYPAC, une étude de sûreté de fonctionnement a été menée et une identification de tous les défauts potentiels a ainsi été réalisée. Il s'agit des défauts impactant l'intégrité de la pile (notamment le cœur même de la pile) ainsi que ceux liés à l'utilisation de l'hydrogène. Remarquons qu'aucun défaut impactant les performances du système n'a été identifié. Cela vient du fait que ce type de défauts est généralement attribué à du confort d'utilisation du système, or le but de ce projet était avant tout de faire une preuve de faisabilité (i.e. : un démonstrateur véhicule) et seuls les défauts critiques au niveau sécurité ont été pris en compte.

Les défauts d'actionneurs et de capteurs sont majoritaires. Néanmoins d'autres défauts, directement liés à des composants physiques, ont été ajoutés à cause de leurs relations avec l'intégrité de la pile ou l'utilisation de l'hydrogène. Citons par exemple : une fuite d'hydrogène (que nous avons déjà expliquée), une fuite d'air qui peut non seulement impacter les performances du système mais aussi provoquer des détériorations d'actionneurs (utilisation du compresseur à plus fort régime), ou encore un défaut de l'humidificateur de la ligne d'air qui peut provoquer un assèchement ou une humidification trop importante du cœur de la pile avec un risque grave d'endommagement de cette pile.

Concernant la ligne d'air et comme elle ne manipule pas d'hydrogène, seuls les défauts pouvant soit impacter l'intégrité de la pile ou détériorer les composants ont été identifiés. Ces défauts potentiels identifiés, présentés dans [BDRF11b], sont décrits dans le tableau 3.6 suivant :

blocage du compresseur	blocage de l'électrovanne
encrassement du compresseur	encrassement de l'électrovanne
fuite d'air dans la tuyauterie	défaut de mesure du capteur de débit
	défaut de mesure du capteur de pression

Tableau 3.6 – Défauts potentiels identifiés.

Remarquons que nous n'avons pas traité de défauts liés à l'humidificateur pouvant provoquer un assèchement ou une humidification trop importante du cœur de la pile, et risquer ainsi de l'endommager gravement. En effet, il apparaît dans la littérature que le traitement de ces défauts se réalise en « se plaçant » directement à l'intérieur de la pile. [Fou10] par exemple surveille des capteurs d'humidité internes à la pile en utilisant les techniques à base de modèles afin de lier les mesures de tension et d'intensité électrique de la pile aux valeurs d'humidité de ces capteurs.

Pour ces différents défauts identifiés, nous nous sommes basés sur l'expertise des personnes ayant travaillé sur le système ainsi que celles maîtrisant les différents composants du système. De ce fait et pour un avancement plus rapide dans la mise en application sur le cas d'étude, nous ne nous sommes que très peu attardés sur l'état de l'art.

3.6.2 Défauts liés au compresseur

Un compresseur est composé d'une partie mécanique et d'une partie électronique. La partie mécanique comprend le compresseur physique (i.e. : le bloc fermé avec les ailettes fixées sur un axe) le moteur et le cardan de transmission entre le moteur et le compresseur. Notons que ces trois éléments peuvent très bien constituer un unique composant. La partie électronique comprend les cartes de puissance et de contrôle du compresseur ainsi que les différents câbles de liaisons électriques et électroniques.

Selon [BH96] et [Blo66], les défauts potentiels d'un compresseur peuvent être des pertes de capacité, qui décrivent usuellement les différences entre le débit ciblé et celui obtenu, les bruits ou vibrations et les pannes de fonctionnement. Ainsi les défauts pouvant soit impacter l'intégrité de la pile ou détériorer les composants sont les défauts pour lesquels le compresseur ne fournit pas la puissance requise. Un blocage, noté $F_{LockCmpr}$ (pour *lock of the compressor*), ou un encrassement du compresseur, noté $F_{DirtCmpr}$ (pour *dirtying of the compressor*), sont les deux défauts identifiés comme liés à ces impacts sur la pile ou les composants de la ligne. La figure 3.10 ci-dessous représente la localisation de ces deux défauts sur cette ligne d'air.

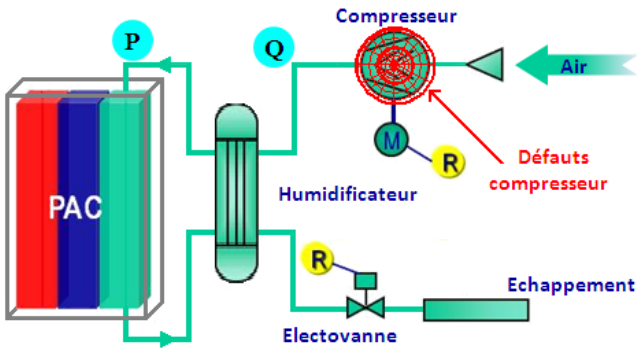

Figure 3.10 – Localisation de défauts du compresseur dans la ligne d'air.

3.6.2.1 Blocage du compresseur

Un blocage du compresseur peut être dû à une casse matérielle (la casse d'une ailette par exemple) ou à un problème électronique. Ainsi, outre le fait que ce défaut détériore le compresseur en lui-même ainsi que ses composants annexes (i.e. : les cartes électroniques de puissance et de contrôle), il peut aussi causer de sérieux dégâts sur l'intégrité de la pile. En effet, lorsque le compresseur se bloque, d'une part la pile n'est plus alimentée en oxygène et d'autre part la différence de pression entre l'anode et la cathode devient importante. Un risque d'endommagement du cœur de la pile est donc fortement possible.

Caractérisation du défaut

Un blocage du compresseur provoque une chute brutale à nul du débit d'air en sortie du compresseur (i.e. : une chute à 0 gramme par seconde). Nous obtenons ainsi facilement le comportement du défaut et son effet.

Comportement : D'une manière générale, il s'agit d'un défaut apparaissant de manière brusque, à un moment aléatoire ou pouvant dépendre d'un évènement et dont la durée de présence est permanente. Les trois attributs du comportement du défaut $F_{LockCmpr}$ sont donc :

- Occurrence : à un instant quelconque $\mathrm{t}_n \in \mathbb{T}_{0.01}$.
- Force d'apparition : brusque.
- Durée de présence : permanente.

Le comportement du défaut $F_{LockCmpr}$ de blocage du compresseur est donné, pour tout instant $\mathrm{t} \in \mathbb{T}_{0.01}$, par l'équation suivante :

$$dft_{(F_{LockCmpr},\mathrm{t}_n)}(\mathrm{t}) = \begin{cases} 0 & \text{si } \mathrm{t} \in [0;\mathrm{t}_n[\\ 1 & \text{si } \mathrm{t} \in \mathbb{T}_{0.01} \setminus [0;\mathrm{t}_n[\end{cases}$$

Effet : Ce défaut concerne le compresseur, plus particulièrement le débit d'air en sortie du compresseur qui chute à nul (i.e. : à 0 gramme par seconde). La variable Q_{cmpr} du débit d'air en sortie du compresseur est par conséquent perturbée multiplicativement et sa perturbation $Q_{cmpr_{F_{LockCmpr}}}$ est donnée, pour tout $\mathrm{t} \in \mathbb{T}_{0.01}$, par l'équation suivante :

$$Q_{cmpr_{F_{LockCmpr}}}(\mathrm{t}) = Q_{cmpr}(\mathrm{t}) \cdot (1 - dft_{(F_{LockCmpr},\mathrm{t}_n)}(\mathrm{t}))$$

Intégration du défaut

La figure 3.11 de la page 81 montre l'intégration du blocage du compresseur dans le modèle MATLAB/Simulink© de la ligne d'air. La partie gauche représente ce compresseur sans l'intégration du défaut et la partie droite le représente avec l'intégration. Il s'agit d'une intégration externe. La variable Q_{cmpr} de débit d'air en sortie du compresseur est perturbée par un bloc *perturbation*, paramétré selon l'effet que nous venons de présenter, qui est contrôlé par un bloc *signal de défaut* paramétré selon le comportement $dft_{(F_{LockCmpr},\mathrm{t}_n)}$.

Simulation du défaut

La figure 3.12 de la page 81 représente une simulation de cette ligne d'air avec le défaut de blocage du compresseur à l'occurrence $\mathrm{t}_n = 23$ secondes. Cette simulation est obtenue suivant le même profil de

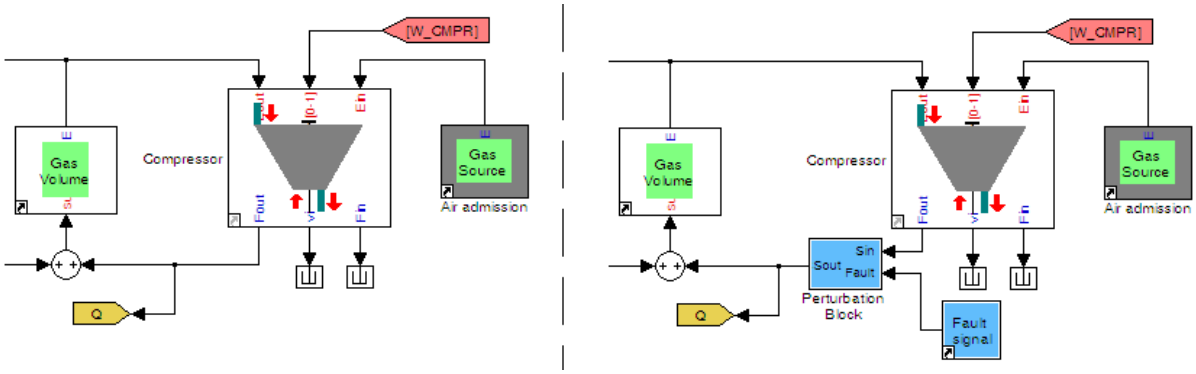

Figure 3.11 – Intégration du blocage du compresseur dans la ligne d'air.

consigne de débit donné au chapitre 2 lorsque nous présentions une simulation normale de cette ligne d'air. Les effets du blocage s'observent bien dans les quatre graphiques par le trait continu bleu. À partir de l'instant t = 23 secondes, la mesure réelle du débit chute brutalement à 0 gramme par seconde (premier graphique) et la mesure réelle de pression chute elle aussi à 1 bar (deuxième graphique). Les commandes du compresseur (troisième graphique) et de l'électrovanne (quatrième graphique) sont mises au maximum par le système de pilotage pour compenser cette chute.

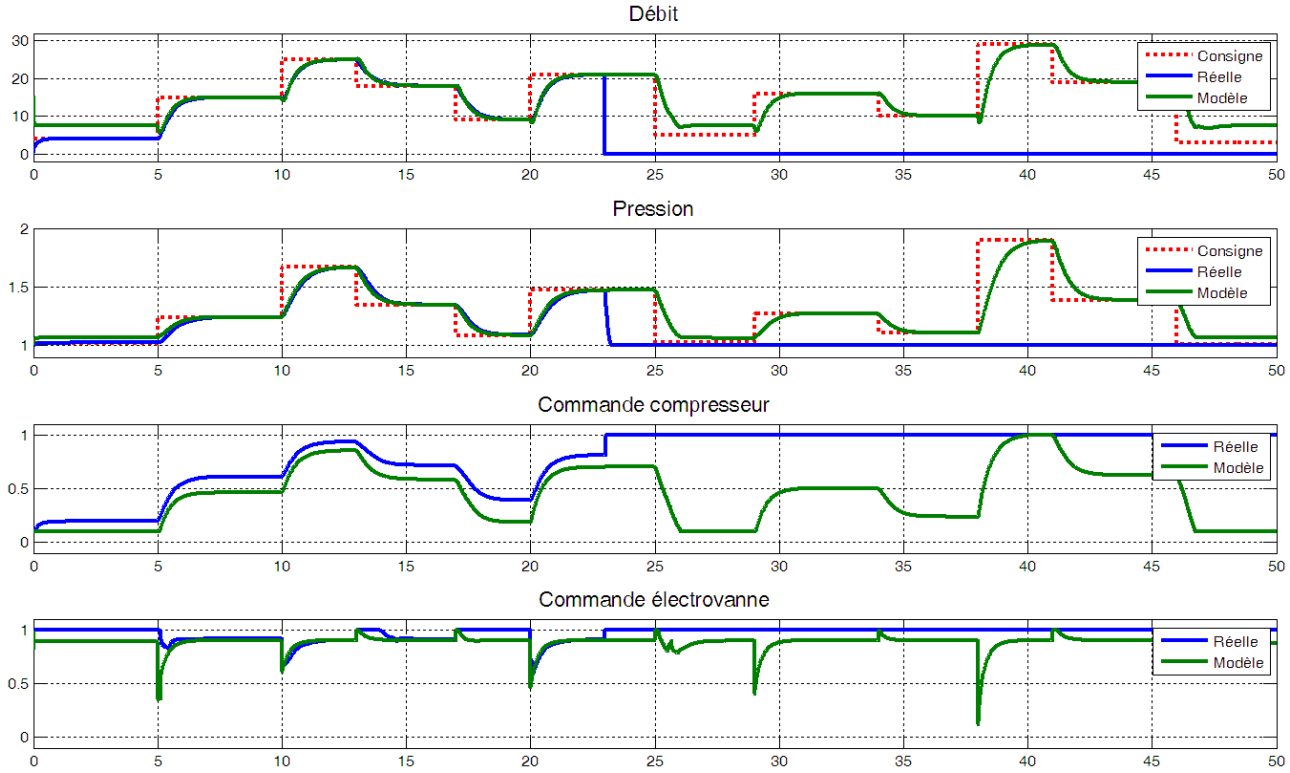

Figure 3.12 – Simulation de la ligne d'air avec le blocage du compresseur.

3.6.2.2 Encrassement du compresseur

Un encrassement du compresseur est dû à une accumulation de matières présentes dans l'air durant son fonctionnement normal. Ce défaut implique donc une baisse de régime de compresseur pour n'importe quelle valeur cible requise, et sa commande sera augmentée dans le but de répondre à cette

valeur cible. Le compresseur va donc fonctionner à plus fort régime pour obtenir les mêmes résultats, ce qui risque d'engendrer une usure plus rapide de celui-ci.

Remarquons que suivant le lieu d'exploitation du système, l'environnement extérieur peut être plus ou moins saturé en matières annexes. Ainsi un encrassement ne se fera pas suivant la même progression selon le lieu considéré. Néanmoins pour des utilisations en milieux fortement saturés, l'étude de sûreté de fonctionnement, lors de l'AMDEC, doit normalement solutionner ce cas d'encrassement en préconisant l'ajout d'un filtre en entrée de la ligne pour retenir ces matières annexes. Pour la suite, nous ne considérons que des encrassements dus à une utilisation normale : c'est-à-dire sur un lieu d'exploitation où l'environnement extérieur est normalement saturé en matières annexes.

Caractérisation du défaut

Un encrassement du compresseur provoque une baisse du régime de celui-ci pour n'importe quelle commande demandée. Le comportement et l'effet du défaut sont donc obtenus de ce constat.

Comportement : D'une manière générale, l'encrassement du compresseur apparaît progressivement, avec un coefficient d'augmentation très faible dans le temps, dès la mise en exploitation du système. Sa durée de présence est donc permanente. Les trois attributs du comportement du défaut $F_{DirtCmpr}$ sont donc :

- Occurrence : à l'instant initial $\mathrm{t}_n = 0$.
- Force d'apparition : progressive.
- Durée de présence : permanente.

Pour le coefficient a de progression, le comportement du défaut $F_{DirtCmpr}$ d'encrassement du compresseur est donné, pour tout $\mathrm{t} \in \mathbb{T}_{0.01}$, par l'équation suivante :

$$dft_{(F_{DirtCmpr},0)}(\mathrm{t}) = a \cdot \mathrm{t}$$

Remarquons que ce coefficient de progression a doit être très petit, de l'ordre de plusieurs jours à quelques mois. Si nous souhaitons par conséquent simuler ce défaut et en étudier son impact, il faut alors rajouter un biais : c'est-à-dire faire commencer la simulation avec le défaut déjà existant à un certain niveau et continuer la progression.

Effet : Ce défaut concerne le compresseur. Pour n'importe quelle commande demandée, le régime du compresseur est réduit et le débit d'air en sortie de ce compresseur est ainsi lui aussi réduit. En perturbant multiplicativement la variable de commande u_ω de la vitesse de rotation du compresseur, nous représentons ce défaut. La perturbation $u_{\omega_{F_{DirtCmpr}}}$ est donnée, pour tout $\mathrm{t} \in \mathbb{T}_{0.01}$, par l'équation suivante :

$$u_{\omega_{F_{DirtCmpr}}}(\mathrm{t}) = u_\omega(\mathrm{t}) \cdot (1 - dft_{(F_{DirtCmpr},0)}(\mathrm{t}))$$

Intégration du défaut

La figure 3.13 de la page 83 montre l'intégration de l'encrassement du compresseur dans le modèle MATLAB/Simulink© de la ligne d'air. La partie gauche représente ce compresseur sans l'intégration du défaut et la partie droite le représente avec l'intégration. Il s'agit d'une intégration externe. La variable u_ω de commande de la vitesse de rotation du compresseur est perturbée par un bloc *perturbation*, paramétré selon l'effet présenté juste avant, qui est contrôlé par un bloc *signal de défaut* paramétré selon le comportement $dft_{(F_{DirtCmpr},0)}$.

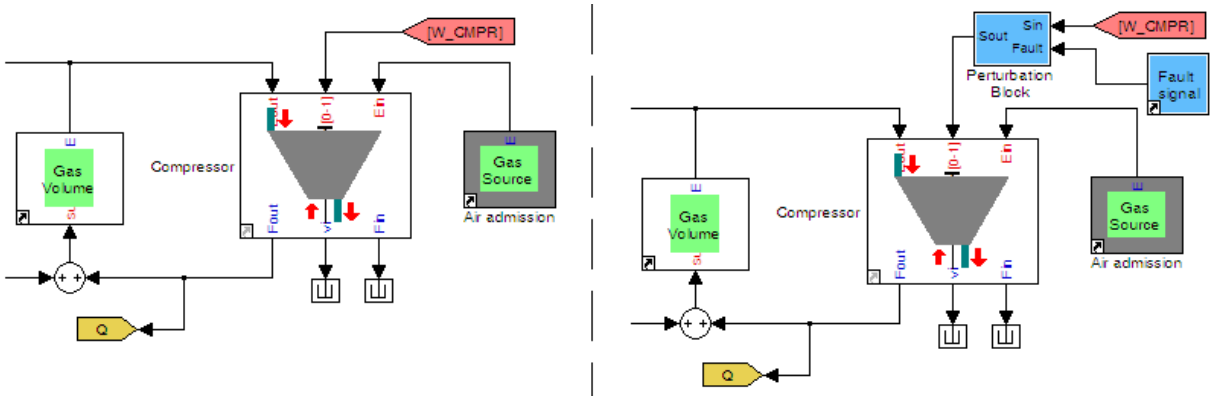

Figure 3.13 – Intégration de l'encrassement du compresseur dans la ligne d'air.

Simulation du défaut

Comme nous venons de l'expliquer, le coefficient de progression a du défaut doit être très petit, de l'ordre de plusieurs jours à quelques mois, et donc si nous souhaitons simuler ce défaut et en étudier son impact, il faut alors faire commencer la simulation avec le défaut déjà existant à un certain niveau et continuer la progression, ce qui signifie rajouter un biais au comportement du défaut. Cependant et comme les simulations que nous étudierons sont assez courtes, nous avons estimé qu'un comportement constant est équivalent au comportement faiblement progressif initialement introduit. Nous avons de ce fait considéré le comportement suivant pour tout $\mathrm{t} \in \mathbb{T}_{0.01}$:

$$dft_{(F_{DirtCmpr},0)}(\mathrm{t}) = 1$$

et la perturbation $u_{\omega_{F_{DirtCmpr}}}$ donnée, pour tout $\mathrm{t} \in \mathbb{T}_{0.01}$, par :

$$u_{\omega_{F_{DirtCmpr}}}(\mathrm{t}) = u_{\omega}(\mathrm{t}) \cdot (0.75 \cdot dft_{(F_{DirtCmpr},0)}(\mathrm{t}))$$

La figure 3.14 de la page 84 représente une simulation de cette ligne d'air avec le défaut d'encrassement du compresseur, toujours suivant le même profil de consigne de débit donné au chapitre 2. Nous observons bien que cette ligne d'air n'arrive pas à fournir les consignes de débit supérieures à 20 grammes par seconde. Les premier et troisième graphiques montrent bien que la mesure de débit est réduite par rapport à ce qui est demandé par la consigne. Cela signifie d'une part que pour les demandes inférieures à 20 grammes par seconde, cette réduction est compensée par une augmentation de la commande du compresseur. D'autre part pour des demandes supérieures à 20 grammes par seconde et comme la commande du compresseur est au maximum, ce débit fourni ne peut donc être supérieur. La mesure de la pression, visible dans le deuxième graphique, est elle aussi impactée par cette réduction et n'arrive donc pas elle non plus à suivre sa consigne de pression.

3.6.3 Défauts liés à l'électrovanne

Une électrovanne permet de réguler le passage du fluide (de l'hydrogène ou de l'air ou encore du liquide de refroidissement dans l'exemple du système pile à combustible) entre différents composants d'un système. Dans le cas de la ligne d'air, une électrovanne proportionnelle est utilisée. Elle permet une ouverture et une fermeture complètes ainsi que toutes les valeurs d'ouverture intermédiaires.

Comme pour un compresseur, une électrovanne proportionnelle est composée d'une partie mécanique et d'une partie électronique. La partie mécanique comprend la vanne physique (i.e. : le bloc fermé avec l'ailette d'ouverture), la bobine ou le moteur électrique ainsi que l'axe de transmission

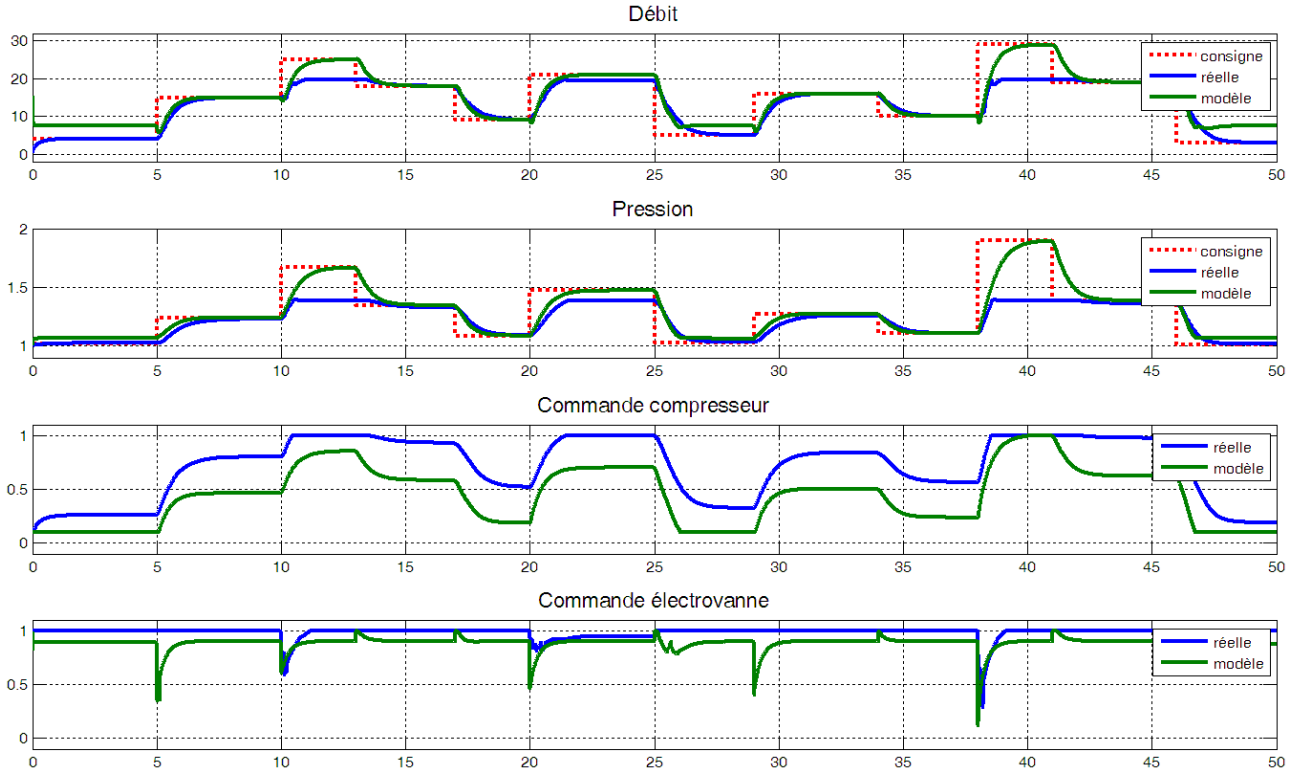

Figure 3.14 – Simulation de la ligne d'air avec l'encrassement du compresseur.

entre la bobine et l'ailette. La partie électronique comprend les cartes de puissance et de contrôle de l'électrovanne ainsi que les différents câbles de liaison (électrique et électronique).

Les défauts les plus courants sur des électrovannes sont des blocages (que nous notons F_{LockEV}) et des encrassements (que nous notons F_{DirtEV}). La figure 3.15 ci-dessous représente la localisation de ces deux défauts de l'électrovanne dans la ligne d'air.

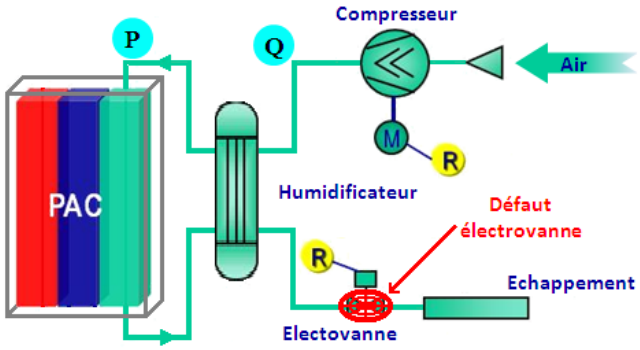

Figure 3.15 – Localisation de défauts de l'électrovanne dans la ligne d'air.

Remarquons que la littérature, qu'elle soit orientée diagnostic ou non, est abondante concernant ces défauts d'électrovannes. [Ise06] utilise d'ailleurs l'exemple d'une électrovanne pour exposer les défauts d'actionneurs. [BKLS03] traite de l'exemple du système à deux réservoirs reliés par deux électrovannes et y incorpore des défauts de blocage d'une des deux électrovannes.

3.6.3.1 Blocage de l'électrovanne

Pour les défauts F_{LockEV} de blocages de l'électrovanne, plusieurs types de blocages peuvent être observés selon que la partie mécanique ou électronique est en cause :

- les blocages dits de « butée » lorsque l'électrovanne est totalement ouverte ou totalement fermée et qu'elle se bloque à cette position,
- les blocages à n'importe quel angle d'ouverture,
- les blocages dus à une défaillance de la partie électronique et pour lesquels l'électrovanne revient à son angle initial d'ouverture (i.e. : soit totalement fermée ou soit totalement ouverte selon le cas) grâce à un ressort de rappel.

Nous avons choisi de ne traiter que les blocages F_{LockEV} de l'électrovanne à n'importe quel angle d'ouverture. Un tel blocage signifie que, quelle que soit la commande de l'angle d'ouverture demandée, l'électrovanne reste bloquée sur l'angle avant le blocage. Ceci implique que la pression de ligne n'est plus contrôlable ce qui peut donc causer une surpression dans la ligne avec comme conséquences un impact sur l'intégrité de la pile et même des détériorations des composants. Remarquons que les deux autres types de blocages sont facilement traitables par notre méthodologie.

Caractérisation du défaut

Nous venons d'expliquer qu'un tel blocage signifie que, quelle que soit la commande de l'angle d'ouverture demandée, l'électrovanne reste bloquée sur l'angle avant le blocage. Nous obtenons ainsi facilement le comportement du défaut et son effet.

Comportement : D'une manière générale et comme pour le blocage du compresseur, un blocage de l'électrovanne est un défaut apparaissant de manière brusque, à un moment aléatoire ou pouvant dépendre d'un évènement et dont la durée de présence est permanente ou transitoire voire intermittente. Nous n'allons considérer que le cas où ce blocage est permanent, ce que nous justifierons au chapitre suivant lors de l'étude de la diagnosticabilité. Les trois attributs du comportement du défaut F_{LockEV} sont donc :

- Occurrence : à un instant quelconque $\mathrm{t}_n \in \mathbb{T}_{0.01}$.
- Force d'apparition : brusque.
- Durée de présence : permanente.

Le comportement du défaut F_{LockEV} de blocage de l'électrovanne est donné, pour tout $\mathrm{t} \in \mathbb{T}_{0.01}$, par l'équation suivante :

$$dft_{(F_{LockEV},\mathrm{t}_n)}(\mathrm{t}) = \begin{cases} 0 & \text{si } \mathrm{t} \in [0;\mathrm{t}_n[\\ 1 & \text{si } \mathrm{t} \in \mathbb{T}_{0.01} \setminus [0;\mathrm{t}_n[\end{cases}$$

Effet : Ce défaut concerne l'électrovanne. Pour n'importe quelle commande de l'angle d'ouverture demandée, l'électrovanne reste bloquée sur l'angle avant le blocage. En bloquant la variable de commande u_x de l'angle d'ouverture de l'électrovanne à la valeur à l'instant t_n^-, nous représentons ce défaut. La perturbation $u_{x_{F_{LockEV}}}$ est donnée, pour tout $\mathrm{t} \in \mathbb{T}_{0.01}$, par l'équation suivante :

$$u_{x_{F_{LockEV}}}(\mathrm{t}) = \begin{cases} u_x(\mathrm{t}) & \text{si } dft_{(F_{LockEV},\mathrm{t}_n)}(\mathrm{t}) = 0 \\ u_x(\mathrm{t}_n^-) & \text{si } dft_{(F_{LockEV},\mathrm{t}_n)}(\mathrm{t}) = 1 \end{cases}$$

Intégration du défaut

La figure 3.16 suivante montre l'intégration du défaut de blocage de l'électrovanne dans le modèle MATLAB/Simulink© de la ligne d'air. La partie gauche représente cette électrovanne sans l'intégration du défaut et la partie droite la représente avec l'intégration. Il s'agit d'une intégration externe. La variable u_x de commande de l'angle d'ouverture de l'électrovanne est perturbée par un bloc *perturbation*, paramétré selon l'effet présenté, qui est contrôlé par un bloc *signal de défaut* paramétré selon le comportement $dft_{(F_{LockEV},\mathrm{t}_n)}$.

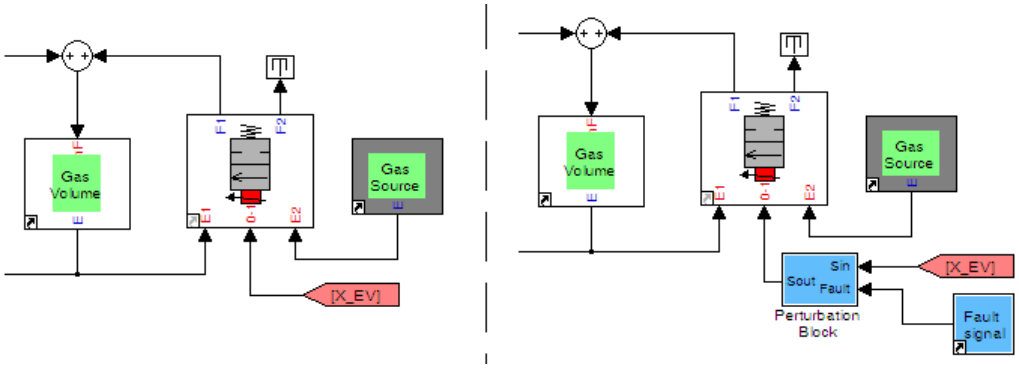

Figure 3.16 – Intégration du défaut de blocage de l'électrovanne dans la ligne d'air.

Simulation du défaut

La figure 3.17 de la page 87 représente une simulation de cette ligne d'air avec le défaut de blocage de l'électrovanne à l'occurrence $\mathrm{t}_n = 23$ secondes et toujours suivant le même profil de consigne de débit donné au chapitre 2. Très peu de choses sont remarquables : un retard dans la réponse du système concernant la mesure de pression au changement de consigne à l'instant t = 38 secondes, visible par le trait continu bleu du deuxième graphique, ainsi qu'une commande de l'électrovanne plus « incertaine » à partir de l'instant t = 29 secondes et visible par trait continu bleu du quatrième graphique.

3.6.3.2 Encrassement de l'électrovanne

Comme pour le compresseur, un défaut F_{DirtEV} d'encrassement de l'électrovanne est dû à une accumulation de matières annexes présentes dans l'air durant le fonctionnement normal de la ligne. Ce défaut implique une réduction de la section d'ouverture de l'électrovanne, ce qui risque d'augmenter la pression dans la ligne d'air et ainsi impacter la pile et détériorer des composants.

Bien sûr et comme nous l'avons déjà vu avec le compresseur, lorsque l'environnement extérieur du lieu d'exploitation du système est fortement saturé en matières annexes, un encrassement se fera assez rapidement comparé à une utilisation en milieu moins saturé. Néanmoins, ce type de défaut doit être pris en compte lors de l'étude de sûreté de fonctionnement par une solution adaptée préconisée lors de l'AMDEC : l'ajout d'un filtre en entrée de la ligne pour retenir ces matières annexes par exemple. Pour la suite et comme pour le compresseur, nous ne considérons que des encrassements dus à une utilisation normale : c'est-à-dire sur un lieu d'exploitation où l'environnement extérieur est normalement saturé en matières annexes.

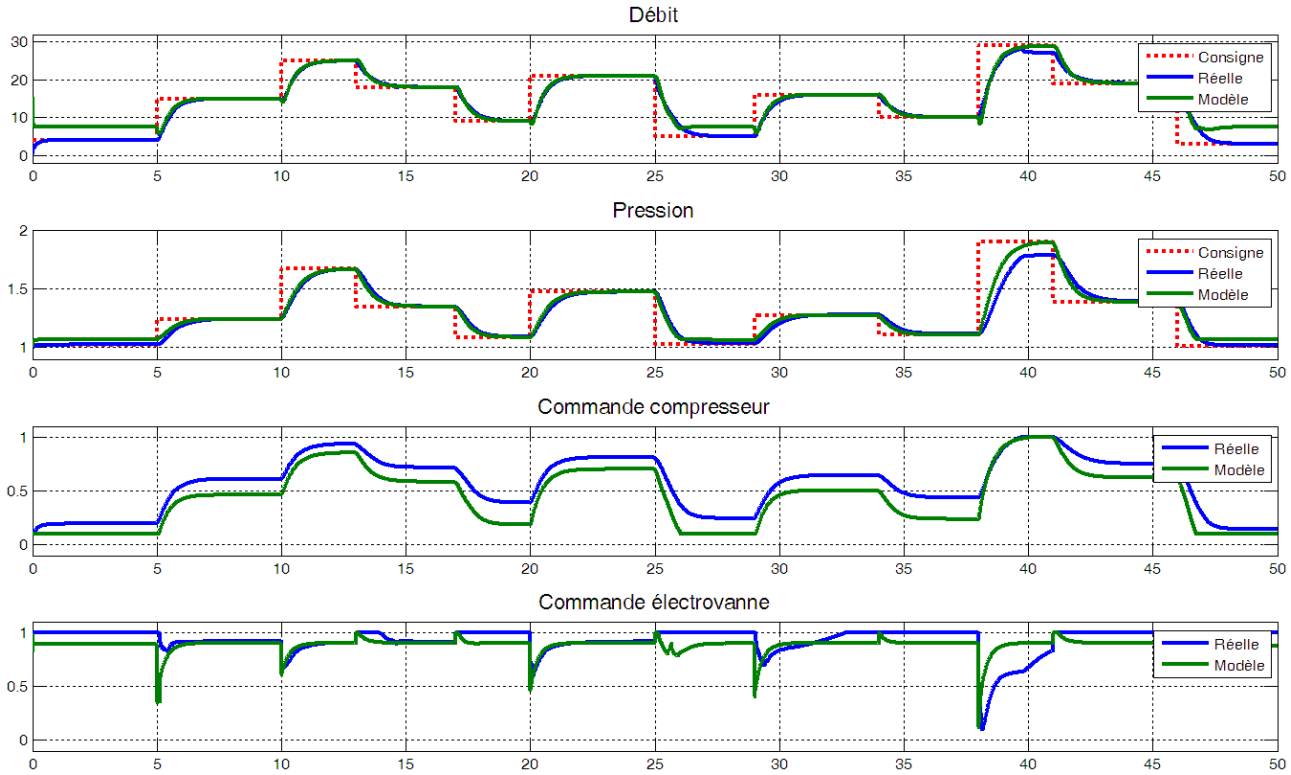

Figure 3.17 – Simulation de la ligne d'air avec le blocage de l'électrovanne.

Caractérisation du défaut

Un encrassement de l'électrovanne provoque une réduction progressive, avec un coefficient d'augmentation très faible dans le temps, de la section d'ouverture. Nous obtenons ainsi facilement le comportement du défaut et son effet.

Comportement : D'une manière générale et comme pour le défaut d'encrassement du compresseur, l'encrassement de l'électrovanne apparaît progressivement, avec un coefficient d'augmentation très faible dans le temps, dès la mise en exploitation du système. Sa durée de présence est donc permanente. Les trois attributs du comportement du défaut F_{DirtEV} sont donc :

- Occurrence : à l'instant initial $\mathrm{t}_n = 0$.
- Force d'apparition : progressive.
- Durée de présence : permanente.

Pour le coefficient a de progression, le comportement du défaut F_{DirtEV} d'encrassement de l'électrovanne est donné, pour tout $\mathrm{t} \in \mathbb{T}_{0.01}$, par l'équation suivante :

$$dft_{(F_{DirtEV},0)}(\mathrm{t}) = a \cdot \mathrm{t}$$

Remarquons que comme pour l'encrassement du compresseur, ce coefficient de progression a doit être très petit, de l'ordre de plusieurs jours à quelques mois, et qu'il faut, ici encore, rajouter un biais si nous souhaitons simuler ce défaut afin d'en étudier son impact.

Effet : Ce défaut concerne l'électrovanne, plus particulièrement la section d'ouverture qui se réduit progressivement. Le paramètre interne S_{EV} de section de l'électrovanne est par conséquent perturbé multiplicativement et sa perturbation $S_{EV_{F_{DirtEV}}}$ est donnée, pour tout $\mathrm{t} \in \mathbb{T}_{0.01}$, par l'équation suivante :

$$S_{EV_{F_{DirtEV}}}(\mathrm{t}) = S_{EV}(\mathrm{t}) \cdot (1 - max\{0; dft_{(F_{DirtEV},0)}(\mathrm{t})\})$$

Intégration du défaut

La figure 3.18 ci-dessous montre l'intégration du défaut d'encrassement de l'électrovanne dans le modèle MATLAB/Simulink© de la ligne d'air. La partie gauche représente cette électrovanne sans l'intégration du défaut et la partie droite la représente avec l'intégration. Il s'agit d'une intégration interne. Le paramètre interne S_{EV} de section de l'électrovanne est perturbé par l'ajout de différents composants, permettant de représenter la perturbation $S_{EV_{F_{DirtEV}}}$, et qui sont contrôlés par un bloc *signal de défaut* paramétré selon le comportement $dft_{(F_{DirtEV},0)}$.

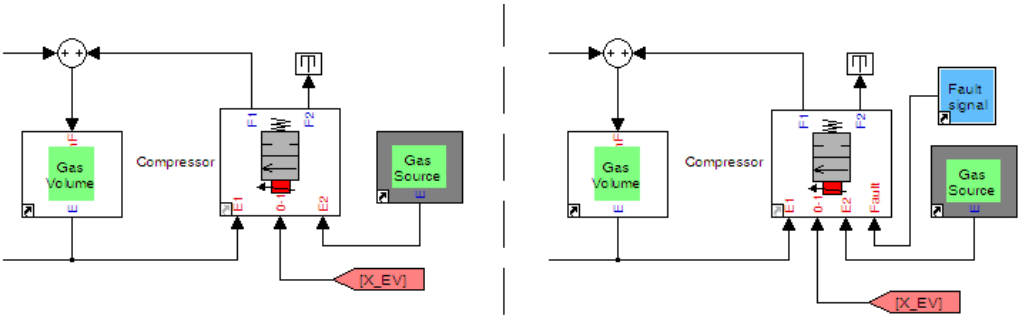

Figure 3.18 – Intégration du défaut d'encrassement de l'électrovanne dans la ligne d'air.

Simulation du défaut

Comme nous l'avons fait pour l'encrassement du compresseur, nous avons estimé qu'un comportement constant est équivalent au comportement faiblement progressif initialement introduit. Nous l'avons de ce fait obtenu en considérant le comportement suivant pour tout $\mathrm{t} \in \mathbb{T}_{0.01}$:

$$dft_{(F_{DirtEV},0)}(\mathrm{t}) = 1$$

et la perturbation $S_{EV_{F_{DirtEV}}}$ donnée, pour tout $\mathrm{t} \in \mathbb{T}_{0.01}$, par :

$$S_{EV_{F_{DirtEV}}}(\mathrm{t}) = S_{EV}(\mathrm{t}) \cdot (0.8 \cdot dft_{(F_{DirtEV},0)}(\mathrm{t}))$$

La figure 3.19 de la page 89 représente une simulation de cette ligne d'air avec le défaut d'encrassement de l'électrovanne. Cette simulation est toujours obtenue suivant le même profil de consigne de débit. Il s'agit d'un encrassement de l'électrovanne à 20%. Nous pouvons remarquer que la mesure réelle de pression est beaucoup plus élevée que normalement par rapport à la consigne (trait continu bleu du second graphique), et que la commande de l'électrovanne est presque tout le temps totalement ouverte (trait continu bleu du quatrième graphique) pour compenser cet encrassement.

3.6.4 Fuite d'air dans la tuyauterie

Une tuyauterie sert à acheminer du fluide (de l'hydrogène ou de l'air ou encore du liquide de refroidissement dans l'exemple du système pile à combustible) entre différents composants d'un système. Pour la ligne d'air, d'une part la tuyauterie sert à acheminer l'air du compresseur vers la pile, en passant par l'humidificateur, et d'autre part, après consommation de l'oxygène de l'air par la pile, elle sert à évacuer cet air vers l'extérieur par l'échappement. Une tuyauterie, entre deux composants

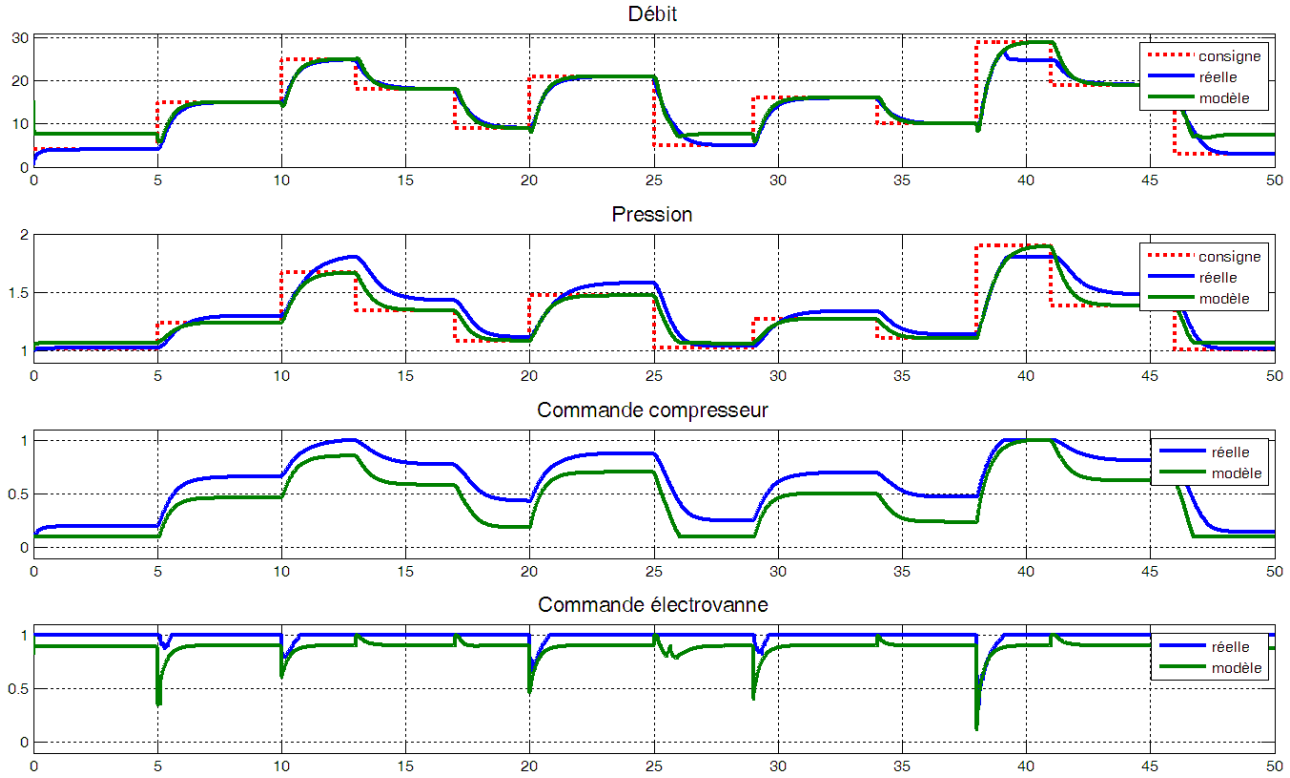

Figure 3.19 – Simulation de la ligne d'air avec l'encrassement de l'électrovanne.

(le compresseur et l'humidificateur par exemple), est composé d'un tuyau et de joints de liaison entre chaque bout du tuyau et le composant relié.

Selon l'étude de sûreté de fonctionnement menée sur le système pile à combustible, une fuite dans la ligne d'hydrogène n'est pas tolérée car cela peut conduire à une explosion comme l'indique [ASC01] : les caractéristiques chimiques de l'hydrogène le rendent approprié aux fuites. Pour la ligne de refroidissement, une fuite peut provoquer une surchauffe de la pile et ainsi impacter son intégrité. Enfin pour la ligne d'air, une fuite peut impacter les performances du système et provoquer des baisses de rendement ainsi que la détérioration de certains composants, notamment le compresseur qui risque de fonctionner à un régime plus élevé pour compenser les pertes d'air dues à cette fuite.

Une fuite dans la tuyauterie représente une ouverture vers l'extérieur. Elle peut être due soit à une usure (normale ou anormale suite à un mauvais composant ou un mauvais assemblage) d'un joint, soit à une sur-pression dans la ligne causée par un fonctionnement à trop haut régime du compresseur, ou encore soit à une rupture de la tuyauterie, causée par un choc externe par exemple. Ces trois causes se caractérisent de la même manière concernant l'effet du défaut, seul le comportement change. Néanmoins, remarquons d'une part qu'une fuite due à une sur-pression dans la ligne, causée par un fonctionnement à trop haut régime du compresseur, est fort peu probable. En effet, celui-ci a normalement été conçu ou choisi selon des caractéristiques, notamment les puissances nominale et maximale, compatibles avec les exigences de bon fonctionnement de la ligne. D'autre part une fuite due à une rupture, causée par un choc externe par exemple, implique que ce choc externe aura sûrement impacté d'autres composants du système. Enfin pour les fuites dues à l'usure d'un joint, qu'elles soient normales ou anormales suite à un mauvais composant ou mauvais assemblage, elles auront le même comportement de défaut : elles apparaîtront progressivement mais avec une constante de progression plus faible pour l'usure normale.

Pour la suite, nous avons donc choisi de traiter un défaut de fuite d'air $F_{LeakAir}$ entre le compresseur et l'humidificateur due à une usure normale et qui occasionne ainsi une ouverture dans la tuyauterie vers l'extérieur de plus en plus importante. La figure 3.20 ci-dessous représente la localisation d'une telle fuite entre le compresseur et l'humidificateur.

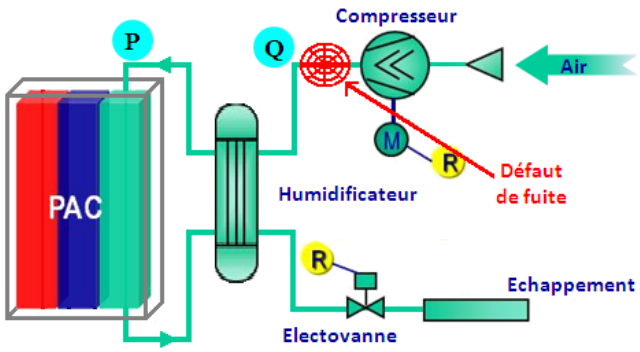

Figure 3.20 – Localisation du défaut de fuite.

Caractérisation du défaut

Lors d'une fuite, la tuyauterie (i.e. : le tuyau ou le joint de liaison entre le tuyau et un composant) est pourvue d'une ouverture vers l'extérieur. Pour une fuite due à une usure normale, l'ouverture se fait de plus en plus importante en commençant à nulle (i.e. : il n'y a pas d'ouverture). Bien que le comportement d'un tel défaut va ainsi être facilement obtenu, il faudra réaliser beaucoup plus de travail pour son effet, notamment pour son intégration.

Comportement : Une fuite due à une usure normale est un défaut apparaissant de manière progressive (avec un coefficient d'augmentation très faible dans le temps) dès la mise en exploitation du système. Sa durée de présence est donc permanente. Les trois attributs du comportement du défaut $F_{LeakAir}$ de fuite sont donc :

- Occurrence : à l'instant initial $\mathrm{t}_n = 0$.
- Force d'apparition : progressive.
- Durée de présence : permanente.

Pour le coefficient a de progression, le comportement du défaut $F_{LeakAir}$ de fuite est donné, pour tout $\mathrm{t} \in \mathbb{T}_{0.01}$, par l'équation suivante :

$$dft_{(F_{LeakAir},0)}(\mathrm{t}) = a \cdot \mathrm{t}$$

Remarquons ici encore que comme pour les encrassements du compresseur ou de l'électrovanne, ce coefficient de progression a doit être très petit, et qu'il faudra donc rajouter un biais si nous souhaitons simuler ce défaut et en étudier son impact.

Effet : Ce défaut de fuite concerne la tuyauterie entre le compresseur et l'humidificateur. Nous avons dit qu'une fuite provoque une ouverture de la tuyauterie vers l'extérieur. Une manière de représenter réellement ce défaut de fuite est d'ajouter sur le tuyau une vanne s'ouvrant vers l'extérieur. Il va donc y avoir une perte de charge additionnelle en amont de la ligne. Nous n'allons pas décrire complètement l'effet de ces défauts dans les équations du modèle car nous n'avons justement pas décrit ces équations. Cependant, cette perte de charge va impliquer des perturbations sur certaines variables de la ligne (débit, pression, taux d'humidité, température, etc.).

Intégration du défaut

La figure 3.21 suivante montre l'intégration de ce défaut de fuite entre le compresseur et l'humidificateur dans le modèle MATLAB/Simulink© de la ligne d'air. La partie gauche représente la tuyauterie entre le compresseur et l'humidificateur sans l'intégration du défaut et la partie droite la représente avec l'intégration. Il s'agit d'une intégration par ajout de composants. Une nouvelle électrovanne, contrôlée par un bloc *signal de défaut* paramétré selon le comportement $dft_{(F_{LeakAir},\mathrm{t}_n)}$, ainsi qu'un bloc « source de fluide » représentant l'ouverture d'air vers l'extérieur, comme le même bloc représentant l'échappement de la ligne, ont été ajoutés.

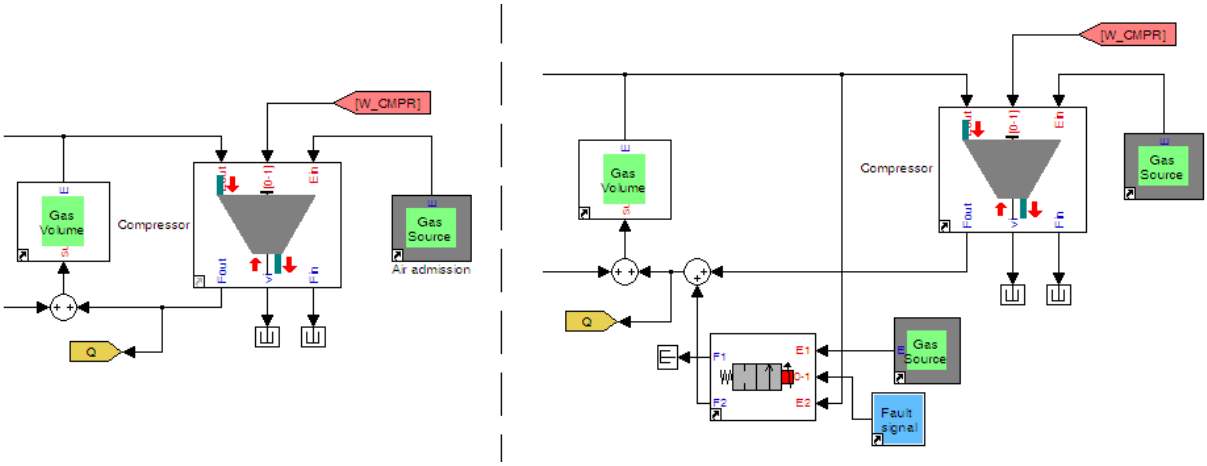

Figure 3.21 – Intégration du défaut de fuite d'air dans la tuyauterie.

Simulation de la fuite

Ici encore et de la même manière que pour les encrassements du compresseur et de l'électrovanne, nous avons estimé qu'un comportement constant est équivalent au comportement faiblement progressif initialement introduit. Nous l'avons de ce fait obtenu en considérant le comportement suivant pour tout $\mathrm{t} \in \mathbb{T}_{0.01}$:

$$dft_{(F_{LeakAir},0)}(\mathrm{t}) = 1$$

La figure 3.22 de la page 92 représente une simulation de cette ligne d'air avec ce défaut de fuite, toujours suivant le profil de consigne de débit donné au chapitre 2. Très peu de choses sont remarquables sur ce graphique. Remarquons néanmoins que sur la fenêtre temporelle $[38; 41]$, où les consignes de débit et de pression sont élevées, les mesures réelles de débit et de pression sont beaucoup moins élevées que normalement (traits continus bleus des premier et deuxième graphiques). Concernant les commandes réelles du compresseur et de l'électrovanne, seule une comparaison « image à image » entre le fonctionnement normal et ce fonctionnement avec défaut pourrait nous permettre de remarquer des différences.

3.6.5 Défauts liés aux capteurs

Un capteur est un dispositif transformant l'état d'une grandeur physique observée dans un système en une grandeur utilisable (i.e. : une information manipulable). Un capteur prélève donc une information sur le comportement du processus physique du système opérant et la transforme en une information exploitable par le système de pilotage. Dans le cas de la ligne d'air, deux capteurs sont

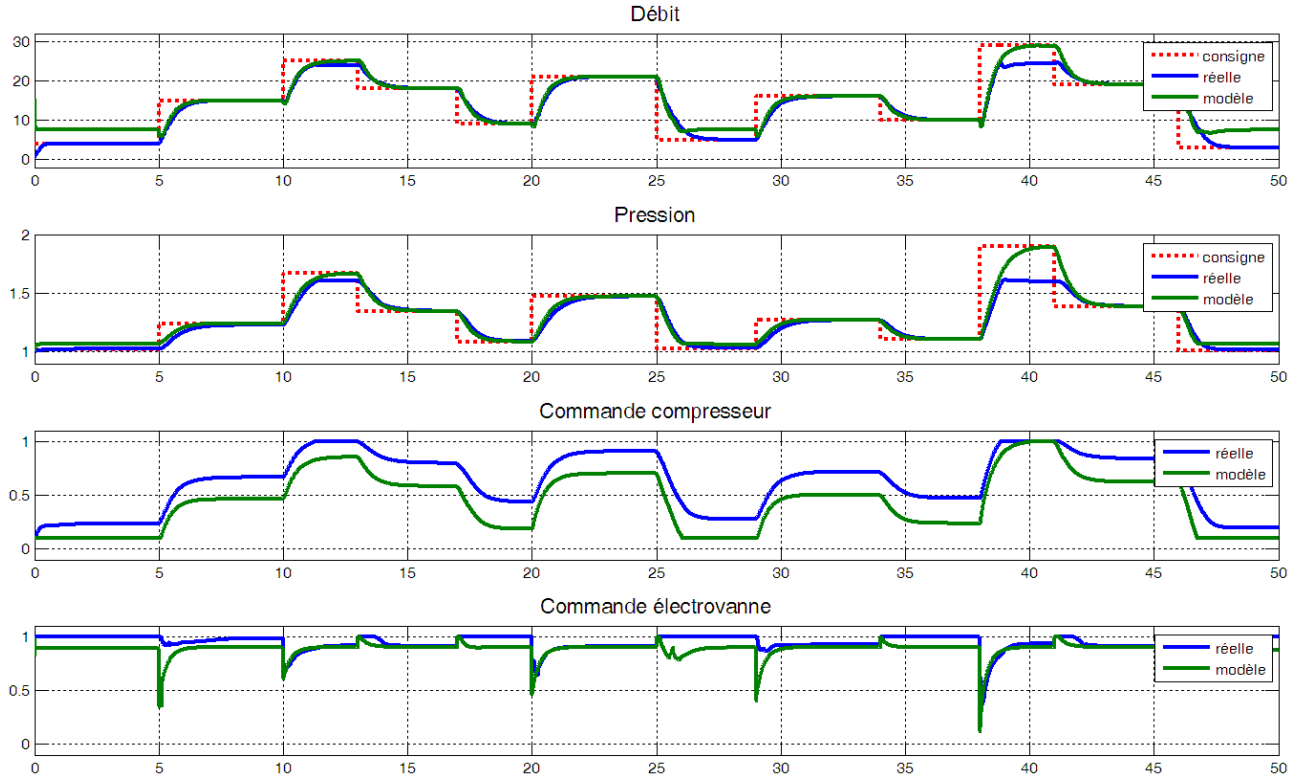

Figure 3.22 – Simulation de la ligne d'air avec la fuite due à l'usure normale dans la tuyauterie.

utilisés : un capteur de pression situé en amont de la pile et mesurant la pression de l'air fourni à la pile, un capteur de débit situé en sortie du compresseur et mesurant le débit d'air envoyé dans la ligne. Nous ne rentrons pas en détail dans la composition d'un capteur.

Les défauts de capteurs ont été abondamment étudiés que ce soit dans la littérature scientifique mais aussi par les industriels. [Ise06] y consacre une partie dans laquelle sont traités la modélisation d'un capteur, les défauts potentiels et leurs modélisations ainsi que l'application des méthodes de diagnostic sur le traitement de ces défauts. Les industriels ont, eux aussi, développé des règles et outils de traitement de défauts de capteurs.

Du fait de l'abondante littérature traitant de ce type de défaut, nous avons choisi de ne traiter que le cas de défauts basiques F_{SenQ} et F_{SenP} de mesure des capteurs de débit et de pressions de la ligne d'air. Il s'agit de ruptures de ces capteurs entraînant des mesures permanentes à nulle (i.e. : 0 gramme par seconde pour le capteur de débit et 0 bar pour le capteur de pression). Lors des occurrences de ces défauts, le pilotage du compresseur et de l'électrovanne peut être modifié et engendrer ainsi des risques pour l'intégrité de la pile et même des détériorations des différents actionneurs : pour compenser ces pertes de mesure, le compresseur peut être mis au maximum, ou l'électrovanne peut être complètement fermée. La figure 3.23 de la page 93 représente la localisation dans la ligne d'air de ces deux défauts de capteurs

3.6.5.1 Défauts de mesure du capteur de débit

Une rupture du capteur de débit entraîne une brusque chute de la mesure à nulle (i.e. : 0 gramme par seconde). Cette mesure à nulle restant permanente.

Caractérisation du défaut

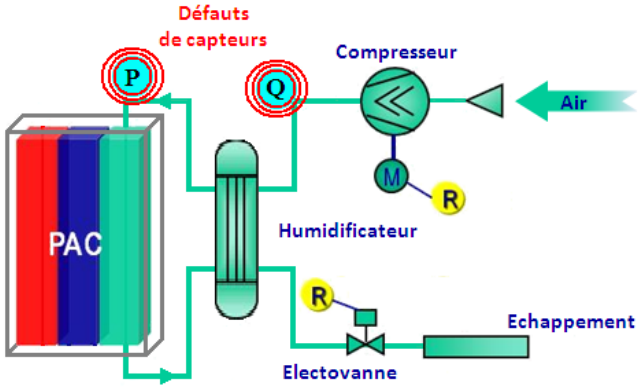

Figure 3.23 – Localisation des défauts de mesure des capteurs.

Suivant ce que nous venons de présenter, nous obtenons facilement le comportement de ce défaut F_{SenQ} et son effet.

Comportements : La rupture du capteur de débit est un défaut apparaissant de manière brusque, à un moment aléatoire ou pouvant dépendre d'un évènement, et dont la durée de présence est permanente. Les trois attributs du comportement du défaut F_{SenQ} sont donc :

- Occurrence : à un instant quelconque $\mathrm{t}_n \in \mathbb{T}_{0.01}$.
- Force d'apparition : brusque.
- Durée de présence : permanente.

Le comportement du défaut F_{SenQ} de rupture du capteur de débit est donné, pour tout $\mathrm{t} \in \mathbb{T}_{0.01}$, par l'équation suivante :

$$dft_{(F_{SenQ},\mathrm{t}_n)}(\mathrm{t}) = \begin{cases} 0 & \text{si } \mathrm{t} \in [0;\mathrm{t}_n[\\ 1 & \text{si } \mathrm{t} \in \mathbb{T}_{0.01} \setminus [0;\mathrm{t}_n[\end{cases}$$

Effets : Ce défaut concerne le capteur de débit, plus particulièrement la mesure de débit fournie par ce capteur. La variable y_Q de mesure du débit d'air est par conséquent perturbée multiplicativement et sa perturbation $y_{Q_{F_{SenQ}}}$ est donnée, pour tout $\mathrm{t} \in \mathbb{T}_{0.01}$, par l'équation suivante :

$$y_{Q_{F_{SenQ}}}(\mathrm{t}) = y_Q(\mathrm{t}) \cdot (1 - dft_{(F_{SenQ},\mathrm{t}_n)}(\mathrm{t}))$$

Intégration du défaut de mesure du capteur de débit

La figure 3.24 de la page 94 montre l'intégration du défaut de mesure du capteur de débit dans le modèle MATLAB/Simulink© de la ligne d'air. La partie gauche représente ce capteur de mesure sans l'intégration du défaut et la partie droite le représente avec l'intégration. Il s'agit d'une intégration externe. La variable y_Q de mesure du débit d'air est perturbée par un bloc *perturbation*, paramétré selon l'effet présenté avant, qui est contrôlé par un bloc *signal de défaut* paramétré selon le comportement $dft_{(F_{SenQ},\mathrm{t}_n)}$.

Simulation du défaut de mesure du capteur de débit

La figure 3.25 suivante de la page 94 représente une simulation de cette ligne d'air, toujours obtenue suivant le même profil de consigne de débit, avec le défaut de mesure du capteur de débit à l'occurrence $\mathrm{t}_n = 23$ secondes. Les traits continus bleus des quatre graphiques montrent clairement

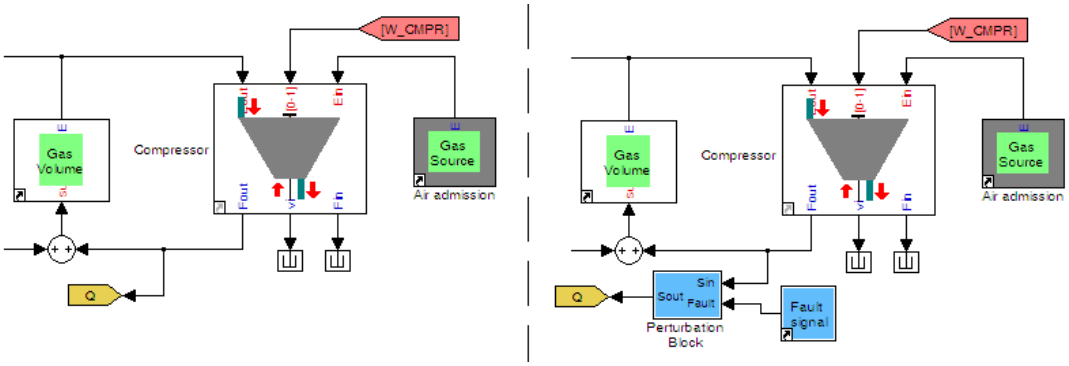

Figure 3.24 – Intégration du défaut de mesure du capteur de débit dans la ligne d'air.

les perturbations engendrées par ce défaut à partir de l'occurrence $t_n = 23$ secondes. La mesure réelle du débit chute brusquement à 0 gramme par seconde (premier graphique) et la mesure réelle de pression augmente fortement à une valeur d'environ 1.6 bar (deuxième graphique). Les commandes du compresseur (troisième graphique) et de l'électrovanne (quatrième graphique) sont mises au maximum par le système de pilotage pour compenser cette chute de la mesure.

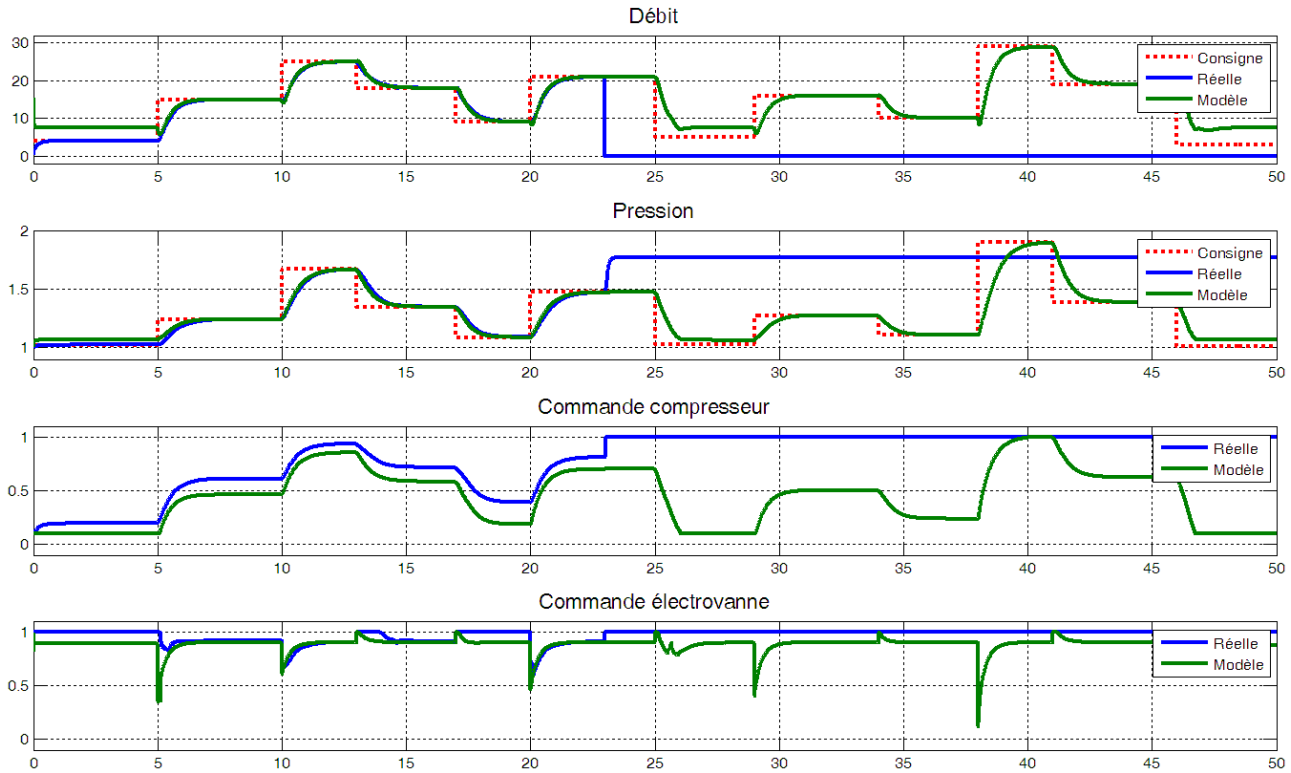

Figure 3.25 – Simulation de la ligne d'air avec le défaut de mesure du capteur de débit.

3.6.5.2 Défauts de mesure du capteur de pression

Une rupture du capteur de pression entraîne une brusque chute de la mesure à nulle (0 bar) ; cette mesure à nulle restant permanente.

Caractérisation du défaut

Nous obtenons facilement, ici encore et suivant ce que nous venons de présenter, le comportement de ce défaut F_{SenP} et son effet.

Comportements : La rupture du capteur de pression est un défaut apparaissant de manière brusque, à un moment aléatoire ou pouvant dépendre d'un évènement, et dont la durée de présence est permanente. Les trois attributs du comportement des défauts F_{SenP} sont donc :

- Occurrence : à un instant quelconque $\mathrm{t}_n \in \mathbb{T}_{0.01}$.
- Force d'apparition : brusque.
- Durée de présence : permanente.

Le comportement du défaut F_{SenP} de rupture du capteur de pression est donné, pour tout $\mathrm{t} \in \mathbb{T}_{0.01}$, par l'équation suivante :

$$dft_{(F_{SenP},\mathrm{t}_n)}(\mathrm{t}) = \begin{cases} 0 & \text{si } \mathrm{t} \in [0;\mathrm{t}_n[\\ 1 & \text{si } \mathrm{t} \in \mathbb{T}_{0.01} \setminus [0;\mathrm{t}_n[\end{cases}$$

Effets : Ce défaut concerne le capteur de pression, plus particulièrement la mesure de pression fournie par ce capteur. La variable y_P de mesure de la pression de l'air est par conséquent perturbée multiplicativement et sa perturbation $y_{P_{F_{SenP}}}$ est donnée, pour tout $\mathrm{t} \in \mathbb{T}_{0.01}$, par l'équation suivante :

$$y_{P_{F_{SenP}}}(\mathrm{t}) = y_P(\mathrm{t}) \cdot (1 - dft_{(F_{SenP},\mathrm{t}_n)}(\mathrm{t}))$$

Intégration du défaut de mesure du capteur de pression

La figure 3.26 ci-dessous montre l'intégration du défaut de mesure du capteur de pression dans le modèle MATLAB/Simulink© de la ligne d'air. La partie gauche représente ce capteur de mesure sans l'intégration du défaut et la partie droite le représente avec l'intégration. Il s'agit d'une intégration externe. La variable y_P de mesure de la pression d'air est perturbée par un bloc *perturbation*, paramétré selon l'effet présenté juste avant, qui est contrôlé par un bloc *signal de défaut* paramétré selon le comportement $dft_{(F_{SenP},\mathrm{t}_n)}$.

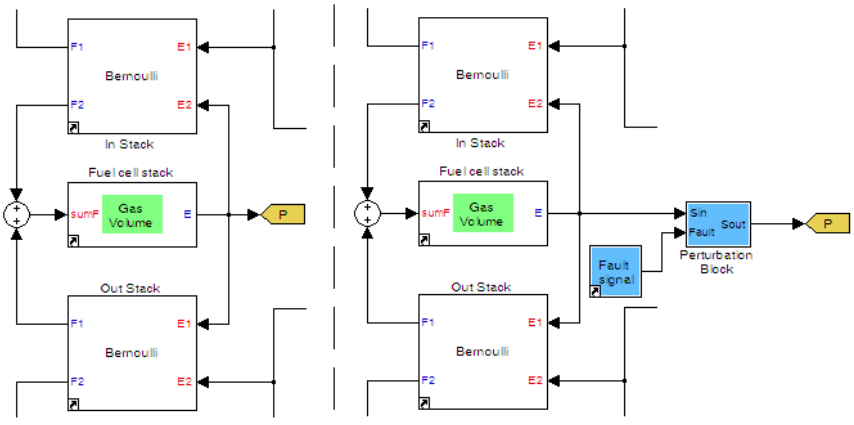

Figure 3.26 – Intégration du défaut de mesure du capteur de pression dans la ligne d'air.

Simulation du défaut de mesure du capteur de pression

La figure 3.27 de la page 96 représente une simulation de cette ligne d'air avec le défaut de mesure

du capteur de pression à l'occurrence $t_n = 23$ secondes et obtenue suivant toujours le même profil de consigne de débit. À partir de cette occurrence $t_n = 23$ secondes du défaut, nous observons très bien par le trait continu bleu du deuxième graphique que la mesure réelle de la pression chute brusquement à 0 bar. Les commandes du compresseur et de l'électrovanne (traits bleus des troisième et quatrième graphiques) sont elles aussi modifiées afin de compenser cette chute. Enfin, remarquons des perturbations de la mesure de débit (trait bleu du premier graphique) à cette occurrence $t_n = 23$ secondes ainsi que pour des consignes élevées, visibles sur la fenêtre temporelle $[38; 41[$.

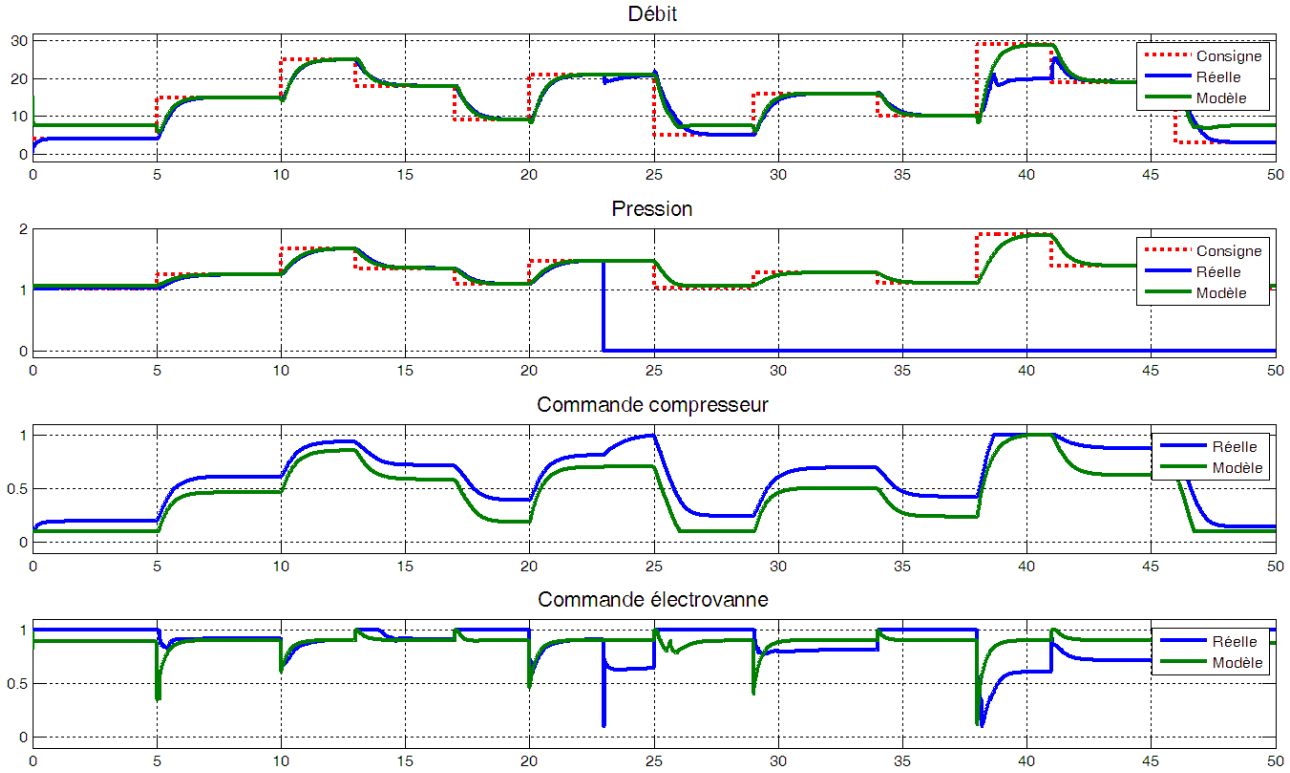

Figure 3.27 – Simulation de la ligne d'air avec le défaut de mesure du capteur de pression.

3.6.6 Défauts considérés par le diagnostiqueur

Nous venons de présenter un ensemble très varié de défauts potentiels du système. Cependant et comme nous l'avons expliqué autant au chapitre 1 d'introduction aux méthodologies de diagnostic que dans la partie précédente présentant la sûreté de fonctionnement, tous les défauts potentiels ne sont pas à prendre en compte suivant la même approche. Comme l'indique [BJL+90], les défauts évolutifs doivent être surveillés de façon périodique et à rythme adapté au processus de vieillissement pour permettre soit une réadaptation des fonctions de commande, soit une action de maintenance préventive appropriée. Ainsi, des usures seront donc à prendre en compte lors des phases de maintenance du système et d'autres défauts auront leurs propres modules de surveillance (comme les fuites d'hydrogène par exemple).

Pour la suite, nous allons malgré tout considérer tous ces défauts présentés afin de montrer que la théorie d'étude de la diagnosticabilité, que nous allons présenter au chapitre 4 suivant, est applicable pour tous les défauts, avec néanmoins de mauvais résultats pour les défauts faiblement progressifs. De bons résultats seront par contre obtenus pour les défauts brusques et permanents ayant, comme nous l'avons déjà expliqué, les plus graves conséquences sur l'intégrité du système. Gardons donc à l'esprit la remarque de [BJL+90] concernant les défauts évolutifs.

L'ensemble $\Gamma_{AirLine}$ des défauts à considérer va donc être :

$$\Gamma_{AirLine} = \{F_{norm}; F_{LockCmpr}; F_{DirtCmpr}; F_{LockEV}; F_{DirtEV}; F_{LeakAir}; F_{SenQ}; F_{SenP}\}$$

3.7 Conclusion sur la typologie des défauts

Dans ce chapitre, nous venons de présenter une méthodologie permettant de modéliser les défauts potentiels d'un système piloté. Suite à l'identification, grâce à une étude de sûreté de fonctionnement du système, de ceux devant être pris en compte par un diagnostiqueur, nous avons pu définir une typologie des défauts. Cette typologie décrit les défauts suivant deux traits caractéristiques génériques, leurs comportements ainsi que leurs effets sur le système, et permet ainsi de les intégrer dans le modèle de bon fonctionnement du système par modification des équations de fonctionnement normal. Nous avons par ailleurs construit une bibliothèque de défaut, implémentée dans l'outil de simulation MATLAB/Simulink©, qui nous a permis d'intégrer des défauts potentiels dans un modèle de simulation. Tous ces concepts furent enfin mis en application sur le cas d'étude.

Grâce à tous ces modèles de fonctionnement, le fonctionnement normal et les fonctionnements anormaux, il va nous être possible, dans les chapitres qui vont suivre, de mener l'étude de la diagnosticabilité de ces défauts puis de générer le diagnostiqueur associé à cette étude.

Chapitre 4

L'étude de la diagnosticabilité du système

Suite à la description d'un système piloté et à l'intégration des défauts potentiels du système dans son modèle de bon fonctionnement, présentées aux chapitres 2 et 3 précédents, nous pouvons maintenant mener l'étude de la diagnosticabilité du système. Il va s'agir de s'assurer qu'en fonctionnement, le diagnostiqueur sera toujours capable de diagnostiquer les défauts potentiels préalablement répertoriés. C'est-à-dire qu'il sera toujours capable d'une part de détecter n'importe lequel de ces défauts répertoriés lorsqu'il apparaît et d'autre part de l'identifier (i.e. : l'isoler) sans aucune ambiguïté par rapport aux autres défauts répertoriés.

Ce chapitre, va donc présenter cette notion de diagnosticabilité du système comme nous l'avons introduit dans [BFRD11] et [BDRF11a]. Il s'agira de la diagnosticabilité de chacun des défauts potentiels du système qui ont été répertoriés lors de l'étude de sûreté de fonctionnement. En commençant par des préliminaires abordant certaines contraintes industrielles ainsi que décrivant intuitivement cette notion de diagnosticabilité, nous serons ensuite amenés à considérer les comportements observables du système en fonctionnement normal ou sous la présence d'un défaut. Il va s'agir de l'évolution des valeurs des variables observables du système suivant différentes suites de valeurs de la consigne de l'opérateur. Ces comportements observables vont ensuite nous permettre de déterminer des propriétés caractéristiques du fonctionnement du système sous la présence de chacun des défauts, y compris le cas normal (i.e. : le fonctionnement normal du système). Ce sera à partir de ces propriétés caractéristiques, qui décriront les règles d'analyse du diagnostiqueur, que va se mener l'étude de la diagnosticabilité. Enfin, nous appliquerons toute cette partie théorique sur le cas d'étude.

4.1 Préliminaires

Nous allons présenter certaines contraintes industrielles qui vont nous obliger à faire des hypothèses préalables sur le nombre et les occurrences des différents défauts répertoriés. Nous pourrons ensuite présenter de manière intuitive cette notion de diagnosticabilité du système, qui va nous permettre de définir un ensemble de paramètres permettant sa définition formelle.

4.1.1 Notations diverses

Avant toute chose et comme nous en aurons besoin dans la suite de ce chapitre, nous introduisons deux notations. Pour un vecteur $v = (v_1, \ldots, v_n)$ et pour un indice $i \in \{1; \ldots; n\}$, le i-ème élément de v est noté $p_i(v) = v_i$. Pour un produit d'ensembles $E = E_1 \times \cdots \times E_n$, pour un sous-ensemble $G \subseteq E$ et pour des indices $i_1, \ldots, i_k \in \{1; \ldots; n\}$ avec $k \leq n$, la projection des éléments de G sur $E_{i_1} \times \cdots \times E_{i_k}$ est l'ensemble $\mathrm{Pr}_{E_{i_1} \times \cdots \times E_{i_k}}(G) = \{(p_{i_1}(v), \ldots, p_{i_k}(v)) \in E_{i_1} \times \cdots \times E_{i_k} / v \in G\}$. Cette seconde notion

nous sera utile lors de la présentation des comportements observables du système, qui seront définis comme des projections de comportements du système uniquement sur les variables observables.

4.1.2 Contraintes industrielles

En toute généralité, un ou plusieurs défauts peuvent apparaître en même temps ou dans un intervalle temporel relativement court. Cependant pour un fonctionnement réel en industrie, il est nécessaire d'une part de détecter le premier défaut qui apparaît et d'autre part que le diagnostic (i.e. : la détection et l'isolation) d'un défaut se fasse en un temps relativement rapide et borné ; ce temps maximum entre l'occurrence d'un défaut et son diagnostic sera exprimé par une certaine borne temporelle b.

De plus et pour des questions de complexité, bien que nous verrons en conclusion comment réaliser cette étude, nous ne traitons pas le cas des défauts multiples : c'est-à-dire le cas où plusieurs défauts apparaissent en même temps ou dans un intervalle temporel très court. Nous supposerons donc toujours que l'intervalle de temps entre les occurrences de deux défauts est supérieur à la borne b de diagnostic. Comme nous souhaitons détecter le premier défaut qui apparaît, cela revient donc à ne traiter que le cas où un seul défaut apparaît.

Par ailleurs et pour la suite du document, nous ne considérons que des défauts apparaissant à une occurrence particulière $\mathrm{t}_n \in \mathbb{T}_\iota$ du temps. En effet, notre but étant de nous assurer qu'un défaut soit diagnosticable lorsqu'il apparaît, nous ne cherchons donc pas à prédire son occurrence. Nous définirons par conséquent nous-même les occurrences potentielles de chacun des défauts.

4.1.3 Description intuitive de la diagnosticabilité du système

Comme nous venons de l'indiquer, l'étude de la diagnosticabilité du système consiste à s'assurer qu'en fonctionnement, le diagnostiqueur sera toujours capable de détecter n'importe quel défaut lorsqu'il apparaît, mais en plus de l'isoler sans aucune ambiguïté par rapport aux autres défauts. Tous ces défauts considérés étant bien sûr ceux préalablement répertoriés lors de l'étude de sûreté de fonctionnement. Il s'agit donc de s'assurer que chacun de ces défauts répertoriés sera bien diagnostiqué par le diagnostiqueur.

4.1.3.1 Notion intuitive de la diagnosticabilité du système

Bien qu'elle se réalise sur une représentation abstraite du système (i.e. : les modèles de bon fonctionnement et de défauts que nous avons présentés aux chapitres 2 et 3 précédents), l'étude de la diagnosticabilité du système est néanmoins inhérente au procédé d'analyse du diagnostiqueur. Cela signifie que cette étude se réalise en considérant la manière dont va réellement fonctionner le diagnostiqueur, c'est-à-dire ses règles d'analyse pour détecter et isoler les différents défauts répertoriés. Nous pourrions en effet réaliser cette étude sur le système réel, ou du moins sur une reconstitution, mais cela aurait un impact financier beaucoup plus important.

En fonctionnement, un diagnostiqueur est « alimenté » par le comportement observé du système : c'est-à-dire l'évolution dans le temps des valeurs que prennent les variables observables du système. Il va donc s'agir d'un flux de données constitué par le vecteur des valeurs de ces variables observables du système à chaque instant du temps. Le diagnostiqueur va donc vérifier que ce comportement observé satisfait bien une propriété de bon fonctionnement, propriété déterminée suivant la méthodologie de diagnostic choisie. Dans le cas contraire où le comportement observé ne satisfait pas cette propriété de bon fonctionnement, le diagnostiqueur va rechercher quelle propriété de mauvais fonctionnement il satisfait, propriété toujours déterminée suivant la méthodologie de diagnostic choisie.

L'étude de la diagnosticabilité des défauts se réalise donc en étudiant les validités de ces propriétés, de bon ou mauvais fonctionnement, par le comportement observé du système. Nous nommons *caractérisation de défauts* une telle famille de propriétés qui doit être constituée d'autant de propriétés que de

défauts considérés et dont chacune doit donc refléter la règles d'analyse du diagnostiqueur suivant le défaut considéré : c'est-à-dire que la *propriété normale* doit représenter le fonctionnement normal du système, et chaque *propriété de défaut* doit représenter le fonctionnement du système sous la présence du défaut considéré. Comme l'étude se réalise par utilisation des différents modèles de bon fonctionnement et de défauts du système, nous allons devoir définir ce que représente ce comportement observé du système du point de vue de ces modèles. Il va s'agir des évolutions dans le temps des variables observables du système obtenues par simulation des différents modèles. C'est ce que nous nommerons les *comportements observables* du système sous la présence ou non d'un défaut.

L'étude de la diagnosticabilité va ainsi revenir à s'assurer non seulement que tout comportement observable normal valide toujours la propriété de bon fonctionnement, mais en plus que tout comportement observable anormal (i.e. : un comportement observable sous la présence d'un défaut) ne valide uniquement que la propriété du défaut considéré et pas les autres propriétés des autres défauts, même celle de bon fonctionnement. De ce fait, toute l'étude de la diagnosticabilité va se baser sur l'étude des validités des propriétés de bon ou mauvais fonctionnement évaluées par les comportements observables du système, sous la présence ou non d'un défaut.

4.1.3.2 Une première formalisation de la diagnosticabilité du système

Pour reprendre plus formellement ce que nous venons de présenter, un diagnostiqueur analyse en temps réel (i.e. : à chaque instant du temps) le comportement réellement observé du système. Lorsque le système fonctionne normalement, le diagnostiqueur doit s'en assurer : c'est-à-dire que la propriété de bon fonctionnement est toujours validée à chaque instant du temps. Par contre dès qu'un défaut préalablement répertorié apparaît, le diagnostiqueur doit le détecter et l'isoler avant une certaine borne temporelle b de diagnostic. Tout ceci afin de pouvoir par la suite prendre les décisions adéquates quant aux suites de fonctionnement du système. Cela signifie donc que dès qu'un défaut apparaît à un instant $\mathrm{t}_n \in \mathbb{T}_\iota$ du temps, le diagnostiqueur doit le diagnostiquer avant l'instant borné $\mathrm{t}_n + b$: c'est-à-dire que dans la fenêtre temporelle $[\mathrm{t}_n; \mathrm{t}_n + b]$ la propriété de bon fonctionnement doit devenir invalide, ce qui permet la détection du défaut, et qu'ensuite, après un délai potentiel $h \in \mathbb{T}_\iota$ d'isolation, seule la propriété du défaut doit rester valide, ce qui permet l'isolation du défaut.

Il est nécessaire de remarquer que ce que nous venons de formaliser est basé sur une hypothèse concernant les conséquences d'un défaut sur le fonctionnement du système. Nous supposons en effet qu'un défaut créé en premier lieu une perturbation momentanée non significative en elle-même, puis une perturbation durable et reconnaissable sur le comportement du système ; ceci pouvant bien sûr ne pas être visible sur le comportement observable du système. Cela justifie donc qu'en fonctionnement et suite à une détection, le diagnostiqueur attende un certain délai d'isolation avant de pouvoir conclure sur le défaut présent. Notons que cette hypothèse, validée dans la pratique, est classiquement admise dans la littérature.

Par ailleurs, ce que nous venons de présenter est une formalisation que nous avons, en tant que concepteur, sur le fonctionnement du diagnostiqueur. Néanmoins, celui-ci ne connaîtra jamais l'occurrence $\mathrm{t}_n \in \mathbb{T}_\iota$ d'un défaut. Son seul point de repère sera donc toujours un instant de détection d'un défaut : c'est-à-dire un instant $\mathrm{t}_k \in \mathbb{T}_\iota$ du temps où la propriété de bon fonctionnement passe de valide à invalide et qui sera normalement postérieur ou égal à cette occurrence t_n du défaut. Suite à cette détection à un instant t_k, postérieur ou égal à l'occurrence t_n, il faudra attendre un certain délai temporel $h \in \mathbb{T}_\iota$ d'isolation avant de pouvoir décider du défaut apparu. Il est en effet fortement probable que suite à l'occurrence t_n d'un défaut, le système subisse différents changements d'états avant de se stabiliser, et les propriétés de mauvais fonctionnement peuvent osciller entre valide et invalide durant cette phase de changement. Il est donc nécessaire d'attendre ce délai h d'isolation avant de pouvoir statuer sur le défaut apparu. Enfin, même si nous supposions qu'au terme de ce délai h, soit en $\mathrm{t}_k + h$, un seul défaut voit sa caractérisation vérifiée, nous devons nous assurer que cela n'est pas fortuit, ce qui serait le cas si dans un voisinage temporel après $\mathrm{t}_k + h$, les propriétés changeraient de valeur de

vérité. Il faut pour cela introduire un délai temporel δ de confiance qui nous assure que durant la fenêtre temporelle $[(\mathrm{t}_k + h); (\mathrm{t}_k + h) + \delta]$ seule la propriété du défaut demeure valide.

Enfin et suivant la méthodologie de diagnostic utilisée, le diagnostiqueur va avoir besoin d'enregistrer le comportement réellement observé du système sur une certaine durée temporelle $\lambda \in \mathbb{T}_\iota$ représentant sa longueur d'enregistrement. Cela signifie qu'à chaque instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps, le diagnostiqueur aura enregistré le comportement réellement observé du système sur la fenêtre temporelle $[\mathrm{t} - \lambda; \mathrm{t}]$.

La figure 4.1 suivante de la page 103 représente les deux points de vue des évolutions temporelles des validités des propriétés lors d'un fonctionnement anormal du système. Le deuxième graphique considère le point de vue selon l'étude de la diagnosticabilité et le troisième graphique considère celui du diagnostiqueur.

Le premier graphique représente un fonctionnement anormal du système sous la présence d'un défaut $F \in \Gamma \setminus \{F_0\}$ à une occurrence t_n, que nous avons « résumé » uniquement par une variable c de consigne et une variable y de mesure. Nous pouvons observer que la mesure suit bien la consigne avant cette occurrence t_n du défaut, ce qui signifie que le système fonctionne normalement. Par contre la mesure chute pour atteindre une valeur arbitraire après cette occurrence, ce qui traduit un fonctionnement anormal du système.

Le deuxième graphique représente, du point de vue de l'étude de la diagnosticabilité, les validités des propriétés suivant ce comportement observable anormal. P_{F_0} représente la propriété de bon fonctionnement, P_F représente celle du fonctionnement sous la présence du défaut F et les autres $\mathrm{P}_{F'}$ représentent les propriétés des autres défauts (autres que le cas normal F_0 et le défaut F). Avant l'occurrence t_n, le système fonctionne normalement et la propriété P_{F_0} de bon fonctionnement est donc toujours valide. À partir de l'occurrence t_n, le défaut doit être détecté à un instant t_k (i.e. : la propriété P_{F_0} de bon fonctionnement doit passer de valide à invalide à cet instant t_k), puis isolé à l'instant $(\mathrm{t}_k + h)$ en s'assurant qu'il reste isolé sur la fenêtre temporelle $[(\mathrm{t}_k + h); (\mathrm{t}_k + h) + \delta]$ (i.e. : seule la propriété P_F de fonctionnement sous la présence de F est valide sur cette fenêtre temporelle $[(\mathrm{t}_k + h); (\mathrm{t}_k + h) + \delta]$, les autres propriétés devant être invalides). Tout cela doit de plus se faire avant la borne b de diagnostic, ce qui signifie que l'instant t_k de détection doit vérifier $(\mathrm{t}_k + h) \leq (\mathrm{t}_n + b)$.

Le troisième graphique représente les validités des propriétés du point de vue du diagnostiqueur. Comme nous l'avons signalé, le diagnostiqueur ne connait pas l'occurrence t_n du défaut F, et son seul point de repère est l'instant de détection t_k lorsque la propriété P_{F_0} de bon fonctionnement passe de valide à invalide. Suite à cette détection, il attend l'instant $(\mathrm{t}_k + h)$ pour conclure sur le défaut apparu. Cette conclusion étant assurée par l'étude de diagnosticabilité.

4.1.3.3 Paramètres d'étude de la diagnosticabilité

Comme nous venons de le voir, non seulement l'étude de la diagnosticabilité doit découler du processus d'analyse du diagnostiqueur, c'est-à-dire ses règles d'analyse données par une caractérisation de défauts suivant la méthodologie de diagnostic adoptée, mais le « point de vue » du diagnostiqueur, qui naturellement n'a qu'une vision limitée, doit aussi être pris en compte dans l'étude de la diagnosticabilité. Cela signifie que cette étude doit intégrer les paramètres permettant de s'assurer qu'il diagnostiquera sans ambiguïté les défauts potentiels suivant une certaine caractérisation de défauts. L'étude de la diagnosticabilité est par conséquent paramétrée par quatre données : la longueur d'enregistrement λ du diagnostiqueur, la borne b de diagnostic, le délai h d'isolation et le délai δ de confiance.

La longueur d'enregistrement λ du diagnostiqueur représente la longueur temporelle du comportement réellement observé que devra enregistrer le diagnostiqueur à chaque instant du temps lors du fonctionnement du système. Cela signifie qu'à chaque instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps, le diagnostiqueur aura enregistré le comportement réellement observé du système sur la fenêtre temporelle $[\mathrm{t} - \lambda; \mathrm{t}]$. Par

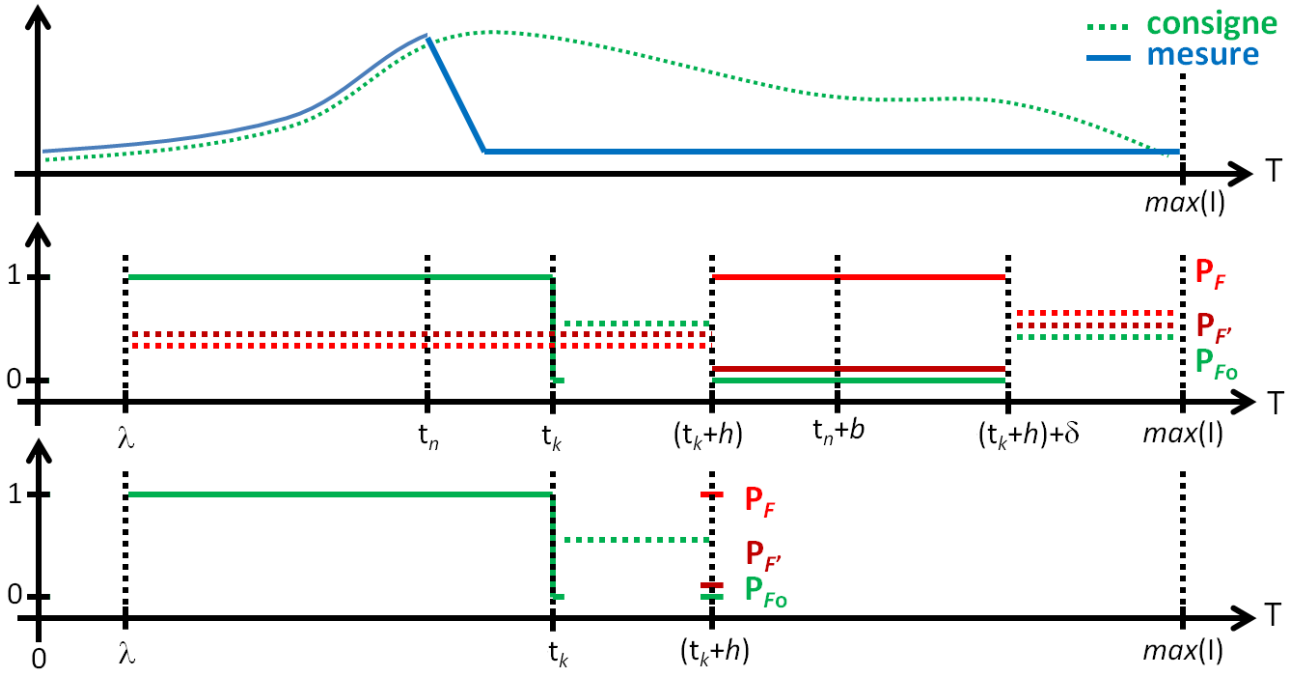

Figure 4.1 – Points de vue des validités des propriétés.

ailleurs, il ne pourra analyser le fonctionnement du système qu'à partir de l'instant $t_0 + \lambda$ après la mise en marche du système à l'instant $t_0 \in \mathbb{T}_\iota$. Cette longueur doit être déterminée suivant la cible sur laquelle sera implanté le diagnostiqueur. Si celle-ci est limitée dans sa capacité de stockage, il faudra donc choisir cette longueur assez petite. Il y aura donc un compromis à faire entre le résultat du diagnostiqueur et cette longueur λ d'enregistrement.

La borne b de diagnostic impose au diagnostiqueur de diagnostiquer un défaut au plus tard à cette borne après son occurrence. C'est-à-dire qu'à une occurrence t_n d'un défaut, le diagnostiqueur doit le détecter et l'isoler avant l'instant $t_n + b$. Cette borne b doit être déterminée suivant non seulement la sévérité des défauts à prendre en compte, mais aussi par rapport à la dynamique de réponse du système. Nous faisons l'hypothèse que cette borne b est unique pour tous les défauts considérés ; il pourrait cependant être possible de déterminer différentes bornes pour différentes classes de défauts.

Le délai h d'isolation permet au diagnostiqueur de conclure, suite à une détection, sur le défaut apparu. Cela signifie que suite à la détection d'un défaut à un instant t_k, le diagnostiqueur va conclure sur le défaut apparu à l'instant $(t_k + h)$. Ce délai doit être fixe pour le diagnostiqueur. Cependant le moyen de l'obtenir peut être variable : lors de l'étude de la diagnosticabilité, il est possible soit de fixer ce délai h, soit de mener l'étude en cherchant le délai minimum h_{min} satisfaisant aux définitions de la diagnosticabilité. Pour la suite, nous supposerons que ce délai est fixé au préalable. Dans le cas où la diagnosticabilité des défauts n'est pas établie (i.e. : certains défauts ne sont pas diagnosticables) il faudra alors voir si ce n'est pas, entre autre, ce délai h qui est trop court. Si oui, nous devrons alors prendre un délai supérieur puis mener à nouveau l'étude. Remarquons d'une part que plus ce délai est important et plus le temps octroyé pour la détection sera court, et d'autre part qu'il est bien évidemment borné par b (i.e. : $h \leq b$). En effet comme nous considérons que l'instant t_k est au moins supérieur ou égal à l'occurrence t_n du défaut, alors $t_n + b \leq t_k + b$, et ainsi si $b < h$ nous avons donc $t_n + b < t_k + h$ ce qui est en contradiction avec cette borne b de diagnostic : $t_k + h \leq t_n + b$. Nous pouvons néanmoins garder à l'esprit la remarque de [VRYK03] indiquant que la phase de détection est généralement assez rapide par rapport à la phase d'isolation.

Enfin le délai δ de confiance des validités des propriétés permet de nous assurer en étude de diagnosticabilité que le diagnostiqueur conclura sans erreur à un défaut apparu suite à une détection.

Comme suite à une détection à un instant t_k, le diagnostiqueur conclura sur le défaut apparu à l'instant $(t_k + h)$; il est nécessaire de nous assurer que ce délai d'isolation h est suffisant. Il faut pour cela vérifier que seule la propriété du défaut conclue par le diagnostiqueur est valide après l'instant d'isolation $(t_k + h)$; ce qui est réalisé sur la fenêtre temporelle $[(t_k + h); (t_k + h) + \delta]$. Ce paramètre δ servant de délai de confiance est fortement lié à la dynamique du système. Nous pouvons donc le supposer égal ou supérieur à cette dynamique.

4.2 Comportements observables du système

Nous avons expliqué qu'en fonctionnement, le diagnostiqueur analysera le comportement réellement observé du système : c'est-à-dire l'évolution dans le temps des valeurs que prendront les variables observables du système. Comme l'étude de la diagnosticabilité se réalise par utilisation des modèles de bon fonctionnement et de défauts du système, nous devons donc définir ce que représente le comportement réellement observé du système du point de vue de ces modèles. C'est ce que nous nommons les *comportements observables* du système qui vont donc représenter la manière de fonctionner du système telle que pourrait le « voir » le diagnostiqueur. Ces comportements observables vont être obtenus à partir du comportement du système en ne le considérant que sur les variables observables. Un *comportement* du système est obtenu du modèle normal ou d'un modèle de défaut, suivant une évolution de la consigne donnée par l'opérateur et sous la présence du défaut (cas normal inclus) considéré à une certaine occurrence (occurrence non-définie pour le cas normal). Il s'agit des valeurs que vont prendre toutes les variables du système lors de son fonctionnement suivant cette évolution de la consigne et sous la présence de ce défaut à l'occurrence considérée. Une évolution de la consigne dans le temps se nomme une *instruction*. En considérant un ensemble d'instructions, nous pourrons définir l'ensemble des comportements du système puis l'ensemble des comportements observables.

4.2.1 Ensemble des instructions de l'opérateur

Une instruction de l'opérateur correspond à l'évolution de la consigne dans le temps. Nous avons considéré l'ensemble $\mathbb{T}_\iota$ du temps d'exécution du système comme étant infini. Or cet ensemble doit au plus représenter la durée de fonctionnement du système entre chaque mise en marche et mise à l'arrêt, et ce durant toute son exploitation. Il s'agira de quelques heures pour certains systèmes (un moteur de véhicule par exemple), à quelques mois voire quelques années pour d'autres (par exemple une chaudière d'habitation), avec très certainement des périodes de maintenance. Cela pourra même aller jusqu'à la durée totale d'exploitation pour des systèmes critiques (une centrale nucléaire par exemple). Nous pouvons par conséquent nous restreindre à des instructions définies sur des fenêtre temporelles (des intervalles de $\mathbb{T}_\iota$) finies et commençant à 0.

Formellement, une instruction est une suite cs, d'une fenêtre temporelle finie $\mathrm{I}_{cs} = [0; max(\mathrm{I}_{cs})] \subseteq \mathbb{T}_\iota$ dans le domaine C du vecteur de consigne ; c'est-à-dire :

$$\begin{array}{rcccl} cs & : & \mathrm{I}_{cs} & \longrightarrow & C \\ & & \mathrm{t} & \longmapsto & cs(\mathrm{t}) \end{array}$$

Comme nous l'avons spécifié, il faut considérer un ensemble $Cons = \{cs_1; \ldots; cs_u\}$ d'instructions pour définir l'ensemble des comportements du système. Par définition d'une instruction $cs \in Cons$, chacune est définie sur sa propre fenêtre temporelle $\mathrm{I}_{cs} \subseteq \mathbb{T}_\iota$ finie et commençant à 0. Or pour l'étude de la diagnosticabilité, nous aurons besoin qu'elles soient toutes définies sur la même fenêtre temporelle finie et commençant à 0. Nous posons donc $\mathrm{I} = max\{\mathrm{I}_{cs} \setminus cs \in Cons\}$ la plus longue fenêtre temporelle des instructions de l'ensemble $Cons$, et définissons chacune des instructions $cs \in Cons$ sur I par extension de la dernière valeur de cs sur I_{cs} lorsque cette fenêtre temporelle I_{cs} est plus petite que I (i.e. : $\mathrm{I}_{cs} \subsetneq \mathrm{I}$). Cela signifie formellement et pour chaque instruction $cs \in Cons$:

$$cs \ : \ \mathrm{I} \longrightarrow C$$
$$\mathrm{t} \longmapsto \begin{cases} cs(\mathrm{t}) & ; \quad \text{si } \mathrm{t} < max(\mathrm{I}_{cs}) \\ cs(max(\mathrm{I}_{cs})) & ; \quad \text{si } \mathrm{t} \geq max(\mathrm{I}_{cs}) \end{cases}$$

Cet ensemble $Cons$ d'instructions est supposé être le plus représentatif possible : c'est-à-dire permettant d'obtenir l'ensemble des plages de fonctionnement du système.

Par ailleurs et comme nous venons de l'indiquer dans le paragraphe concernant les paramètres d'étude de la diagnosticabilité, selon la méthodologie de diagnostic utilisée, le diagnostiqueur va avoir besoin d'enregistrer le comportement réellement observé du système sur une longueur d'enregistrement $\lambda \in \mathbb{T}_\iota$. Par conséquent à sa mise en marche à un instant initial $\mathrm{t}_{init} \in \mathbb{T}_\iota$, il ne pourra analyser le fonctionnement du système qu'à partir de l'instant $\mathrm{t}_{init} + \lambda$. Du point de vue des instructions et comme elles sont définies à partir de l'instant initial $\mathrm{t}_{init} = 0$, nous pouvons supposer sans perte de généralité qu'elles sont toutes représentatives (i.e. : reflètent bien l'ensemble des plages de fonctionnement du système) à partir d'un instant t_0 fixé dans I tel que d'une part $\lambda \leq \mathrm{t}_0$ et d'autre part $b \leq \mathrm{t}_0$; ceci quitte à les définir constantes sur la fenêtre temporelle $[0; \mathrm{t}_0]$ et d'augmenter, si besoin, en conséquence I afin qu'elles reflètent bien l'ensemble des plages de fonctionnement du système sur l'intervalle temporel $[\mathrm{t}_0; max(\mathrm{I})]$. Le fait que cet instant t_0 soit aussi supérieur à b sera par la suite justifié lorsque nous considérerons les caractérisations de défauts, notamment la caractérisation parfaite.

La figure 4.2 suivante de la page 106 représente une simulation d'une instruction d'un système arbitraire pour lequel il y a trois variables de consigne (i.e. : le vecteur des consignes est donc de la forme $c = (c_1, c_2, c_3)$) et dont le domaine temporel est $\mathrm{I} = [0; 60]$. Contrairement au cas d'étude, que nous allons voir à la fin de ce chapitre, nous ne pouvons observer aucune corrélation entre ces trois évolutions. En supposant que la borne b de diagnostic vaille 2, que la longueur d'enregistrement λ vaille 3 et que l'instant t_0 vaille 5, qui est donc bien supérieur à λ et b, nous observons bien que les valeurs des consignes sont constantes sur la fenêtre temporelle $[0; \mathrm{t}_0]$.

4.2.2 Occurrences des défauts

Une des caractéristiques du comportement d'un défaut est son occurrence : c'est-à-dire l'instant du temps $\mathbb{T}_\iota$ où il apparaît. Cela signifie que le système fonctionne normalement avant cet instant t_n, puis qu'il fonctionne sous la présence du défaut après. Nous avons vu, au chapitre 3 de présentation de la typologie des défauts, que cette occurrence peut être un instant aléatoire ou un instant donné ou encore peut être dû à un événement particulier interne ou externe au système. Comme ce qui nous intéresse est de nous assurer qu'un défaut soit diagnosticable lorsqu'il apparaît et non de prédire son occurrence, nous n'étudierons donc, comme nous l'avons indiqué dans le paragraphe précédent de contraintes industrielles, que des occurrences particulières à des instants arbitraires t_n du temps $\mathbb{T}_\iota$. Cela signifie que nous allons manipuler ces occurrences, nous allons les maîtriser.

Un comportement avec défaut du système va être défini suivant une instruction $cs \in Cons$ donnée. Comme nous n'avons fait aucune hypothèse sur cet ensemble $Cons$ d'instructions, nous n'allons donc pas en faire non plus sur les occurrences potentielles de chacun des défauts, mises à part celles induites par l'étude de la diagnosticabilité que nous allons faire dès à présent. Notre but est encore une fois de garder une certaine liberté dans ces hypothèses. Nous pourrions très bien par exemple considérer que pour chacun des défauts, les occurrences possibles représentent toute la fenêtre temporelle I de chacune des instructions $cs \in Cons$. Or comme nous le verrons par la suite, il sera possible de se restreindre à un nombre limité d'occurrences suivant les instructions. Par exemple, lorsque la consigne n'évolue pas sur une portion de la fenêtre temporelle, il n'est peut-être pas nécessaire de considérer tous les instants de cette portion comme occurrences potentielles du défaut. Rappelons par ailleurs que nous avons exigé de considérer un diagnostic en temps borné b, la détection et l'isolation d'un défaut doivent donc s'effectuer uniquement en un temps borné par b après son occurrence.

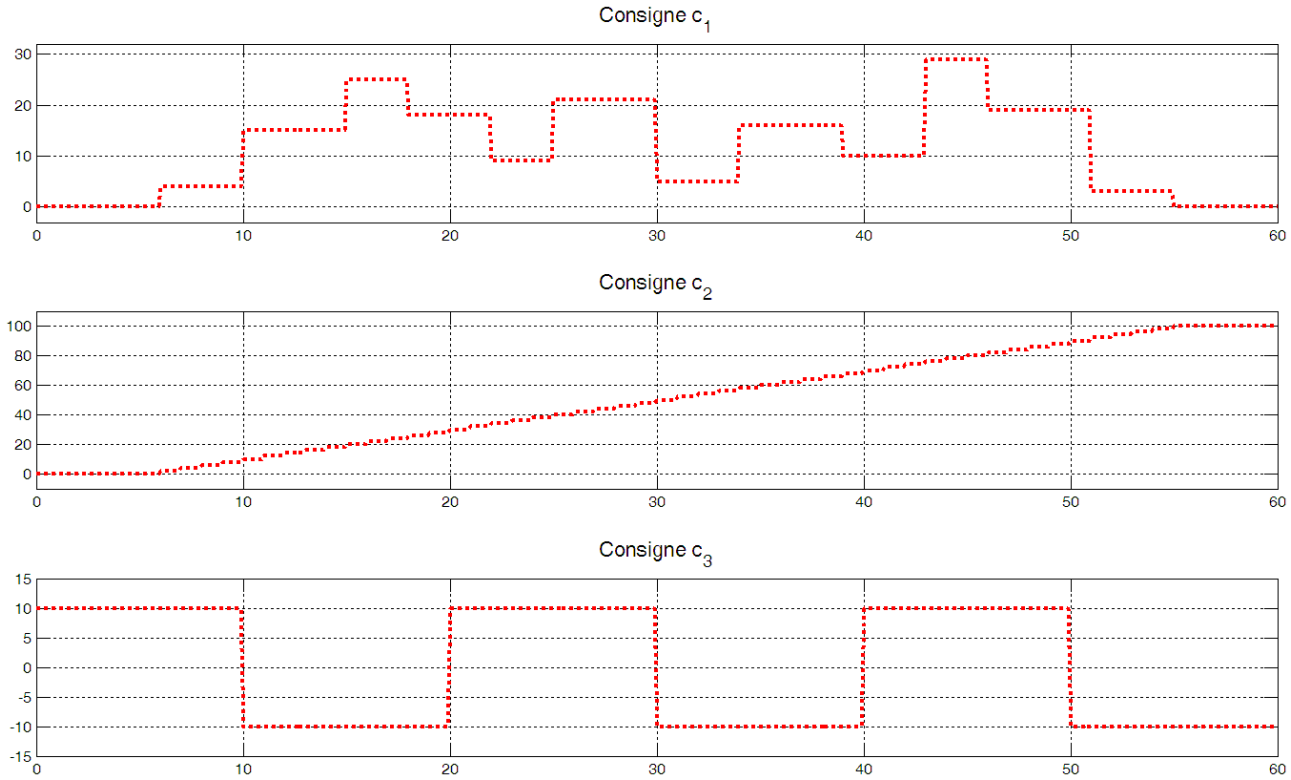

Figure 4.2 – Simulation d'une instruction.

4.2.2.1 Contraintes liées à l'étude de la diagnosticabilité

Différentes contraintes induites par l'étude de la diagnosticabilité vont être faites dès à présent. Qu'elles soient liées à la borne de diagnostic ou à la période d'initialisation du diagnostiqueur, ou encore qu'elles soient liées au type de défauts considérés, ces contraintes vont nous permettre de construire des ensembles $\Omega_{(F,cs)}$ d'occurrences des défauts pour chacun des défauts $F \in \Gamma \setminus \{F_0\}$ et chacune des instructions $cs \in Cons$.

Contraintes liées à la borne de diagnostic En fonctionnement, le diagnostiqueur va surveiller le comportement observé du système et, à l'occurrence d'un défaut, va devoir le détecter et l'isoler, en analysant ce comportement observé, au plus tard à la borne b après cette occurrence. Cela signifie qu'en étude de diagnosticabilité, l'analyse va se faire de la même manière mais sur la fenêtre temporelle finie I et en s'assurant que le défaut est bien diagnostiqué jusqu'au délai δ de confiance.

Ainsi pour une occurrence t_n considérée sur la fenêtre temporelle $[max(\mathrm{I}) - (b + \delta); max(\mathrm{I})]$, le comportement ne pourra pas être analysé car il ne sera potentiellement pas défini sur une partie de cette fenêtre temporelle. En effet comme le défaut doit être diagnostiqué au plus tard à la borne b après son occurrence t_n, cela signifie que l'instant de détection t_k peut être au maximum à l'instant $\mathrm{t}_n + (b - h)$. Donc pour s'assurer de l'isolation du défaut durant la fenêtre temporelle $[(\mathrm{t}_k + h); (\mathrm{t}_k + h) + \delta]$, il faut que la fenêtre temporelle $[\mathrm{t}_n + (b - h) + h; \mathrm{t}_n + (b - h) + h + \delta]$ soit incluse dans I, c'est-à-dire que la fenêtre temporelle $[\mathrm{t}_n + b; \mathrm{t}_n + b + \delta]$ soit incluse dans I, ce qui n'est possible que si $\mathrm{t}_n < max(\mathrm{I}) - (b + \delta)$.

Contraintes liées à la période d'initialisation du diagnostiqueur Comme nous venons de le voir lors la définition des instructions, il est possible que selon la méthodologie de diagnostic utilisée, le diagnostiqueur ait une période d'initialisation $\lambda \in \mathbb{T}_\iota$. C'est-à-dire qu'à sa mise en marche à un instant initial $\mathrm{t}_{init} \in \mathbb{T}_\iota$ du temps, il ne pourra analyser le fonctionnement du système, et ainsi détecter puis

isoler un défaut potentiel apparu, qu'à partir de l'instant $t_{init} + \lambda$. Ainsi pour une occurrence t_n considérée sur la fenêtre temporelle $[t_{init}; t_{init} + \lambda]$, le comportement ne pourra pas être analysé car le diagnostiqueur ne sera pas totalement initialisé.

Pour l'étude de la diagnosticabilité, nous avons imposé que toutes les instructions ne soient représentatives qu'à partir d'un instant fixé $t_0 \in I$ tel que $\lambda \leq t_0$. Nous pouvons donc imposer que les occurrences des défauts ne soient pas considérées sur la sous-fenêtre temporelle $[0; t_0]$ de I, c'est-à-dire qu'elles ne soient donc considérées qu'à partir de l'instant $t_0 \in I$.

Contraintes liées aux défauts progressifs Au chapitre 3 de présentation de la typologie des défauts, nous avons défini les défauts faiblement progressifs représentant des usures pouvant être liées au fonctionnement normal du système (des encrassements ou des fuites par exemple). Nous avions remarqué que le délai entre l'occurrence t_n d'un tel défaut et l'instant où il devient « significatif » (i.e. : l'instant où il est considéré comme présent, sans nécessairement l'être totalement) pouvait être extrêmement long (de l'ordre de la journée à l'année suivant les systèmes et les défauts considérés). Ainsi et bien que [BJL+90] indique que ces défauts doivent être pris en compte lors des phases de maintenance du système, si nous souhaitons les intégrer dans l'étude de la diagnosticabilité, nous remarquons que cette notion d'occurrence n'est pas adéquate pour ces défauts progressifs. En effet la borne b de diagnostic, qui représente la dynamique de réponse d'un diagnostiqueur (de l'ordre de quelques secondes) sera généralement trop courte par rapport au coefficient de progression du défaut (de l'ordre de la journée à l'année suivant les systèmes et les défauts considérés). Il nous faut donc définir d'une manière différente les occurrences et/ou comportements de ces défauts afin d'en étudier leurs impacts.

Pour un défaut faiblement progressif, ayant donc une faible pente de progression, l'occurrence t_n du défaut à considérer est lorsqu'il se situe à un certain niveau jugé critique pour le fonctionnement du système (lorsque le défaut atteint un certain niveau ou que le système ne peut plus être efficacement contrôlé par exemple). Pour un encrassement, par exemple, ce peut être lorsqu'il atteint une certaine valeur d'obstruction, et pour une fuite, ce peut être lorsqu'elle atteint une certaine valeur d'ouverture.

En étude de diagnosticabilité, les instructions considérées n'auront généralement pas de longs domaines temporels permettant de rendre compte de l'évolution de ce type de défauts où leurs évolutions sont étendues sur des périodes de un jour à un an. Pour un ensemble $Cons$ d'instructions ayant un domaine temporel assez court vis-à-vis de la pente de progression d'un tel défaut, il va donc falloir rajouter un biais s au comportement du défaut. Ce comportement va donc commencer à partir de ce biais s puis continuer sa progression jusqu'à atteindre le niveau critique l à l'occurrence $t_n \in I$ souhaitée. Dans certains cas, que nous allons d'ailleurs étudier sur le cas d'étude, ce domaine temporel I sera tellement court que la présence du défaut n'aura pas le temps d'augmenter. Il faudra donc définir son comportement comme constant au niveau requis et considérer un signal de défaut *ad-hoc* permettant de fournir l'occurrence $t_n \in I$ du défaut. C'est d'ailleurs ce que nous avons vu au chapitre 3 lors de la présentation des défauts d'usure de la ligne d'air : les encrassements du compresseur et de l'électrovanne ainsi que la fuite d'air dans la tuyauterie.

4.2.2.2 Ensembles d'occurrences des défauts

Par conséquent, pour chaque défaut $F \in \Gamma \backslash \{F_0\}$ et chaque instruction $cs \in Cons$, nous considérons un ensemble $\Omega_{(F,cs)}$ d'occurrences $t_n \in I$ du défaut F suivant cette instruction cs. Nous supposons que l'intersection de chacun de ces ensembles d'occurrences $\Omega_{(F,cs)}$ avec les fenêtres temporelles $[0; t_0]$ et $[max(I) - (b+\delta); max(I)]$ sont vides (i.e. : $\Omega_{(F,cs)} \cap [0; t_0] = \emptyset$ et $\Omega_{(F,cs)} \cap [max(I) - (b+\delta); max(I)] = \emptyset$). Cela signifie que les occurrences sont uniquement définies dans la fenêtre temporelle $[t_0; max(I) - (b + \delta)]$:

$$\Omega_{(F,cs)} \subseteq [t_0; max(I) - (b + \delta)]$$

Faisons attention au fait que chaque ensemble $\Omega_{(F,cs)}$ n'est pas nécessairement un intervalle temporel ou une réunion d'intervalles temporels. Ils peuvent très bien représenter un nombre limité d'instants dans une fenêtre temporelle : par exemple $\Omega_{(F,cs)} = \{20; 32.05; 40.05; 42\}$.

La figure 4.3 suivante représente l'ensemble d'occurrences potentielles d'un défaut suivant la simulation de l'instruction précédente. En supposant toujours que la borne b de diagnostic vaille 2, que l'instant t_0 vaille 5 (toujours supérieur à b et $\lambda = 3$) et que le délai δ de confiance des validités des propriétés vaille 1, l'ensemble $\Omega_{(F,cs)}$ d'occurrences potentielles d'un défaut quelconque $F \in \Gamma$ est donc défini par l'ensemble des points (représentés par des croix sur l'axe noir) dans la fenêtre temporelle $[5; 57]$.

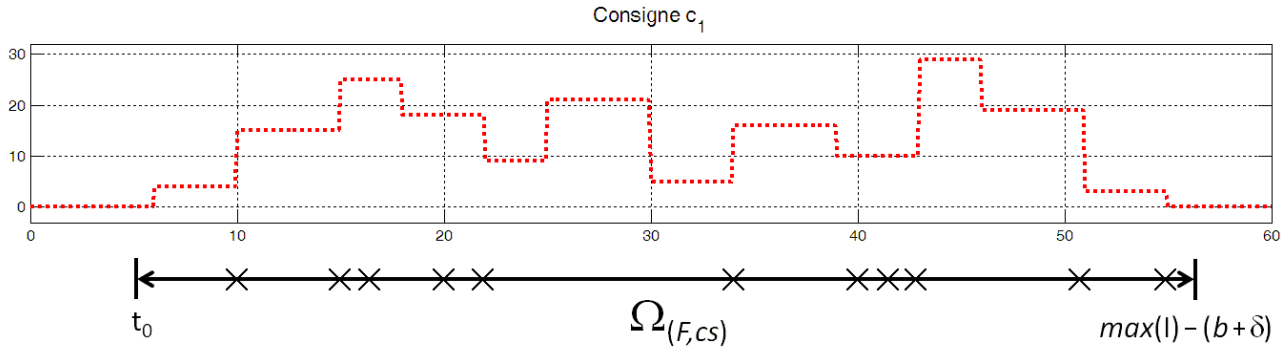

Figure 4.3 – Ensemble d'occurrences potentielles d'un défaut suivant une simulation d'une instruction.

4.2.3 Comportements du système

Un comportement du système représente l'ensemble des valeurs prises par les variables lors de son fonctionnement suivant une instruction de l'opérateur ; ce fonctionnement pouvant être sous la présence ou non d'un défaut. Un comportement est donc déterminé non seulement en fonction d'une instruction $cs \in Cons$ donnée mais aussi en fonction d'un défaut $F \in \Gamma$ considéré.

Rappelons pour la suite que $\overline{\mathrm{VD}}$ représente le produit de tous les domaines des variables du système : $\overline{\mathrm{VD}} = C \times A \times U \times X \times Y \times D \times A^M \times U^M \times X^M \times Y^M \times D^M$. De plus, pour un vecteur $v = (v_1, \ldots, v_n)$ et un indice $i \in \{1; \ldots; n\}$, le i-ème élément de v est $p_i(v) = v_i$. Par exemple pour un vecteur $v(\mathrm{t}) \in \overline{\mathrm{VD}}$ représentant les valeurs des variables du système à un instant $\mathrm{t} \in \mathbb{T}_\iota$, ce qui signifie que $v(\mathrm{t}) = (c(\mathrm{t}), a(\mathrm{t}), u(\mathrm{t}), x(\mathrm{t}), y(\mathrm{t}), d(\mathrm{t}), a^M(\mathrm{t}), u^M(\mathrm{t}), x^M(\mathrm{t}), y^M(\mathrm{t}), d^M(\mathrm{t}))$, alors $p_1(v(\mathrm{t}))$ représente la valeur $c(\mathrm{t})$ de la consigne de l'opérateur à cet instant t et $p_{10}(v(\mathrm{t}))$ représente la valeur $y^M(\mathrm{t})$ de la mesure du modèle embarqué toujours à cet instant t. Remarquons que nous avons fait un abus de notation car ici, $p_1(v(\mathrm{t}))$ ou $p_{10}(v(\mathrm{t}))$, et même les autres projections de $v(\mathrm{t})$, peuvent désigner un vecteur selon le système considéré.

4.2.3.1 Comportements normaux du système

Un comportement normal du système est donné suivant une instruction $cs \in Cons$ donnée. Il représente l'ensemble des valeurs que prennent les variables du système lorsqu'il fonctionne normalement (i.e. : sans présence d'un défaut) suivant cette instruction cs. Il s'agit donc de l'ensemble des vecteurs de données $v(\mathrm{t}) \in \overline{\mathrm{VD}}$, considérés à chaque instant $\mathrm{t} \in \mathrm{I}$ suivant la consigne cs et en fonction de l'ensemble des équations du système en fonctionnement normal (i.e. : les équations de $\overline{S_{F_0}}$ données au chapitre 2).

Définition 4.1 *Pour une instruction $cs \in Cons$, le* comportement normal *du système $\overline{S}$ suivant cs est l'ensemble, noté $B(cs, F_0, 0)$, constitué des vecteurs $(\mathrm{t}, v(\mathrm{t})) \in \mathbb{T}_\iota \times \overline{\mathrm{VD}}$ vérifiant :*

1. *pour tout instant* $\mathrm{t} \in \mathrm{I}$*, il existe un unique vecteur* $v(\mathrm{t}) \in \overline{\mathrm{VD}}$ *tel que* $(\mathrm{t}, v(\mathrm{t})) \in B(cs, F_0, 0)$.
2. *pour tout instant* $\mathrm{t} \in \mathrm{I}$, $p_1(v(\mathrm{t})) = cs(\mathrm{t})$.
3. *pour tout instant* $\mathrm{t} \in \mathrm{I} \setminus \{max(\mathrm{I})\}$,
   (a) $p_4(v(\mathrm{t}+1)) = f(p_4(v(\mathrm{t})), \theta, p_3(v(\mathrm{t})), p_6(v(\mathrm{t})))$
   (b) $p_5(v(\mathrm{t})) = g(p_4(v(\mathrm{t})))$
   (c) $p_2(v(\mathrm{t}+1)) = h(p_1(v(\mathrm{t})), p_2(v(\mathrm{t})), p_5(v(\mathrm{t})))$
   (d) $p_3(v(\mathrm{t})) = k(p_2(v(\mathrm{t})))$
   (e) $p_4(v(0)) = x_{init}$
   (f) $p_2(v(0)) = a_{init}$
   (g) $p_9(v(\mathrm{t}+1)) = f^M(p_9(v(\mathrm{t})), \theta^M, p_8(v(\mathrm{t})), p_{11}(v(\mathrm{t})))$
   (h) $p_{10}(v(\mathrm{t})) = g^M(p_9(v(\mathrm{t})))$
   (i) $p_7(v(\mathrm{t}+1)) = h(p_1(v(\mathrm{t})), p_7(v(\mathrm{t})), p_{10}(v(\mathrm{t})))$
   (j) $p_8(v(\mathrm{t})) = k(p_7(v(\mathrm{t})))$
   (k) $p_9(v(0)) = x^M_{init}$
   (l) $p_7(v(0)) = a^M_{init}$

Dans cette définition de comportement normal du système, la notation $B(cs, F_0, 0)$ exprime qu'il s'agit d'un comportement du système suivant l'instruction $cs \in Cons$ et sous la présence du « défaut » F_0, le cas normal. Le paramètre 0 représente l'occurrence du cas normal F_0 qui peut être considéré comme un « défaut » apparaissant toujours à l'instant $\mathrm{t}_n = 0$. Il est rajouté pour harmoniser avec la notation des comportements avec défauts que nous allons présenter juste après.

La première condition signifie qu'il y a existence et unicité dans le temps du vecteur des valeurs des variables du système. C'est-à-dire qu'à chaque instant t de la fenêtre temporelle I, le vecteur $v(\mathrm{t})$, des valeurs que prennent les variables du système à l'instant t, existe et est unique. $B(cs, F_0, 0)$ peut donc être considéré comme une application de la fenêtre temporelle I vers l'ensemble produit $\overline{\mathrm{V}}$.

La deuxième condition signifie que cet ensemble $B(cs, F_0, 0)$ est construit suivant l'instruction cs. Cela signifie qu'à chaque instant t de la fenêtre temporelle I, le premier élément de cet unique vecteur $v(\mathrm{t})$ (i.e. : l'élément $p_1(v(\mathrm{t})) = c(t)$) est égal à la valeur $cs(t)$ de l'instruction à cet instant t.

La troisième condition signifie que l'ensemble $B(cs, F_0, 0)$ est construit suivant l'ensemble des équations du système en fonctionnement normal (i.e. : les équations de $\overline{S_{F_0}}$). Ce qui veut dire qu'à chaque instant t de la fenêtre temporelle $\mathrm{I} \setminus \{max(\mathrm{I})\}$, l'unique vecteur $v(\mathrm{t})$ doit satisfaire toutes les équations de $\overline{S_{F_0}}$.

4.2.3.2 Comportements avec défauts du système

Nous venons de voir qu'un comportement normal du système, suivant une instruction $cs \in Cons$ donnée, représente l'ensemble des valeurs que prennent les variables de ce système lorsqu'il fonctionne normalement suivant cette instruction cs. Un comportement avec défaut du système, toujours suivant une instruction $cs \in Cons$ donnée, va donc représenter de la même manière l'ensemble des valeurs que prennent les variables du système suivant cette instruction cs ; ce sera néanmoins lorsque le système fonctionne sous la présence d'un défaut. Pour un défaut $F \in \Gamma \setminus \{F_0\}$, il s'agira donc de l'ensemble des équations du système en fonctionnement avec le défaut F (i.e. : les équations de $\overline{S_F}$ données au chapitre 2).

Comportements du système sous la présence d'un défaut

Un comportement du système avec un défaut $F \in \Gamma \setminus \{F_0\}$ est considéré suivant une instruction $cs \in Cons$ donnée et pour une occurrence $\mathrm{t}_n \in \Omega_{(F,cs)}$ du défaut F. Nous le notons $B(cs, F, \mathrm{t}_n)$ et le qualifions de comportement du système suivant la consigne cs et sous la présence du défaut F à l'occurrence t_n. Il s'agit comme dans le cas normal de l'ensemble constitué par les vecteurs de données $v(\mathrm{t}) \in \overline{\mathrm{VD}}$ considérés à chaque instant $\mathrm{t} \in \mathrm{I}$ suivant la consigne cs et en fonction de l'ensemble des équations de $\overline{S_F}$ du système en fonctionnement avec le défaut F à l'occurrence t_n. La notation $B(cs, F, \mathrm{t}_n)$ reflète bien qu'il s'agit de la consigne $cs \in Cons$ et du défaut $F \in \Gamma \setminus \{F_0\}$ à l'occurrence $\mathrm{t}_n \in \Omega_{(F,cs)}$.

Formellement, $B(cs, F, \mathrm{t}_n)$ est défini comme dans le cas normal (suivant la définition donnée pour un comportement normal) mais où le point (3) est remplacé par la satisfaction de l'ensemble des équations du système en fonctionnement avec le défaut F à l'occurrence t_n (i.e. : les équations de $\overline{S_F}$).

Prenons par exemple un défaut $F \in \Gamma \setminus \{F_0\}$ de capteur. Comme indiqué au chapitre 3 sur la typologie des défauts, il se modélise par une perturbation des mesures y et se représente par $y_F(\mathrm{t}) = Pert_F(\mathrm{t}, y(\mathrm{t}), dft_{(F,\mathrm{t}_n)}(\mathrm{t}))$, avec $dft_{(F,\mathrm{t}_n)}$ désignant le comportement du défaut et $\mathrm{t}_n \in \mathbb{T}_\iota$ son occurrence. Donc, pour une consigne $cs \in Cons$ et ce défaut $F \in \Gamma \setminus \{F_0\}$ de capteur apparaissant à l'occurrence $\mathrm{t}_n \in \Omega_{(F,cs)}$, le comportement de défaut du système est l'ensemble $B(cs, F, \mathrm{t}_n) \subset \mathbb{T}_\iota \times \overline{\mathrm{VD}}$ des vecteurs de données vérifiant :

1. Existence et unicité dans le temps : pour tout instant $\mathrm{t} \in \mathrm{I}$, il existe un unique vecteur $v(\mathrm{t}) \in \overline{\mathrm{VD}}$ tel que $(\mathrm{t}, v(\mathrm{t})) \in B(cs, F, \mathrm{t}_n)$.
2. Construction suivant l'instruction cs : pour tout instant $\mathrm{t} \in \mathrm{I}$, $p_1(v(\mathrm{t})) = cs(\mathrm{t})$.
3. Satisfaction des équations de $\overline{S_F}$: pour tout instant $\mathrm{t} \in \mathrm{I} \setminus \{max(\mathrm{I})\}$,
   (a) $p_4(v(\mathrm{t}+1)) = f(p_4(v(\mathrm{t})), \theta, p_3(v(\mathrm{t})), p_6(v(\mathrm{t})))$
   (b) $p_5(v(\mathrm{t})) = Pert_F(\mathrm{t}, g(p_4(v(\mathrm{t}))), dft_{(F,\mathrm{t}_n)}(\mathrm{t}))$
   (c) $p_2(v(\mathrm{t}+1)) = h(p_1(v(\mathrm{t})), p_2(v(\mathrm{t})), p_5(v(\mathrm{t})))$
   (d) $p_3(v(\mathrm{t})) = k(p_2(v(\mathrm{t})))$
   (e) $p_4(v(0)) = x_{init}$
   (f) $p_2(v(0)) = a_{init}$
   (g) $p_9(v(\mathrm{t}+1)) = f^M(p_9(v(\mathrm{t})), \theta^M, p_8(v(\mathrm{t})), p_{11}(v(\mathrm{t})))$
   (h) $p_{10}(v(\mathrm{t})) = g^M(p_9(v(\mathrm{t})))$
   (i) $p_7(v(\mathrm{t}+1)) = h(p_1(v(\mathrm{t})), p_7(v(\mathrm{t})), p_{10}(v(\mathrm{t})))$
   (j) $p_8(v(\mathrm{t})) = k(p_7(v(\mathrm{t})))$
   (k) $p_9(v(0)) = x^M_{init}$
   (l) $p_7(v(0)) = a^M_{init}$

4.2.4 Comportements observables du système

Le comportement observable du système désigne sa manière de fonctionner telle que peut le « voir » un observateur extérieur, le futur diagnostiqueur dans notre cas. Il s'agit de la projection du comportement du système uniquement sur les variables observables du système (i.e. : l'ensemble des variables $\overline{\mathrm{V}}_{obs} = \{c; u; y; u^M; y^M\}$). Cette projection du comportement du système se considère donc uniquement sur l'ensemble $\overline{\mathrm{VD}}_{obs} = C \times U \times Y \times U^M \times Y^M$ du produit des domaines des variables observables. Rappelons que nous utilisons le terme « observable » dans le sens « mesuré » de l'automatique.

4.2.4.1 Comportements observables normaux du système

Un comportement normal du système est défini suivant une instruction $cs \in Cons$ de l'opérateur. En le considérant uniquement sur les variables observables du système, nous en obtenons un comportement observable normal. Formellement, pour un comportement normal $B(cs, F_0, 0)$ suivant une instruction $cs \in Cons$, le comportement observable normal $ObsB(cs, F_0, 0)$ sous-jacent est la projection de $B(cs, F_0, 0)$ sur l'ensemble produit $\mathbb{T}_\iota \times \overline{\mathrm{VD}}_{obs}$ des variables observables :

$$ObsB(cs, F_0, 0) = \mathrm{Pr}_{\mathbb{T}_\iota \times \overline{\mathrm{VD}}_{obs}}(B(cs, F_0, 0))$$

4.2.4.2 Comportements observables avec défauts du système

Un comportement du système sous la présence d'un défaut $F \in \Gamma \setminus \{F_0\}$ est défini suivant une instruction $cs \in Cons$ de l'opérateur et pour une occurrence $\mathrm{t}_n \in \Omega_{(F,cs)}$ de F. Comme pour un comportement observable normal du système, en considérant ce comportement sous la présence de F uniquement sur les variables observables du système, nous en obtenons un comportement observable sous la présence du défaut F. Formellement pour un comportement $B(cs, F, \mathrm{t}_n)$, suivant une instruction $cs \in Cons$ et sous la présence d'un défaut $F \in \Gamma \setminus \{F_0\}$ à une occurrence $\mathrm{t}_n \in \Omega_{(F,cs)}$, le comportement observable $ObsB(cs, F, \mathrm{t}_n)$ sous-jacent est la projection de $B(cs, F, \mathrm{t}_n)$ sur l'ensemble produit $\mathbb{T}_\iota \times \overline{\mathrm{VD}}_{obs}$ des variables observables :

$$ObsB(cs, F, \mathrm{t}_n) = \mathrm{Pr}_{\mathbb{T}_\iota \times \overline{\mathrm{VD}}_{obs}}(B(cs, F, \mathrm{t}_n))$$

4.2.4.3 Ensemble des comportements observables du système

Un comportement observable est déterminé en fonction d'une instruction $cs \in Cons$ et d'un défaut $F \in \Gamma$ à une occurrence $\mathrm{t}_n \in \Omega_{(F,cs)}$, avec bien sûr $\mathrm{t}_n = 0$ pour le cas normal $F_0 \in \Gamma$. Pour un défaut $F \in \Gamma \setminus \{F_0\}$, l'ensemble $ObsBeh_{Cons}(F)$ des comportements observables, suivant l'ensemble $Cons$ des instructions et sous la présence du défaut F, est donc la réunion de tous les comportements observables pour toutes les instructions $cs \in Cons$ et toutes les occurrences $\mathrm{t}_n \in \Omega_{(F,cs)}$ de F :

$$ObsBeh_{Cons}(F) = \bigcup_{cs \in Cons} \bigcup_{\mathrm{t}_n \in \Omega_{(F,cs)}} \{ObsB(cs, F, \mathrm{t}_n)\}$$

Pour le cas normal $F_0 \in \Gamma$, l'ensemble $ObsBeh_{Cons}(F_0)$ des comportements observables normaux suivant l'ensemble $Cons$ des instructions est la réunion de tous les comportements observables normaux pour toutes les instructions $cs \in Cons$:

$$ObsBeh_{Cons}(F_0) = \bigcup_{cs \in Cons} \{ObsB(cs, F_0, 0)\}$$

Par conséquent, l'ensemble des comportements observables suivant l'ensemble $Cons$ des instructions est la réunion de tous les ensembles des comportements observables pour tous les défauts potentiels de l'ensemble Γ :

$$ObsBeh_{Cons} = \bigcup_{F \in \Gamma} ObsBeh_{Cons}(F)$$

4.2.4.4 Domaine temporel d'un comportement observable

Le domaine temporel d'un comportement observable du système désigne la fenêtre temporelle sur laquelle il est défini , c'est-à-dire le sous-ensemble du temps $\mathbb{T}_\iota$ pour lequel il peut être considéré. Bien entendu, il s'agit à chaque fois de la fenêtre temporelle I. Pour un comportement observable quelconque $Obs \in ObsBeh_{Cons}$ du système, son domaine temporel est noté $TDom(Obs)$ et est égal à la fenêtre temporelle I.

4.3 Diagnosticabilité du système

Comme nous l'avons expliqué, l'étude de la diagnosticabilité d'un système consiste à s'assurer qu'en fonctionnement le diagnostiqueur sera toujours capable de détecter et d'isoler sans ambiguïté n'importe quel défaut préalablement répertorié lorsqu'il apparaît. Nous avons expliqué que cette étude de la diagnosticabilité est inhérente au procédé d'analyse du diagnostiqueur et qu'elle doit donc se mener identiquement à la manière dont le diagnostiqueur va analyser le comportement observé du système : c'est-à-dire la vérification des validités des propriétés de bon ou mauvais fonctionnements par le flux de données constitué par le vecteur des valeurs des variables observables du système à chaque instant du temps. C'est donc relativement à une famille $\Lambda = (\mathrm{P}_F)_{F\in\Gamma}$ de propriétés, que nous avons nommé une *caractérisation de défauts* et qui caractérise le comportement observé du système sous la présence ou non d'un défaut, que va se réaliser l'étude de la diagnosticabilité. Cela signifie non seulement que lorsqu'il n'y a pas présence de défaut, la propriété de bon fonctionnement doit toujours être vraie, mais aussi que lorsqu'il y a présence d'un défaut, alors cette propriété de bon fonctionnement ne doit plus être vraie et seule la propriété du défaut doit être vraie.

Pour chacun des défauts $F \in \Gamma$, sa propriété P_F est évaluée à chaque instant du temps par le comportement observé *Obs* du système. Elle est dite *validée* (ou valide) par ce comportement observé à un instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps si sa valeur, évaluée par ce comportement observé à cet instant t, est vraie ; ce que nous noterons $\mathrm{P}_F(Obs, \mathrm{t}) = \texttt{True}$. Dans le cas contraire, elle est dite *invalidée* (ou invalide) et nous le noterons $\mathrm{P}_F(Obs, \mathrm{t}) = \texttt{False}$.

Du point de vue de la diagnosticabilité, l'étude se réalise en considérant les comportements observables du système, que nous venons de définir et qui sont obtenus par utilisation des modèles de bon fonctionnement et de défauts. Les propriétés P_F, pour chacun des défauts $F \in \Gamma$, vont donc être évaluées par n'importe quel comportement observable $Obs \in ObsBeh_{Cons}$. Un défaut sera donc dit *diagnosticable*, relativement à une caractérisation de défauts Λ, si dès qu'il apparaît, la propriété P_{F_0} du bon fonctionnement devient invalide puis ensuite seule la propriété P_F du défaut F considéré est valide. Bien entendu, cette définition intuitive ne prend pas rigoureusement en compte non seulement l'aspect temporel entre l'occurrence du défaut et sa diagnosticabilité, mais aussi notre choix de considérer un diagnostic en temps borné après l'occurrence du défaut.

Pour le moment, nous allons juste supposer qu'il est possible de définir une telle caractérisation de défauts $\Lambda = (\mathrm{P}_F)_{F\in\Gamma}$. Nous supposons donc que chaque propriété P_F existe et caractérise plus ou moins parfaitement le comportement observable du système sous la présence du défaut $F \in \Gamma$ considéré. Nous verrons dans la partie suivante comment construire de telles caractérisations de défauts, notamment comment apprendre les propriétés grâce à l'utilisation des comportements observables.

Dans cette partie, nous allons donc rigoureusement définir la notion de diagnosticabilité d'un défaut. Nous verrons ensuite que cette notion peut être divisée en trois sous-notions dont la conjonction est une condition nécessaire mais pas suffisante pour conclure à la diagnosticabilité d'un défaut. Ce résultat nous sera alors utile par la suite pour justement étudier la diagnosticabilité des défauts de manière « progressive ».

4.3.1 La notion de diagnosticabilité des défauts

La diagnosticabilité d'un défaut consiste à s'assurer qu'après son occurrence et avant la borne b de diagnostic, non seulement la propriété du bon fonctionnement devient invalide, mais aussi qu'ensuite seule la propriété du défaut est valide. Comme les validités des différentes propriétés sont obtenues en les évaluant par les comportements observables du système, la diagnosticabilité se définit suivant ces comportements observables. Un défaut $F \in \Gamma \setminus \{F_0\}$ sera donc diagnosticable si pour chacun de ses comportements observables, alors à son occurrence t_n et avant la borne b de diagnostic, la propriété normale P_{F_0} devient invalide à un instant de détection $\mathrm{t}_k \geq \mathrm{t}_n$, puis suite à cette détection et après le

délai h d'isolation, seule la propriété du défaut P_F du défaut est valide jusqu'au délai δ de confiance (i.e. : valide sur la fenêtre temporelle $[\mathrm{t}_k + h; \mathrm{t}_k + (h + \delta)]$).

Rappelons que nous avons considéré le cas normal F_0 comme faisant partie de l'ensemble Γ des défauts potentiels. Or la manière dont nous venons de décrire la diagnosticabilité n'est pas applicable à ce cas normal. Il faudrait en effet qu'après l'occurrence t_n de F_0, qui rappelons-le vaut 0, sa propriété devienne à la fois valide et invalide ; ce qui est totalement absurde. Pour ce cas normal, il faut donc que tous les comportements observables normaux valident la propriété normale. Par conséquent nous allons définir différemment cette notion de diagnosticabilité selon le défaut considéré : un « vrai » défaut $F \in \Gamma \setminus \{F_0\}$ ou le cas normal $F_0 \in \Gamma$.

Diagnosticabilité d'un défaut Comme nous venons juste de l'indiquer, un « vrai » défaut $F \in \Gamma \setminus \{F_0\}$ est diagnosticable si après chacune de ses occurrences et dans une fenêtre temporelle bornée par la borne b de diagnostic, la propriété normale devient invalide et ensuite seule la propriété de ce défaut reste valide jusqu'au délai δ de confiance.

Cela signifie que pour tout comportement observable $ObsB(cs, F, \mathrm{t}_n)$ sous la présence de ce défaut F, donc pour n'importe quelle instruction $cs \in Cons$ et n'importe quelle occurrence $\mathrm{t}_n \in \Omega_{(F,cs)}$ de F, alors entre son occurrence t_n et avant la borne b de diagnostic il doit y avoir :

- la propriété normale P_{F_0} passe de valide à invalide, il y a détection d'un défaut (ou au moins d'un mauvais comportement) ;
- suite à cette détection et après un temps d'attente h de décision, seule la propriété P_F de ce défaut reste valide jusqu'au délai δ de confiance, il y a détection du défaut F.

La figure 4.4 suivante représente cette notion de diagnosticabilité d'un défaut $F \in \Gamma \setminus \{F_0\}$. Le graphique supérieur représente un comportement observable $ObsB(cs, F, \mathrm{t}_n)$ sous la présence du défaut F à l'occurrence t_n, toujours uniquement représenté que les évolutions temporelles de la consigne c (représentée par le trait discontinu vert) et de la mesure y (représentée par le trait continu bleu). Ce comportement est qualifié de défaut car à l'occurrence t_n, la mesure y diverge fortement de la consigne c pour se fixer à une valeur arbitraire. Le graphique inférieur représente les évolutions temporelles des validités des différentes propriétés P_{F_i} (pour tous les défauts $F_i \in \Gamma$, y compris le défaut F et le cas normal F_0) suivant ce comportement observable $ObsB(cs, F, \mathrm{t}_n)$. Le trait vert représente la validité de la propriété normale P_{F_0}, le trait rouge celle de la propriété P_F du défaut F considéré et le trait bordeaux celles des propriétés $\mathrm{P}_{F'}$ des autres défauts $F' \in \Gamma \setminus \{F; F_0\}$.

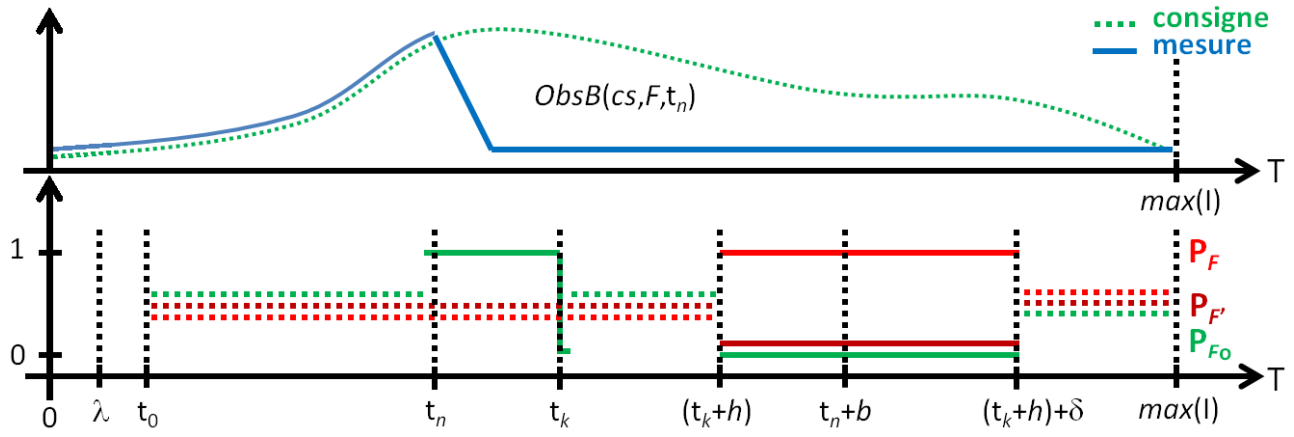

Figure 4.4 – Diagnosticabilité d'un défaut.

Ce graphique représente toutes les conditions que nous venons d'exposer :

- Le fait que la propriété normale P_{F_0} passe de valide à invalide signifie qu'il existe un instant t_k de détection supérieur ou égal à l'occurrence t_n de F et tel que P_{F_0} soit valide sur la fenêtre temporelle $[\mathrm{t}_n{}^-; \mathrm{t}_k[$, puis invalide à l'instant t_k. Cela signifie que pour tout $\mathrm{t} \in [\mathrm{t}_n{}^-; \mathrm{t}_k[$ alors $\mathrm{P}_{F_0}(ObsB(cs, F, \mathrm{t}_n), \mathrm{t}) = \texttt{True}$, et pour $\mathrm{t} = \mathrm{t}_k$ alors $\mathrm{P}_{F_0}(ObsB(cs, F, \mathrm{t}_n), \mathrm{t}) = \texttt{False}$. C'est ce que représente la ligne continue verte sur la fenêtre temporelle $[\mathrm{t}_n{}^-; \mathrm{t}_k]$.
- Le fait que seule la propriété P_F de ce défaut reste valide, suite à cette détection à l'instant t_k et après le délai h de décision, signifie qu'à partir de l'instant (t_k+h) et au moins jusque $(\mathrm{t}_k+h)+\delta$, où δ représente le délai de confiance, seule la propriété P_F de ce défaut reste valide et les autres sont invalides. C'est-à-dire que pour tout $\mathrm{t} \in [\mathrm{t}_k + h; \mathrm{t}_k + (h + \delta)]$ alors $\mathrm{P}_F(ObsB(cs, F, \mathrm{t}_n), \mathrm{t}) = \texttt{True}$ et $\mathrm{P}_{F'}(ObsB(cs, F, \mathrm{t}_n), \mathrm{t}) = \texttt{False}$ pour tout autre défaut $F' \in \Gamma\{F\}$, y compris pour le cas normal F_0. C'est ce que représentent les lignes continues verte, rouge et bordeaux sur la fenêtre temporelle $[\mathrm{t}_k + h; \mathrm{t}_k + (h + \delta)]$.
- Le fait que tout ceci se fasse avant la borne b de diagnostic signifie que l'instant t_k de détection vérifie $(\mathrm{t}_k + h) \leq (\mathrm{t}_n + b)$, c'est-à-dire que $\mathrm{t}_k \leq \mathrm{t}_n + (b - h)$. Comme nous imposons que l'instant de détection t_k soit supérieur ou égal à l'occurrence t_n du défaut, nous avons donc $\mathrm{t}_k \in [\mathrm{t}_n; \mathrm{t}_n + (b - h)]$.

Remarquons bien l'ordre dans lequel doivent se passer les différentes actions : il doit d'abord y avoir apparition du défaut à une occurrence $\mathrm{t}_n \in \Omega_{(F,cs)}$, puis détection de ce défaut à un instant t_k, et enfin, suite à cette détection, isolation du défaut à l'instant $\mathrm{t}_k + h$ avec uniquement validité de P_F sur la fenêtre temporelle $[\mathrm{t}_k + h; \mathrm{t}_k + (h + \delta)]$; cet instant t_k de détection devant nécessairement vérifier $\mathrm{t}_k \in [\mathrm{t}_n; \mathrm{t}_n + (b - h)]$ pour rendre compte que tout ceci doit se passer avant la borne b (i.e. : $(\mathrm{t}_k + h) \leq (\mathrm{t}_n + b)$).

Les validités des propriétés P_{F_i} (pour tous les défauts $F_i \in \Gamma$, y compris le défaut F et le cas normal F_0), durant la fenêtre temporelle $]\mathrm{t}_k; \mathrm{t}_k + h[$, ne nous intéressent pas. En effet et comme nous l'avons indiqué, il ne serait pas surprenant que, suite à l'instant t_k de détection (i.e. : lorsque la propriété P_{F_0} passe de valide à invalide) et durant la fenêtre temporelle $]\mathrm{t}_k; \mathrm{t}_k + h[$, les validités des propriétés P_{F_i} oscillent entre valide et invalide. Donc suite à un instant de détection t_k, la décision du diagnostiqueur ne se fera qu'après le délai h de décision, donc à l'instant $(\mathrm{t}_k + h)$.

De plus sur la fenêtre temporelle $]\mathrm{t}_n; \mathrm{t}_k[$ entre l'occurrence t_n du défaut et sa détection à l'instant t_k, ces validités des propriétés P_{F_i} ne nous intéressent pas non plus. Effectivement, suivant l'ordre des actions établi précédemment, nous cherchons d'abord à détecter un défaut avant de l'isoler. Ici encore, il ne serait pas surprenant que, suite à l'occurrence t_n du défaut et durant la fenêtre temporelle $]\mathrm{t}_n; \mathrm{t}_k[$, les validités des propriétés P_{F_i} oscillent entre valide et invalide.

Enfin avant l'occurrence t_n du défaut (i.e. : sur la fenêtre temporelle $]\mathrm{t}_0; \mathrm{t}_n[$) les validités des propriétés P_{F_i} de tous les défauts $F_i \in \Gamma$, y compris cette fois-ci le défaut F et le cas normal F_0, ne nous intéressent pas ici encore. D'une part, avant la période d'initialisation λ du diagnostiqueur, les validités de toutes les formules ne peuvent être évaluées. Comme nous avons supposé que l'instant t_0 est supérieur à λ et que toutes les instructions $cs \in Cons$ ne sont représentatives du fonctionnement du système qu'à partir de t_0, nous ne nous intéressons donc pas à ces validités sur la fenêtre temporelle initiale $[0; \mathrm{t}_0[$. De plus, concernant la validité de la propriété normale P_{F_0}, elle devrait normalement être assurée par la diagnosticabilité du cas normal que nous allons voir après. Ce sera dû au fait que par construction d'un comportement observable sous la présence d'un défaut $F \in \Gamma \setminus \{F_0\}$, le système fonctionne normalement avant l'occurrence de ce défaut, du moins pour les défauts autres que les faiblement progressifs. Pour la validité des autres propriétés, nous n'imposons rien de spécifique. Il pourrait néanmoins être possible d'imposer qu'elles soient invalides sur cette fenêtre temporelle.

Nous obtenons la définition de la diagnosticabilité d'un défaut $F \in \Gamma \setminus \{F_0\}$.

Définition 4.2 *Un défaut $F \in \Gamma \setminus \{F_0\}$ est* diagnosticable *si et seulement si pour toute instruction*

$cs \in Cons$ et pour toute occurrence $\mathrm{t}_n \in \Omega_{(F,cs)}$ de F, il existe un instant $\mathrm{t}_k \in \mathrm{I}$ tel que :

1. $\mathrm{t}_k \in [\mathrm{t}_n; \mathrm{t}_n + (b - h)]$.
2. *pour tout instant* $\mathrm{t} \in \mathrm{I}$, *si* $\mathrm{t} \in [\mathrm{t}_n{}^-; \mathrm{t}_k[$ *alors* $\mathrm{P}_{F_0}(ObsB(cs, F, \mathrm{t}_n), \mathrm{t}) = \texttt{True}$.
   pour tout instant $\mathrm{t} \in \mathrm{I}$, *si* $\mathrm{t} = \mathrm{t}_k$ *alors* $\mathrm{P}_{F_0}(ObsB(cs, F, \mathrm{t}_n), \mathrm{t}) = \texttt{False}$.
3. *pour tout instant* $\mathrm{t} \in \mathrm{I}$, *si* $\mathrm{t} \in [\mathrm{t}_k + h; \mathrm{t}_k + (h + \delta)]$ *alors* $\mathrm{P}_F(ObsB(cs, F, \mathrm{t}_n), \mathrm{t}) = \texttt{True}$.
4. *pour tout instant* $\mathrm{t} \in \mathrm{I}$, *si* $\mathrm{t} \in [\mathrm{t}_k + h; \mathrm{t}_k + (h + \delta)]$ *alors pour tout autre défaut* $F' \in \Gamma \setminus \{F\}$, $\mathrm{P}_{F'}(ObsB(cs, F, \mathrm{t}_n), \mathrm{t}) = \texttt{False}$.

Dans cette définition, le premier point exprime la diagnosticabilité du défaut F en temps borné par b après son occurrence t_n. Le deuxième point exprime la détection de F par la propriété P_{F_0}. Les troisième et quatrième points expriment l'isolation de F : le troisième concerne la propriété de F et le quatrième concerne les propriétés des autres défauts F' différents de F, y compris la propriété du cas normal F_0.

Diagnosticabilité du cas normal Le cas normal $F_0 \in \Gamma$ est diagnosticable si en l'absence de défaut la propriété normale P_{F_0} est toujours valide. C'est-à-dire que tout comportement observable normal doit valider cette propriété normale P_{F_0}.

En considérant un comportement observable normal, cela signifie que la propriété normale P_{F_0} doit toujours être valide sur la fenêtre temporelle $[\mathrm{t}_0; max(\mathrm{I})]$ où I représente le domaine temporel de définition du comportement et t_0 l'instant à partir duquel nous avons supposé que les instructions sont représentatives du fonctionnement du système. Rappelons que cet instant t_0 est supérieur à la période $\lambda \in \mathbb{T}_\iota$ d'initialisation nécessaire au diagnostiqueur. Ainsi la propriété normale P_{F_0} pourra de ce fait tout à fait être évaluée à partir de cet instant t_0.

La figure 4.5 suivante représente cette notion de diagnosticabilité du cas normal F_0. Le graphique supérieur représente uniquement la consigne c (représentée par le trait discontinu vert) et la mesure y (représentée par le trait continu bleu). Ce comportement est qualifié de normal car cette mesure y suit « correctement » la consigne c durant toute la fenêtre temporelle I. Le graphique inférieur représente l'évolution temporelle de la validité de la propriété normale P_{F_0} (la ligne verte) suivant ce comportement observable normal $ObsB(cs, F_0, 0)$. Nous remarquons bien que la propriété normale P_{F_0} est toujours valide à partir de l'instant t_0 et jusque la borne maximum $max(\mathrm{I})$: c'est-à-dire que pour tout instant $\mathrm{t} \in [\mathrm{t}_0; max(\mathrm{I})]$ alors $\mathrm{P}_{F_0}(ObsB(cs, F, \mathrm{t}_n), \mathrm{t}) = \texttt{True}$.

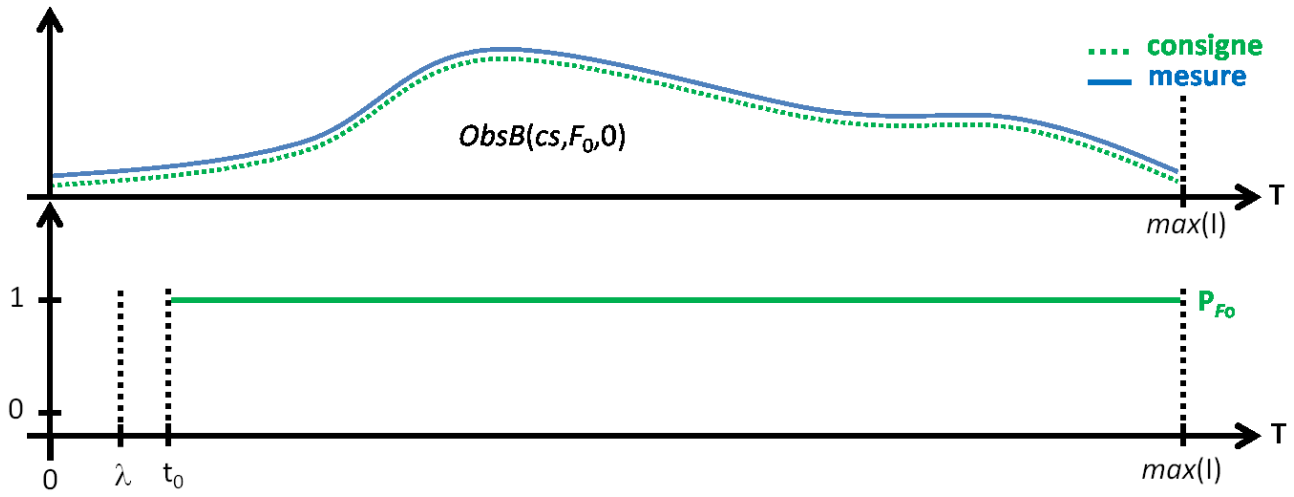

Figure 4.5 – Diagnosticabilité du cas normal.

Nous obtenons la définition de la diagnosticabilité du cas normal F_0.

Définition 4.3 *Le cas normal* $F_0 \in \Gamma$ *est* diagnosticable *si et seulement si pour toute instruction* $cs \in Cons$ *et pour tout instant* $\mathrm{t} \in \mathrm{I}$*, si* $\mathrm{t} \in [\mathrm{t}_0; max(\mathrm{I})]$ *alors* $\mathrm{P}_{F_0}(ObsB(cs, F, \mathrm{t}_n), \mathrm{t}) = \texttt{True}$.

Nous utilisons le terme *diagnosticable* pour ce cas normal $F_0 \in \Gamma$ afin de signifier que même sans présence de défaut, le diagnostiqueur doit s'assurer qu'il n'y en ait pas.

4.3.2 Les notions d'éligibilité, de détectabilité et d'isolabilité

Nous venons de définir formellement la notion de diagnosticabilité d'un défaut, que ce soit pour le cas normal $F_0 \in \Gamma$ ou pour un « vrai » défaut $F \in \Gamma \setminus \{F_0\}$. En analysant la définition pour les « vrais » défauts, nous remarquons qu'il est possible de la diviser en plusieurs parties : une première partie décrivant l'évolution de la validité de la propriété du défaut considéré, une deuxième décrivant l'évolution de la validité de la propriété normale et une troisième décrivant les évolutions des validités des propriétés des autres défauts. Pour le cas normal, il va s'agir d'une seule partie car la propriété du défaut considéré est la propriété normale.

Ces trois parties vont définir les notions d'*éligibilité*, de *détectabilité* et d'*isolabilité* d'un défaut. L'éligibilité d'un défaut va ainsi rendre compte du fait que sa propriété soit valide après son occurrence. La détectabilité d'un défaut va, quant à elle, rendre compte du fait que la propriété normale devienne invalide après son occurrence. Enfin l'isolabilité d'un défaut va rendre compte du fait que les propriétés des autres défauts soient invalides après son occurrence.

L'introduction de ces trois notions est motivée par le fait qu'en étude de diagnosticabilité, il est important de connaître pour quelle(s) raison(s) un défaut n'est pas diagnosticable.

- Est-ce la propriété normale qui ne permette pas de détecter le défaut ?
- Est-ce la propriété du défaut ou les propriétés des autres défauts qui ne permettent pas de l'isoler ?
- Est-ce la borne b de diagnostic ou le délai h d'isolation qui sont trop courts ou longs ?
- Est-ce l'information disponible (i.e. : les comportements observables issus des modèles de fonctionnement du système) qui n'est pas assez explicite sur les effets observables du défaut ?

Nous allons voir que ces trois notions vont permettre de rendre compte des trois premiers cas de non-diagnosticabilité. Le quatrième sera, pour sa part, mis en évidence lors de la présentation des caractérisations de défauts.

4.3.2.1 Éligibilité d'un défaut

L'*éligibilité* d'un défaut va décrire l'évolution de la validité de la propriété de ce défaut. Nous cherchons ainsi à rendre compte que la propriété du défaut considéré « reconnaisse » bien son défaut dans un temps borné dès qu'il apparaît : c'est-à-dire de manière intuitive qu'après chacune de ses occurrences sa propriété soit valide dans un temps borné par b après l'occurrence.

Plus précisément, un défaut est éligible si et seulement si chacun de ses comportements observables valide sa propriété dans un temps borné par b après son occurrence et jusqu'au délai de confiance δ. Pour le cas normal, cela signifie que chacun de ses comportements observables valide sa propriété (i.e. : la propriété normale) sur tout le domaine temporel du comportement modulo l'instant t_0 : c'est-à-dire sur la fenêtre temporelle $[\mathrm{t}_0; max(\mathrm{I})]$.

Cette notion d'éligibilité va donc être donnée selon le défaut considéré : un « vrai » défaut $F \in \Gamma \setminus \{F_0\}$ ou le cas normal $F_0 \in \Gamma$.

Éligibilité d'un défaut La définition suivante décrit formellement cette notion d'éligibilité d'un « vrai » défaut $F \in \Gamma \setminus \{F_0\}$:

Définition 4.4 *Un défaut* $F \in \Gamma \setminus \{F_0\}$ *est* éligible *si et seulement si pour toute instruction* $cs \in Cons$ *et pour toute occurrence* $\mathrm{t}_n \in \Omega_{(F,cs)}$ *de* F*, il existe un instant* $\mathrm{t}_{ke} \in \mathrm{I}$ *tel que :*

1. $\mathrm{t}_{ke} \in [\mathrm{t}_n; \mathrm{t}_n + (b - h)]$.
2. *pour tout instant* $\mathrm{t} \in \mathrm{I}$*, si* $\mathrm{t} \in [\mathrm{t}_{ke} + h; \mathrm{t}_{ke} + (h + \delta)]$ *alors* $\mathrm{P}_F(ObsB(cs, F, \mathrm{t}_n), \mathrm{t}) = \texttt{True}$.

La figure 4.6 suivante représente cette notion d'éligibilité d'un défaut $F \in \Gamma \setminus \{F_0\}$. Le graphique supérieur représente un comportement observable $ObsB(cs, F, \mathrm{t}_n)$ sous la présence du défaut F à une occurrence $\mathrm{t}_n \in \Omega_{(F,cs)}$. Le graphique inférieur représente l'évolution temporelle de la validité de la propriété P_F de ce défaut F suivant ce comportement observable $ObsB(cs, F, \mathrm{t}_n)$. À partir de l'occurrence t_n du défaut F, il existe un instant t_{ke} tel que la propriété P_F devient valide à partir de l'instant $(\mathrm{t}_{ke} + h)$, donc avant la borne b, et jusqu'au délai de confiance δ (i.e. : jusqu'à l'instant $(\mathrm{t}_{ke} + h) + \delta$). C'est ce que représente le trait rouge.

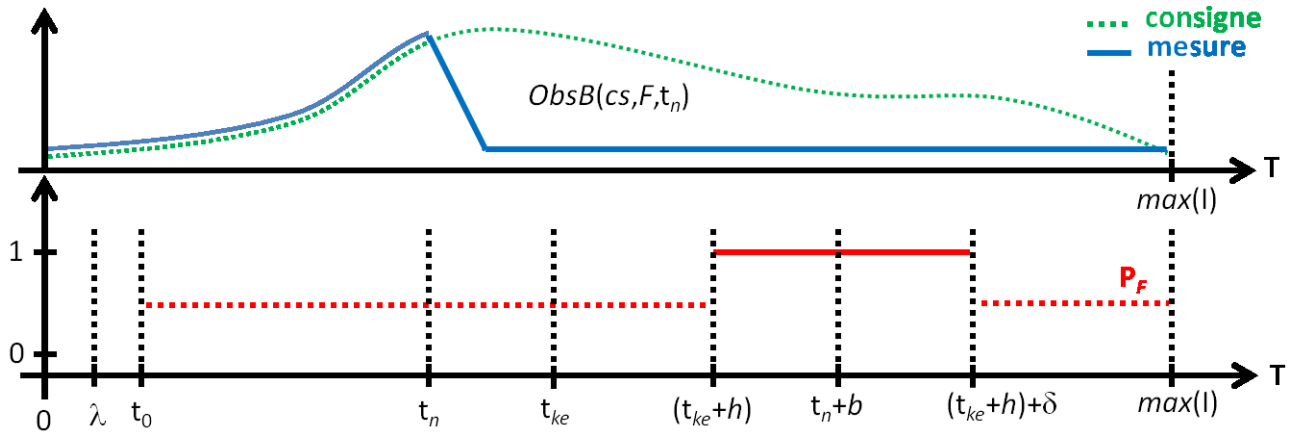

Figure 4.6 – Éligibilité d'un défaut.

Éligibilité du cas normal La définition suivante décrit formellement cette notion d'éligibilité pour le cas normal $F_0 \in \Gamma$:

Définition 4.5 *Le cas normal* $F_0 \in \Gamma$ *est* éligible *si et seulement si pour toute instruction* $cs \in Cons$ *et pour tout instant* $\mathrm{t} \in \mathrm{I}$*, si* $\mathrm{t} \in [\mathrm{t}_0; max(\mathrm{I})]$ *alors* $\mathrm{P}_{F_0}(ObsB(cs, F, \mathrm{t}_n), \mathrm{t}) = \texttt{True}$.

Remarquons que pour ce cas normal $F_0 \in \Gamma$, cette définition d'éligibilité est identique à la définition de diagnosticabilité.

4.3.2.2 Détectabilité d'un défaut

La *détectabilité* d'un défaut va décrire l'évolution de la validité de la propriété normale. Dire qu'un défaut est détectable signifie intuitivement qu'il est possible de le détecter : c'est-à-dire qu'il est possible de dire que le comportement observable du système n'est pas normal. Remarquons que suivant cette idée, la détectabilité ne peut donc concerner que les « vrais » défauts, les défauts autres que le cas normal.

Avec la détectabilité, nous cherchons à rendre compte que la propriété normale ne « reconnaisse » pas le défaut considéré dans un temps borné dès qu'il apparaît : c'est-à-dire de manière intuitive qu'après chacune des occurrences du défaut, la propriété normale passe de valide à invalide dans un

temps borné par b après l'occurrence. En étant plus précis, un défaut est détectable si et seulement si chacun de ses comportements observables fait passer la propriété normale de valide à invalide dans un temps borné par b après son occurrence. La définition suivante décrit formellement cette notion de détectabilité d'un défaut.

Définition 4.6 *Un défaut* $F \in \Gamma \setminus \{F_0\}$ *est* détectable *si et seulement si pour toute instruction* $cs \in Cons$ *et pour toute occurrence* $\mathrm{t}_n \in \Omega_{(F,cs)}$ *de* F*, il existe un instant* $\mathrm{t}_{kd} \in \mathrm{I}$ *tel que :*

1. $\mathrm{t}_{kd} \in [\mathrm{t}_n; \mathrm{t}_n + (b - h)]$.
2. *pour tout instant* $\mathrm{t} \in \mathrm{I}$*, si* $\mathrm{t} \in [\mathrm{t}_n{}^-; \mathrm{t}_{kd}[$ *alors* $\mathrm{P}_{F_0}(ObsB(cs, F, \mathrm{t}_n), \mathrm{t}) = \texttt{True}$.
   pour tout instant $\mathrm{t} \in \mathrm{I}$*, si* $\mathrm{t} = \mathrm{t}_{kd}$ *alors* $\mathrm{P}_{F_0}(ObsB(cs, F, \mathrm{t}_n), \mathrm{t}) = \texttt{False}$.

La figure 4.7 suivante représente cette notion de détectabilité d'un défaut $F \in \Gamma \setminus \{F_0\}$. Le graphique supérieur représente un comportement observable $ObsB(cs, F, \mathrm{t}_n)$ sous la présence du défaut $F \in \Gamma \setminus \{F_0\}$ à une occurrence $\mathrm{t}_n \in \Omega_{(F,cs)}$. Le graphique inférieur représente l'évolution temporelle de la validité de la propriété normale P_{F_0} suivant ce comportement observable $ObsB(cs, F, \mathrm{t}_n)$. La propriété normale P_{F_0} est toujours valide entre l'occurrence $\mathrm{t}_n \in \Omega_{(F,cs)}$ du défaut et l'instant t_{kd} de détection (i.e. : valide sur toute la fenêtre temporelle $[\mathrm{t}_n; \mathrm{t}_{kd}[$). À l'instant de détection t_{kd}, P_{F_0} doit être invalide. C'est ce que représente le trait vert.

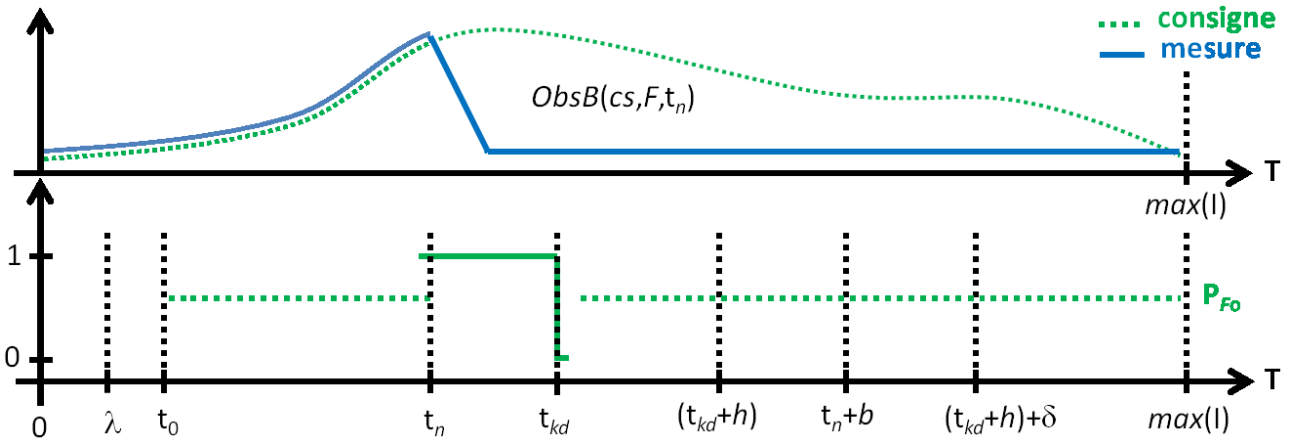

Figure 4.7 – Détectabilité d'un défaut.

Nous comprenons bien ici pourquoi cette définition de détectabilité n'est donnée que pour les défauts autres que le cas normal F_0. En effet la propriété normale doit reconnaître le cas normal, et dire que le cas normal est détectable signifierait donc que la propriété normale ne reconnaitrait pas le cas normal. Cela serait donc absurde.

4.3.2.3 Isolabilité d'un défaut

L'*isolabilité* d'un défaut va décrire les évolutions des validités des propriétés des autres défauts. Dire qu'un défaut est isolable signifie intuitivement qu'il est possible de l'isoler des autres défauts : c'est-à-dire qu'il est possible de dire que le comportement observable du système correspond à un défaut précis. Remarquons, comme pour la notion de détectabilité, que suivant cette idée, l'isolabilité ne doit elle aussi concerner que les « vrais » défauts.

Avec l'isolabilité, nous cherchons à rendre compte que les propriétés des autres défauts qu'un défaut considéré ne le « reconnaissent » pas dans un temps borné dès qu'il apparaît. Cela signifie, de manière intuitive, qu'après chacune de ses occurrences, les propriétés des autres défauts soient invalides dans

un temps borné par b après l'occurrence. En étant plus précis, un défaut est isolable si et seulement si chacun de ses comportements observables rend les propriétés des autres défauts invalides dans un temps borné par b après son occurrence et ce jusqu'au délai de confiance δ. Dans les propriétés des autres défauts, nous incluons aussi la propriété normale afin de nous assurer que le système ne revienne pas à un comportement normal. La définition suivante décrit formellement cette notion d'isolabilité d'un défaut.

Définition 4.7 *Un défaut $F \in \Gamma \backslash \{F_0\}$ est* isolable *si et seulement si pour toute instruction $cs \in Cons$ et pour toute occurrence $\mathrm{t}_n \in \Omega_{(F,cs)}$ de F, il existe un instant $\mathrm{t}_{ki} \in \mathrm{I}$ tel que :*

1. $\mathrm{t}_{ki} \in [\mathrm{t}_n; \mathrm{t}_n + (b - h)]$.
2. *pour tout instant* $\mathrm{t} \in \mathrm{I}$, *si* $\mathrm{t} \in [\mathrm{t}_{ki} + h; \mathrm{t}_{ki} + (h + \delta)]$ *alors pour tout autre défaut* $F' \in \Gamma \setminus \{F\}$, $\mathrm{P}_{F'}(ObsB(cs, F, \mathrm{t}_n), \mathrm{t}) = \texttt{False}$.

La figure 4.8 suivante représente cette notion d'isolabilité d'un défaut $F \in \Gamma \setminus \{F_0\}$. Le graphique supérieur représente un comportement observable $ObsB(cs, F, \mathrm{t}_n)$ sous la présence d'un défaut F à une occurrence $\mathrm{t}_n \in \Omega_{(F,cs)}$. Le graphique inférieur représente les évolutions temporelles des validités des propriétés $\mathrm{P}_{F'}$ des défauts $F' \in \Gamma \setminus \{F\}$ suivant ce comportement observable. À partir de l'occurrence t_n du défaut F, il existe un instant t_{ki} tel que toutes les propriétés $\mathrm{P}_{F'}$, pour n'importe quel défaut $F' \in \Gamma \setminus \{F\}$ y compris le cas normal F_0, sont invalides à partir de l'instant $(\mathrm{t}_{ki} + h)$ (donc avant la borne b) et jusqu'au délai de confiance δ (i.e. : jusqu'à l'instant $(\mathrm{t}_{ki} + h) + \delta$). C'est ce que représente les traits vert, pour la propriété normale P_{F_0}, et bordeaux pour les propriétés $\mathrm{P}_{F'}$ des autres défauts $F' \in \Gamma \setminus \{F; F_0\}$.

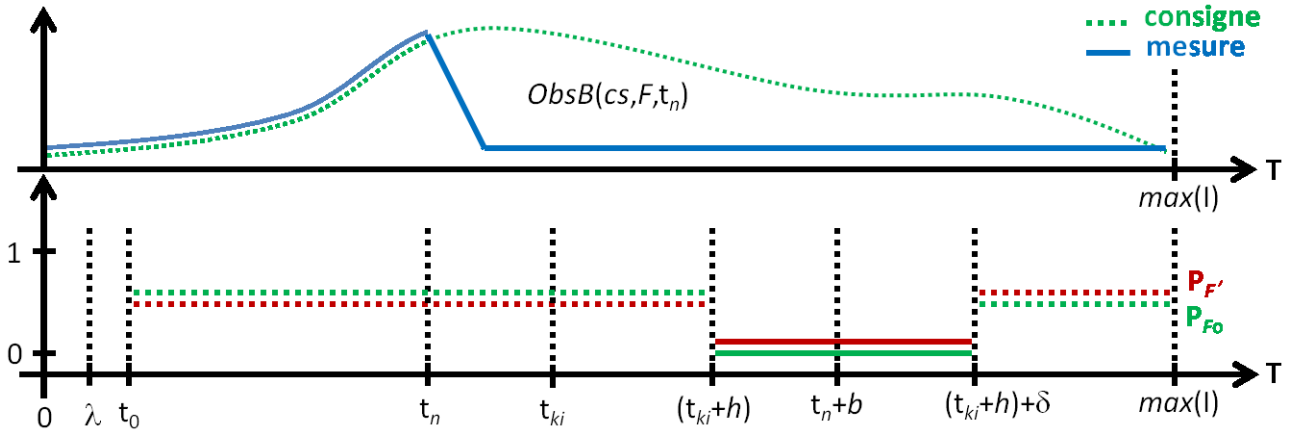

Figure 4.8 – Isolabilité d'un défaut.

Nous aurions pu imposer cette définition d'isolabilité au cas normal F_0. C'est-à-dire que tous les comportements observables normaux rendent toutes les propriétés des autres défauts $F \in \Gamma \setminus \{F_0\}$ invalides durant la fenêtre temporelle $[\mathrm{t}_0; max(\mathrm{I})]$. Cependant, nous pensons que cela aurait engendré plus de contraintes sur les propriétés des défauts. Par ailleurs, lorsque le système fonctionne et dans le but de ne pas surcharger le calculateur du diagnostiqueur, nous supposerons que seule la propriété normale sera vérifiée.

4.3.3 Rapport entre les différentes notions

Nous venons de définir la notion de diagnosticabilité et trois notions dérivées de celle-ci : l'éligibilité, la détectabilité et l'isolabilité. Pour le cas normal $F_0 \in \Gamma$, la diagnosticabilité est équivalente

à l'éligibilité car la détectabilité et l'isolabilité sont définies uniquement pour de « vrais » défauts $F \in \Gamma \setminus \{F_0\}$.

Comme ces trois notions sont dérivées de la diagnosticabilité, il est clair que si un défaut est diagnosticable, il sera alors éligible, détectable et isolable. Par contre la réciproque est vraie uniquement lorsque les instants d'éligibilité de détectabilité et d'isolabilité sont tous égaux, c'est-à-dire que $t_{ke} = t_{kd} = t_{ki}$. En effet, d'une part si l'un au moins des instants d'éligibilité t_{ke} ou d'isolabilité t_{ki} est supérieur à l'instant de détectabilité t_{kd} (i.e. : $t_{kd} < t_{ke}$ ou $t_{kd} < t_{ki}$), alors rien ne garantit qu'il y aura éligibilité ou isolabilité, selon que ce soit $t_{kd} < t_{ke}$ ou $t_{kd} < t_{ki}$, à l'instant de décision $t_{kd}+h$. Cela car l'éligibilité est garantie à l'instant $t_{ke}+h$ et l'isolabilité est garantie à l'instant $t_{ki}+h$ qui peuvent potentiellement être supérieurs à l'instant $t_{kd}+h$. D'autre part si l'un au moins des instants d'éligibilité t_{ke} ou d'isolabilité t_{ki} est inférieur à l'instant de détectabilité t_{kd} (i.e. : $t_{kd} > t_{ke}$ ou $t_{kd} > t_{ki}$), alors rien ne garantie non plus qu'il y ait éligibilité ou isolabilité, selon que ce soit $t_{kd} > t_{ke}$ ou $t_{kd} > t_{ki}$, durant la fenêtre temporelle $[t_{kd}+h; t_{kd}+(h+\delta)]$

Le fait d'imposer $t_{ke} = t_{kd} = t_{ki}$ est donc important pour le passage au diagnostiqueur car en fonctionnement, celui-ci ne connaîtra pas l'instant d'occurrence $t_n \in \mathbb{T}_\iota$ d'un défaut F quelconque. Rappelons que son seul repère sera par conséquent l'instant t_{kd} de détection. Ainsi et comme nous venons de l'expliquer, si $t_{ke} \neq t_{kd}$ ou $t_{ki} \neq t_{kd}$ alors il pourra être possible qu'à l'instant $t_{kd}+h$ les conditions d'éligibilité et/ou d'isolabilité ne soient pas satisfaites. Le diagnostiqueur ne pourra donc pas rendre son verdict à l'instant $t_{kd}+h$. Ceci impose donc bien que $t_{ke} = t_{kd} = t_{ki}$.

Cette équivalence modulo la condition d'égalité des instants d'éligibilité, de détectabilité et d'isolabilité est donnée formellement par la proposition suivante.

Proposition 4.1 *Un défaut $F \in \Gamma \setminus \{F_0\}$ est diagnosticable si et seulement s'il est éligible, détectable et isolable avec à chaque fois $t_{ke} = t_{kd} = t_{ki}$.*

Le sens de « à chaque fois » signifie que pour toute instruction $cs \in Cons$ et pour toute occurrence $t_n \in \Omega_{(F,cs)}$ de F, il existe des instants t_{ke} d'éligibilité, t_{kd} de détectabilité et t_{ki} d'isolabilité tels que $t_{ke} = t_{kd} = t_{ki}$. Par ailleurs, cette proposition ne concerne pas le cas normal F_0 car celui-ci n'est pas considéré dans les définitions de détectabilité et d'isolabilité.

Démonstration :
Nous devons montrer que si $F \in \Gamma \setminus \{F_0\}$ est diagnosticable, alors F est éligible, détectable et isolable avec à chaque fois $t_{ke} = t_{kd} = t_{ki}$, et inversement que si $F \in \Gamma \setminus \{F_0\}$ est éligible, détectable et isolable avec à chaque fois $t_{ke} = t_{kd} = t_{ki}$, alors F est diagnosticable.

A) Considérons un défaut $F \in \Gamma \setminus \{F_0\}$ en supposant qu'il soit diagnosticable et montrons alors qu'il est éligible, détectable et isolable avec à chaque fois $t_{ke} = t_{kd} = t_{ki}$. Considérons une consigne $cs \in Cons$ et une occurrence $t_n \in \Omega_{(F,cs)}$ de F. Par définition de la diagnosticablité de F, il existe un instant $t_k \in I$ tel que :

(a) $t_k \in [t_n; t_n + (b-h)]$

(b) pour tout instant $t \in I$, si $t \in [t_n{}^-; t_k[$ alors $P_{F_0}(ObsB(cs, F, t_n), t) =$ `True`, et si $t = t_k$ alors $P_{F_0}(ObsB(cs, F, t_n), t) =$ `False`.

(c) pour tout instant $t \in I$, si $t \in [t_k+h; t_k+(h+\delta)]$ alors $P_F(ObsB(cs, F, t_n), t) =$ `True`.

(d) pour tout instant $t \in I$, si $t \in [t_k+h; t_k+(h+\delta)]$ alors pour tout défaut $F' \in \Gamma \setminus \{F\}$, $P_{F'}(ObsB(cs, F, t_n), t) =$ `False`.

En posant $t_{ke} = t_{kd} = t_{ki} = t_k$, nous avons donc :

(1) clairement $t_{ke} = t_{kd} = t_{ki}$.

(2) par (a) et comme $t_{ke} = t_k$, nous avons donc $t_{ke} \in [t_n; t_n + (b-h)]$. De plus par (c) et comme $t_{ke} = t_k$, pour tout instant $t \in I$, si $t \in [t_{ke}+h; t_{ke}+(h+\delta)]$ alors $P_F(ObsB(cs, F, t_n), t) =$ `True`. F est donc éligible.

(3) par (a) et comme $t_{kd} = k$, nous avons donc $t_{kd} \in [t_n; t_n + (b-h)]$. De plus par (b) et comme $t_{kd} = t_k$, pour tout instant $t \in I$, non seulement si $t \in [t_n{}^-; t_{kd}[$ alors $P_{F_0}(ObsB(cs, F, t_n), t) =$ `True`, mais de plus si $t = t_{kd}$ alors $P_{F_0}(ObsB(cs, F, t_n), t) =$ `False`. F est donc détectable.

(4) par (a) et comme $t_{ki} = t_k$, nous avons donc $t_{ki} \in [t_n; t_n + (b - h)]$. De plus par (d) et comme $t_{ki} = t_k$, pour tout instant $t \in I$, si $t \in [t_{ki} + h; t_{ki} + (h + \delta)]$ alors pour tout autre défaut $F' \in \Gamma \setminus \{F\}$, $P_{F'}(ObsB(cs, F, t_n), t) = \texttt{False}$. F est donc isolable.

B) Considérons un défaut $F \in \Gamma \setminus \{F_0\}$ en supposant qu'il soit éligible, détectable et isolable avec à chaque fois $t_{ke} = t_{kd} = t_{ki}$ et montrons alors qu'il est diagnosticable. Considérons une consigne $cs \in Cons$ et une occurrence $t_n \in \Omega_{(F,cs)}$ de F. Par définition de l'éligibilité, la détectabilité et l'isolabilité de F, il existe des instants $t_{ke} \in I$, $t_{kd} \in I$ et $t_{ki} \in I$ avec $t_{ke} = t_{kd} = t_{ki}$ tels que :

(1) $t_{ke} \in [t_n; t_n + (b - h)]$, $t_{kd} \in [t_n; t_n + (b - h)]$ et $t_{ki} \in [t_n; t_n + (b - h)]$

(2) pour tout instant $t \in I$, si $t \in [{t_n}^-; t_{kd}[$ alors $P_{F_0}(ObsB(cs, F, t_n), t) = \texttt{True}$.

(3) pour tout instant $t \in I$, si $t = t_{kd}$ alors $P_{F_0}(ObsB(cs, F, t_n), t) = \texttt{False}$.

(4) pour tout instant $t \in I$, si $t \in [t_{ke} + h; t_{ke} + (h + \delta)]$ alors $P_F(ObsB(cs, F, t_n), t) = \texttt{True}$.

(5) pour tout instant $t \in I$, si $t \in [t_{ki} + h; t_{ki} + (h + \delta)]$ alors pour tout autre défaut $F' \in \Gamma \setminus \{F\}$, $P_{F'}(ObsB(cs, F, t_n), t) = \texttt{False}$.

En posant $t_k = t_{kd}$, nous avons donc :

(a) $t_k \in [t_n; t_n + (b - h)]$ car par (1) $t_{kd} \in [t_n; t_n + (b - h)]$.

(b) comme $t_k = t_{kd}$, alors par (2) et (3) :
pour tout instant $t \in I$, si $t \in [{t_n}^-; t_k[$ alors $P_{F_0}(ObsB(cs, F, t_n), t) = \texttt{True}$;
pour tout instant $t \in I$, si $t = t_k$ alors $P_{F_0}(ObsB(cs, F, t_n), t) = \texttt{False}$.

(c) comme $t_{ke} = t_{kd}$ et $t_k = t_{kd}$ alors $t_{ke} = t_k$ et ainsi par (4) pour tout instant $t \in I$, si $t \in [t_k + h; t_k + (h + \delta)]$ alors $P_F(ObsB(cs, F, t_n), t) = \texttt{True}$

(d) comme $t_{ki} = t_{kd}$ et $t_k = t_{kd}$ alors $t_{ki} = t_k$ et ainsi par (5) pour tout instant $t \in I$, si $t \in [t_k + h; t_k + (h + \delta)]$ alors pour tout autre défaut $F' \in \Gamma \setminus \{F\}$, $P_{F'}(ObsB(cs, F, t_n), t) = \texttt{False}$.

F est donc diagnosticable.
□

Nous allons voir un corollaire très intéressant de cette proposition que nous utiliserons lors de l'étude de la diagnosticabilité :

Corollaire 4.1 *Si un défaut $F \in \Gamma \setminus \{F_0\}$ n'est pas éligible ou détectable ou isolable, alors il n'est pas diagnosticable.*

Ce corollaire se démontre facilement par contraposée du point (A) de la démonstration précédente. Nous utiliserons ce corollaire afin de ne pas avoir à faire une étude complète de diagnosticabilité de tous les défauts $F \in \Gamma$ répertoriés d'un système. Nous étudierons donc en premier lieu l'éligibilité du cas normal F_0 ainsi que la détectabilité des défauts $F \in \Gamma \setminus \{F_0\}$, ensuite nous étudierons uniquement la diagnosticabilité des défauts détectables. Ce choix de ne pas étudier la diagnosticabilité de tous les défauts est justifié par le fait que si un défaut n'est au préalable pas détectable, cela signifie que du point de vue de la caractérisation de défauts Λ considérée, le comportement observable du système sous la présence de ce défaut est identique au comportement observable du système en fonctionnement normal. Le diagnostiqueur, basé sur cette caractérisation de défauts utilisée lors de cette étude de la diagnosticabilité, ne pourra donc pas détecter un comportement anormal du système. Ce comportement du système sous la présence de ce défaut observé par le diagnostiqueur sera donc analysé comme normal.

4.3.4 Retour sur la diagnosticabilité des défauts faiblement progressifs

Les notions de diagnosticabilité ainsi que d'éligibilité, de détectabilité et d'isolabilité, ont été définies autant pour des défauts brusques ou fortement progressifs que pour des défauts faiblement progressifs représentant, entre autres, des usures du système. Or la différence entre un défaut faiblement progressif et un fortement progressif vient dans le coefficient de progression qui, pour un faiblement

progressif, est trop faible pour être représentatif suivant la borne b de diagnostic. En effet, cette borne b représentant la dynamique de réponse du diagnostiqueur, elle va donc être de l'ordre d'un petit multiple de la dynamique de réponse β du système (6 secondes par exemple pour un système ayant un temps de réponse de 3 secondes). Elle sera ainsi généralement trop courte par rapport au coefficient de progression du défaut, qui lui est très faible (de l'ordre de la journée à l'année suivant les systèmes et les défauts considérés). Nous avons de ce fait introduit l'occurrence d'un défaut faiblement progressif, non pas comme pour les autres défauts à l'instant où ils apparaissent, mais comme étant l'instant du temps pour lequel le défaut se situe à un certain niveau jugé critique pour le fonctionnement du système. Nous avons ainsi établi que pour des instructions ayant un domaine temporel I très court, la présence du défaut pourrait être considérée dès le début à un certain niveau. Elle pourrait même être constante au niveau critique dès ce début de domaine temporel I et l'utilisation d'un signal de défaut *ad-hoc* permettrait de fournir l'occurrence $\mathrm{t}_n \in \mathrm{I}$ du défaut qui est utilisé lors de l'étude de la diagnosticabilité.

4.3.4.1 Problème de non détection des défauts faiblement progressifs

En étudiant ces notions, notamment la détectabilité et la diagnosticabilité, nous pouvons remarquer que, suivant le niveau de présence considéré du défaut faiblement progressif, il ne sera généralement pas détectable. En effet pour qu'un défaut, pas nécessairement faiblement progressif, soit détectable, il faut que la propriété normale P_{F_0} passe de valide à invalide dans l'intervalle $[\mathrm{t}_n; \mathrm{t}_n + (b - h)] \subseteq \mathrm{I}$ (ce que représente le trait vert du second graphique de la figure 4.7 de la page 118). Or et comme nous venons de le dire, suivant le niveau de présence considéré d'un défaut faiblement progressif, cette propriété normale P_{F_0} risque de ne pas être préalablement valide avant cette occurrence t_n du défaut. Notons que c'est d'ailleurs ce que nous avons obtenu comme résultats pour les défauts d'usure (les encrassements et fuites) du cas d'étude.

Le problème que nous venons de présenter et que nous ne rencontrons pas avec les défauts brusques ou fortement progressifs, vient de la manière dont nous avons défini l'occurrence d'un défaut faiblement progressif par rapport à l'occurrence des autres défauts. Pour un défaut faiblement progressif, son occurrence est l'instant du temps pour lequel le défaut se situe à un certain niveau jugé critique pour le fonctionnement du système, alors que pour les autres défauts brusques ou fortement progressifs, l'occurrence est l'instant où il apparaît. Cela signifie donc que pour un de ces autres défauts brusques ou fortement progressifs, le système fonctionne normalement avant l'occurrence et il ne fonctionne sous la présence du défaut qu'à partir de l'occurrence. À l'inverse d'un défaut faiblement progressif où avant son occurrence, qui représente l'instant où le défaut se situe au niveau jugé critique pour le fonctionnement du système, le système ne fonctionne pas obligatoirement normalement. Cette définition d'occurrence d'un défaut faiblement progressif est différente car pour des instructions ayant un domaine temporel I très court, la longueur entre le moment où le défaut apparaît (qui peut même être à l'instant initial de mise en fonctionnement du système) et le moment où il atteint ce niveau jugé critique peut-être potentiellement très longue.

4.3.4.2 Une solution possible

Nous avons envisagé une solution permettant de résoudre ce problème, tout en gardant à l'esprit la remarque de [BJL+90] concernant ces défauts faiblement progressifs : les défauts évolutifs doivent être surveillés de façon périodique et à rythme adapté au processus de vieillissement du système pour permettre soit une réadaptation des fonctions de commande, soit une action de maintenance préventive appropriée. Cette solution consiste à supposer que le système fonctionne normalement avant toute occurrence d'un défaut quelconque, donc pas nécessairement faiblement progressif. Cela signifie donc que tout comportement observable quelconque $ObsB(cs, F, \mathrm{t}_n)$, donné suivant une instruction $cs \in Cons$ et sous la présence d'un défaut $F \in \Gamma \setminus \{F_0\}$ à une occurrence $\mathrm{t}_n \in \Omega_{(F,cs)}$, valide la

propriété normale P_{F_0} avant cette occurrence t_n : c'est-à-dire $\mathrm{P}_{F_0}(ObsB(cs, F, \mathrm{t}_n), \mathrm{t}) = \texttt{True}$ pour tout instant de temps $\mathrm{t} \in [\mathrm{t}_0; \mathrm{t}_n[$. Remarquons que cette hypothèse est superflue pour les défauts brusques ou fortement progressifs car selon la manière dont nous avons construit les comportements observables sous la présence de ces défauts, le système fonctionne normalement avant leurs occurrences.

En pratique, cette solution peut néanmoins apporter un problème d'« atteignabilité » de défauts diagnosticables. Prenons comme exemple une fuite faiblement évolutive dans une tuyauterie et pour laquelle l'occurrence est lorsque l'ouverture vers l'extérieur atteint 10 millimètres. Suivant cette solution, il serait possible de conclure que ce défaut est diagnosticable, ce qui signifie que si une telle fuite ayant une ouverture de 10 millimètres apparaît, elle sera alors diagnostiquée. Or en fonctionnement réel, cette fuite apparaît de manière faiblement évolutive et rien ne nous garantit que l'invalidité de la propriété normale se fasse lorsque l'ouverture est à 10 millimètres. Cette invalidité de la propriété normale pourrait très bien se réaliser à un instant où l'ouverture se trouve aux alentours de 6 millimètres par exemple. Le diagnostiqueur ne pourrait par la suite pas conclure du fait que ce défaut à une ouverture de 6 millimètres n'a pas été étudié, et ainsi ce défaut de fuite avec une ouverture à 10 millimètres ne serait jamais « atteignable » par le diagnostiqueur car celui-ci ne conclurait jamais à ce tel défaut. Ce problème d'« atteignabilité » n'apparaît néanmoins qu'en pratique car d'un point de vue théorique, l'étude de la diagnosticabilité d'un tel défaut faiblement progressif suivant cette solution est valide : la réponse apportée par l'étude de la diagnosticabilité est rigoureuse.

Pour permettre que l'étude de la diagnosticabilité de tels défauts puisse être considérée en fonctionnement par le diagnostiqueur, il faudrait par conséquent rechercher le niveau du défaut qui fasse chuter la propriété normale. Notons que cette approche risque par ailleurs de ne pas fournir de résultats probants en fonctionnement car suivant la précision considérée pour le niveau de présence du défaut, le passage entre la dernière valeur qui valide la propriété normale et celle qui la fait chuter ne se fait généralement pas à un instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps mais oscille entre ces deux valeurs durant une certaine période temporelle.

4.4 Caractérisation de défauts

L'étude de la diagnosticabilité se menant relativement à une caractérisation Λ de défauts, nous allons formellement définir de quoi il s'agit. Une caractérisation de défauts Λ est une famille de propriétés P_F données pour chaque défaut $F \in \Gamma$ et caractérisant, à chaque instant du temps, le comportement observé du système, qu'il soit sous la présence ou non d'un défaut. Rappelons que le comportement observé du système représente le flux de données constitué par le vecteur des valeurs des variables observables du système à chaque instant du temps. Ces propriétés traduisent les règles d'analyse du diagnostiqueur : c'est-à-dire les règles permettant d'analyser le comportement observé du système sous la présence ou non d'un défaut.

Comme nous l'avons indiqué, pour chacun des défauts $F \in \Gamma$, sa propriété P_F est évaluée à chaque instant du temps par le comportement observé *Obs* du système. Elle est dite *validée* (ou valide) par ce comportement observé *Obs* à un instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps si sa valeur, évaluée par ce comportement observé à cet instant t, est vraie ; ce que nous notons $\mathrm{P}_F(Obs, \mathrm{t}) = \texttt{True}$. Dans le cas contraire, elle est dite *invalidée* (ou invalide) et nous le notons $\mathrm{P}_F(Obs, \mathrm{t}) = \texttt{False}$.

Bien sûr, chacune des propriétés P_F peut nécessiter, pour être évaluée, du fonctionnement passé du système depuis une certaine longueur temporelle λ_{P_F}. C'est-à-dire qu'à chaque instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps, la propriété P_F évalue le comportement observé du système sur la fenêtre temporelle glissante $[\mathrm{t} - \lambda_{\mathrm{P}_F}; \mathrm{t}]$. En considérant $\lambda = max\{\lambda_{\mathrm{P}_F} / F \in \Gamma\}$, nous obtenons ainsi le paramètre correspondant à la longueur d'enregistrement nécessaire au fonctionnement du diagnostiqueur.

Pour chaque défaut $F \in \Gamma$, sa propriété P_F doit exprimer le mieux possible le comportement observé du système sous la présence de ce défaut. C'est ici qu'apparaît ce que nous avons montré

au chapitre 1 d'introduction au diagnostic : la distinction entre le type de données disponibles au diagnostiqueur et son (i.e. : celui du diagnostiqueur) procédé d'analyse de ces données. Le type de données disponibles au diagnostiqueur représente donc les comportements potentiellement observés du système, sous la présence ou non d'un défaut, que nous avons obtenus par les modèles de bon fonctionnement et de défauts, et que nous avons nommés les comportements observables du système. Le procédé d'analyse des données du diagnostiqueur représente pour sa part les règles d'analyse du diagnostiqueur : c'est-à-dire les règles permettant d'analyser le comportement observé du système sous la présence ou non d'un défaut et que nous avons nommées une caractérisation de défauts Λ.

Ainsi, tout l'enjeu d'une bonne caractérisation est qu'elle ait besoin du minimum de connaissance sur le fonctionnement du système, normal ou sous la présence d'un défaut, tout en étant la plus représentative possible : c'est-à-dire qu'elle détecte et isole sans ambiguïté tous les défauts répertoriés. Du point de vue de la diagnosticabilité, où nous travaillons avec les comportements observables du système, cela signifie qu'il faut qu'elle exprime le mieux possible les comportements observables du système sans avoir besoin de stocker au préalable une connaissance importante du fonctionnement du système.

Par conséquent, deux cas peuvent se présenter lorsque la diagnosticabilité du système n'est pas vérifiée. Soit l'ensemble des comportements observables du système (i.e. : l'information disponible) n'est pas assez explicite pour permettre la mise en évidence du défaut par la caractérisation de défauts (i.e. : le procédé d'analyse des données). Soit il s'agit de cette caractérisation de défauts en elle-même qui n'est pas assez « subtile » pour diagnostiquer le défaut.

Enfin, rappelons que nous souhaitons reprendre l'idée générique des méthodologies de diagnostic à base de modèles. Ces méthodologies consistent à comparer le fonctionnement réel du système à un fonctionnement prédit, issu d'un modèle embarqué et simulé par le diagnostiqueur. Concrètement et suivant l'architecture que nous avons considérée, le système complet avec la boucle réelle et la boucle modèle, il est alors possible de comparer non seulement la mesure réelle y à la mesure modèle y^M, mais aussi la commande réelle u à la commande modèle u^M. Cela signifie de considérer les comparaisons $|y - y^M|$ et $|u - u^M|$.

Cependant, nous nous rendons bien compte qu'en ne considérant non plus les évolutions de toutes les variables observables, comme le représentent les comportements observables, mais en ne considérant que les évolutions de ces comparaisons $|y - y^M|$ et $|u - u^M|$, nous avons déjà une perte d'information. En définissant donc une caractérisation de défauts basée sur ces comparaisons, et non pas sur toutes les variables observables du système complet, cela implique donc qu'un défaut pourra ne pas être diagnosticable à cause de cette perte d'information. Il est donc nécessaire de s'assurer au préalable que ce ne sont pas les comportements observables du système qui ne sont pas assez explicites pour permettre la mise en évidence d'un défaut.

Nous allons, dans cette partie, définir deux caractérisations de défauts. La première que nous obtiendrons en considérant simplement les comportements observables sur des fenêtres temporelles de longueur b, le paramètre représentant la borne de diagnostic, et qui va nous permettre de nous assurer de manière intrinsèque de la diagnosticabilité des défauts du système en temps borné par b. Nous la nommerons par conséquent la *caractérisation parfaite*. La seconde, que nous nommerons la *caractérisation par formules temporelles*, va être obtenue en considérant des formules temporelles définies sur les variables observables du système. Notons que ce sera par ailleurs cette caractérisation que nous utiliserons par la suite car elle permettra, à faible coût, de bien représenter la comparaison établie entre le système réel et son modèle embarqué lors du fonctionnement du diagnostiqueur suivant une approche à base de modèles.

4.4.1 La caractérisation parfaite

La solution la plus efficace permettant de vérifier le comportement du système est de le comparer en temps réel à une base de données constituée d'enregistrements de morceaux, de longueur temporelle b, des comportements observables du système. Ce paramètre b correspond à la borne de diagnostic que nous avons introduite en début de chapitre. Pour cette caractérisation, il va aussi s'agir du paramètre λ correspondant à la longueur d'enregistrement : la longueur temporelle d'enregistrement nécessaire au fonctionnement du diagnostiqueur.

L'idée est donc de créer une base de données contenant des morceaux de comportements observables de longueur temporelle b permettant la comparaison, en temps réel, au comportement réellement observé du système. Ainsi en fonctionnement, le diagnostiqueur enregistrera à chaque instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps le comportement réellement observé du système sur la fenêtre temporelle glissante $[\mathrm{t}-b;\mathrm{t}]$, et le comparera aux morceaux de comportements observables enregistrés dans sa base de données. Si ce morceau du comportement réellement observé correspond à un enregistrement normal, alors le système sera jugé comme fonctionnant normalement. Dans le cas contraire, le diagnostiqueur recherchera à quel enregistrement anormal correspond ce morceau de comportement observé.

Il faut donc définir cette base de données qui contient des morceaux de comportements observables permettant la comparaison au comportement réellement observé. Nous allons pour cela considérer les morceaux de longueur temporelle b de chacun des comportements observables Obs de $ObsBeh_{Cons}$. C'est ce que nous allons nommer les *comportements observables bornés* (de longueur temporelle b) et que nous noterons $ObsBeh_{Cons}^{Bd(b)}(F)$ pour n'importe quel défaut $F \in \Gamma$. Le fait que b soit considéré comme un paramètre est justifié par le fait qu'il pourrait être possible de borner par une longueur λ quelconque, inférieure ou supérieure à b.

4.4.1.1 Les comportements observables bornés normaux

Pour le cas normal F_0, nous notons $ObsBeh_{Cons}^{Bd(b)}(F_0)$ l'ensemble des comportements observables bornés normaux (i.e. : sous la présence de F_0). Cet ensemble $ObsBeh_{Cons}^{Bd(b)}(F_0)$ est construit en restreignant, à chaque instant $\mathrm{t} \in \mathrm{I}$ du temps et à partir de l'instant t_0, les comportements observables normaux sur des fenêtres temporelles $[\mathrm{t}-b;\mathrm{t}]$, donc de longueur b.

Il faut bien sûr commencer à les considérer à partir de l'instant $\mathrm{t} = \mathrm{t}_0$. D'une part pour être sûr de se trouver après l'instant $\lambda = b$, car sinon les morceaux ne seront pas complètement définis sur la fenêtre temporelle $[\mathrm{t}-b;\mathrm{t}]$. D'autre part pour être sûr de se trouver à partir de l'instant temporel t_0 pour lequel les instructions sont supposées être exhaustives.

Il s'agit par conséquent de considérer tous les comportements observables normaux $ObsB(cs, F_0, 0)$ quelconques, qui rappelons-le sont définis suivant n'importe quelles instructions $cs \in Cons$, et de les restreindre à chaque instant $\mathrm{t} \in [\mathrm{t}_0; max(\mathrm{I})]$ du temps sur la fenêtre temporelle $[\mathrm{t}-b;\mathrm{t}]$. Cela signifie formellement pour un comportement observable normal $ObsB(cs, F_0, 0)$ et un instant $\mathrm{t} \in [\mathrm{t}_0; max(\mathrm{I})]$ du temps, de considérer sa projection sur le produit d'ensembles $[\mathrm{t}-b;\mathrm{t}] \times \overline{\mathrm{VD}}_{obs}$. C'est-à-dire :

$$\mathrm{Pr}_{[\mathrm{t}-b;\mathrm{t}]\times\overline{\mathrm{VD}}_{obs}}(ObsB(cs, F_0, 0))$$

L'ensemble $ObsBeh_{Cons}^{Bd(b)}(F_0)$ des comportements observables bornés normaux est donc défini par l'union, pour toute instruction $cs \in Cons$ et tout instant $\mathrm{t} \in [\mathrm{t}_0; max(\mathrm{I})]$ du temps, de toutes les projections des comportements observables normaux $ObsB(cs, F_0, 0)$ sur tous les produits d'ensembles $[\mathrm{t}-b;\mathrm{t}] \times \overline{\mathrm{VD}}_{obs}$. C'est-à-dire :

$$ObsBeh_{Cons}^{Bd(b)}(F_0) = \bigcup_{cs\in Cons} \Big(\bigcup_{\mathrm{t}\in[\mathrm{t}_0;max(\mathrm{I})]} \{\mathrm{Pr}_{[\mathrm{t}-b;\mathrm{t}]\times\overline{\mathrm{VD}}_{obs}}(ObsB(cs, F_0, 0))\}\Big)$$

4.4.1.2 Les comportements observables bornés de défauts

Pour un défaut $F \in \Gamma \setminus \{F_0\}$ quelconque, l'ensemble des comportements observables bornés sous la présence de F est noté $ObsBeh^{Bd(b)}_{Cons}(F)$. Il est construit comme pour le cas normal F_0 en restreignant, sur des fenêtres temporelles de longueur b, les comportements observables sous la présence de F à chaque instant du temps, mais uniquement à partir de l'occurrence du défaut.

Il ne faut en effet commencer qu'à partir de l'occurrence $\mathrm{t}_n \in \Omega_{(F,cs)}$ du défaut car par construction d'un comportement de défaut, avant cette occurrence t_n le comportement est identique au comportement normal suivant la même instruction considérée. La restriction sur les fenêtres temporelles ne doit donc se faire qu'à partir des instants de temps supérieurs à l'occurrence du défaut (i.e. : pour les instants $\mathrm{t} \geq \mathrm{t}_n$).

Il n'est par ailleurs pas utile de considérer ces restrictions jusque l'instant $\mathrm{t} = max(\mathrm{I})$. En effet, d'une part le défaut doit être diagnostiqué avant la borne b après son occurrence $\mathrm{t}_n \in \Omega_{(F,cs)}$, ce qui signifie que nous pourrions aller jusqu'à l'instant $\mathrm{t} = (\mathrm{t}_n + b)$: c'est-à-dire de ne considérer les restrictions que pour chaque instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps dans la fenêtre temporelle $[\mathrm{t}_n; \mathrm{t}_n + b]$. D'autre part pour l'étude de la diagnosticabilité, nous souhaitons nous assurer de l'isolation d'un défaut jusqu'au délai de confiance δ. Cela signifie donc qu'il est nécessaire de considérer les restrictions pour chaque instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps dans la fenêtre temporelle $[\mathrm{t}_n; \mathrm{t}_n + (b + \delta)]$.

Pour un comportement observable quelconque $ObsB(cs, F, \mathrm{t}_n)$ sous la présence d'un défaut $F \in \Gamma \setminus \{F_0\}$, qui rappelons-le est donné suivant une instruction $cs \in Cons$ et à une occurrence $\mathrm{t}_n \in \Omega_{(F,cs)}$, et pour un instant $\mathrm{t} \in [\mathrm{t}_n; \mathrm{t}_n + (b+\delta)]$ du temps, cela signifie de considérer la projection de $ObsB(cs, F, \mathrm{t}_n)$ sur le produit d'ensembles $[\mathrm{t} - b; \mathrm{t}] \times \overline{\mathrm{VD}}_{obs}$. C'est-à-dire :

$$\mathrm{Pr}_{[\mathrm{t}-b;\mathrm{t}]\times\overline{\mathrm{VD}}_{obs}}(ObsB(cs, F, \mathrm{t}_n))$$

Remarquons que comme nous avons défini l'ensemble d'occurrences potentielles $\Omega_{(F,cs)}$ de F disjoint avec la fenêtre temporelle $[0; \mathrm{t}_0]$ et que nous avons supposé l'instant t_0 supérieur à b, il est donc tout à fait possible de considérer un comportement observable $ObsB(cs, F, \mathrm{t}_n)$ avec $\mathrm{t}_n = \mathrm{t}_0$ sur la fenêtre temporelle $[0; b]$.

L'ensemble $ObsBeh^{Bd(b)}_{Cons}(F)$ des comportements observables bornés sous la présence du défaut $F \in \Gamma \setminus \{F_0\}$ est donc défini par l'union, pour toutes les instructions $cs \in Cons$, pour toutes les occurrences $\mathrm{t}_n \in \Omega_{(F,cs)}$ de F et tous les instants $\mathrm{t} \in [\mathrm{t}_n; \mathrm{t}_n + (b + \delta)]$ du temps, de toutes les projections de tous les comportements observables $ObsB(cs, F, \mathrm{t}_n)$ sous la présence de F sur tous les produits d'ensembles $[\mathrm{t} - b; \mathrm{t}] \times \overline{\mathrm{VD}}_{obs}$. C'est-à-dire :

$$ObsBeh^{Bd(b)}_{Cons}(F) = \bigcup_{cs\in Cons} \Big(\bigcup_{\mathrm{t}_n\in\Omega_{(F,cs)}} \Big(\bigcup_{\mathrm{t}\in[\mathrm{t}_n;\mathrm{t}_n+(b+\delta)]} \{\mathrm{Pr}_{[\mathrm{t}-b;\mathrm{t}]\times\overline{\mathrm{VD}}_{obs}}(ObsB(cs, F, \mathrm{t}_n))\}\Big)\Big)$$

4.4.1.3 Les propriétés des défauts de la caractérisation parfaite

Pour chaque défaut $F \in \Gamma$, sa propriété P_F est définie de la manière suivante : pour un instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps, $\mathrm{P}_F(Obs, \mathrm{t}) = \texttt{True}$ si et seulement si d'une part la fenêtre temporelle $[\mathrm{t} - b; \mathrm{t}]$ est bien définie et d'autre part l'ensemble des vecteurs de Obs considérés sur la fenêtre temporelle $[\mathrm{t} - b; \mathrm{t}]$ est un élément de l'ensemble $ObsBeh^{Bd(b)}_{Cons}(F)$ des comportements observables bornés sous la présence du défaut F.

Comme pour l'étude de la diagnosticabilité, nous considérons l'ensemble $ObsBeh_{Cons}$ des comportements observables du système obtenus grâce aux modèles de bon fonctionnement et de défauts. Ces

propriétés peuvent donc être évaluées par ces comportements observables. Pour chaque défaut $F \in \Gamma$, sa propriété P_F est définie de la manière suivante : pour un comportement observable $Obs \in ObsBeh_{Cons}$ et un instant $\mathrm{t} \in \mathrm{I}$ du temps, $\mathrm{P}_F(Obs, \mathrm{t}) = \texttt{True}$ si et seulement si d'une part la fenêtre temporelle $[\mathrm{t} - b; \mathrm{t}]$ est incluse dans le domaine temporel $Tdom(Obs)$ de Obs et d'autre part la projection de Obs sur le produit d'ensembles $[\mathrm{t} - b; \mathrm{t}] \times \overline{\mathrm{VD}}_{Obs}$ est un élément de l'ensemble $ObsBeh_{Cons}^{Bd(b)}(F)$ des comportements observables bornés sous la présence de F. C'est-à-dire formellement :

Définition 4.8 *Pour un défaut $F \in \Gamma$, un comportement observable $Obs \in ObsBeh_{Cons}$ et un instant $\mathrm{t} \in \mathrm{I}$ du temps, $\mathrm{P}_F(Obs, \mathrm{t}) = \texttt{True}$ si et seulement si :*

- $[\mathrm{t} - b; \mathrm{t}] \subseteq Tdom(Obs)$
- $\mathrm{Pr}_{[\mathrm{t}-b;\mathrm{t}]\times\overline{\mathrm{VD}}_{obs}}(Obs) \in ObsBeh_{Cons}^{Bd(b)}(F)$

Le fait d'imposer que $[\mathrm{t} - b; \mathrm{t}] \subseteq Tdom(Obs)$ découle du fait qu'à l'instant $\mathrm{t} \in \mathrm{I}$, le comportement observable $Obs \in ObsBeh_{Cons}$ pourrait ne pas être totalement, voire pas du tout, défini sur cette fenêtre temporelle $[\mathrm{t} - b; \mathrm{t}]$. En considérant par exemple $\mathrm{t} \in [0; b]$.

Pour un comportement observable $Obs \in ObsBeh_{Cons}$ et un instant $\mathrm{t} \in \mathrm{I}$ du temps tel que $[\mathrm{t} - b; \mathrm{t}] \subseteq Tdom(Obs)$, sa restriction sur la fenêtre temporelle $[\mathrm{t} - b; \mathrm{t}]$ est un élément de l'ensemble $ObsBeh_{Cons}^{Bd(b)}(F)$ des comportements observables bornés sous la présence de F si et seulement s'il existe un élément $Obs_F^{bd(b)}$ de l'ensemble $ObsBeh_{Cons}^{Bd(b)}(F)$ tels que les vecteurs de données de Obs, considérés uniquement sur la fenêtre temporelle $[\mathrm{t} - b; \mathrm{t}]$, sont à chaque instant égaux aux vecteurs de données de $Obs_F^{bd(b)}$. Formellement si pour tout instant de temps $k \in [0; b] \subseteq \mathbb{T}_\iota$, le vecteur de données de Obs à l'instant $(\mathrm{t} - b) + k$ est égal au vecteur de données de $Obs_F^{bd(b)}$ à l'instant k : c'est-à-dire pour tout instant de temps $k \in [0; b] \subseteq \mathbb{T}_\iota$, $v_{Obs}((\mathrm{t} - b) + k) = v_{Obs_F^{bd(b)}}(k)$.

4.4.1.4 Étude de la diagnosticabilité du système par utilisation de la caractérisation parfaite

En ayant défini les comportements observables bornés de défauts permettant de construire la caractérisation parfaite, il nous est possible d'étudier la diagnosticabilité du système.

Diagnosticabilité du cas normal La première étape consiste à vérifier la diagnosticabilité du cas normal $F_0 \in \Gamma$. Il faut pour cela vérifier que pour chacun des comportements observables normaux $ObsB(cs, F_0, 0) \in ObsBeh_{Cons}(F_0)$ suivant une instruction $cs \in Cons$, sa restriction sur la fenêtre temporelle glissante $[\mathrm{t} - b; \mathrm{t}]$, considérée à chaque instant $\mathrm{t} \in [\mathrm{t}_0; max(\mathrm{I})]$, appartienne à l'ensemble $ObsBeh_{Cons}^{Bd(b)}(F_0)$ des comportements observables bornés normaux. Suivant la construction de cet ensemble $ObsBeh_{Cons}^{Bd(b)}(F_0)$, cela est toujours vrai. Ainsi le cas normal $F_0 \in \Gamma$ est toujours diagnosticable. Comme pour ce cas normal les définitions de diagnosticabilité et d'éligibilité sont équivalentes, F_0 est donc aussi toujours éligible.

Les seules fois où ce cas normal F_0 risque de ne pas être diagnostiqué en fonctionnement, c'est-à-dire que le diagnostiqueur conclurait à un fonctionnement anormal alors que le système fonctionne normalement, seraient lorsque la suite des consignes, apparues lors de ce fonctionnement réel, n'a pas été utilisée dans l'ensemble $Cons$ des instructions ayant servi à l'étude de la diagnosticabilité. Ceci nécessite donc d'avoir un ensemble $Cons$ d'instructions le plus représentatif possible.

Diagnosticabilité des défauts Les étapes qui suivent l'étude de la diagnosticabilité du cas normal F_0 consistent à étudier la diagnosticabilité des défauts $F \in \Gamma \setminus \{F_0\}$. Pour chacun de ces défauts $F \in \Gamma \setminus \{F_0\}$, il faut considérer tous les comportements observables $ObsB(cs, F, \mathrm{t}_n) \in ObsBeh_{Cons}(F)$ suivant n'importe quelle instruction $cs \in Cons$ et sous la présence de F à n'importe quelle occurrence

$t_n \in \Omega_{(F,cs)}$. Pour chacun de ces comportements observables $ObsB(cs, F, t_n)$, il faut vérifier qu'il existe un instant de détection $t_k \in I$ tel que :

1. $t_k \in [t_n{}^-; t_n + (b-h)]$;
2. pour tout instant $t \in I$, si $t \in [t_n; t_k[$ alors :
   (a) $[t-b; t] \subseteq Tdom(ObsB(cs, F, t_n))$
   (b) et $\Pr_{[t-b;t]\times\overline{VD}_{obs}}(ObsB(cs, F, t_n)) \in ObsBeh^{Bd(b)}_{Cons}(F_0)$.
3. pour tout instant $t \in I$, si $t = t_k$ alors :
   (a) $[t-b; t] \subseteq Tdom(ObsB(cs, F, t_n))$
   (b) et $\Pr_{[t-b;t]\times\overline{VD}_{obs}}(ObsB(cs, F, t_n)) \notin ObsBeh^{Bd(b)}_{Cons}(F_0)$.
4. pour tout instant $t \in I$, si $t \in [t_k + h; t_k + (h+\delta)]$ alors :
   (a) $[t-b; t] \subseteq Tdom(ObsB(cs, F, t_n))$
   (b) et $\Pr_{[t-b;t]\times\overline{VD}_{obs}}(ObsB(cs, F, t_n)) \in ObsBeh^{Bd(b)}_{Cons}(F)$.
5. pour tout instant $t \in I$, si $t \in [t_k + h; t_k + (h+\delta)]$ alors :
   (a) $[t-b; t] \subseteq Tdom(ObsB(cs, F, t_n))$
   (b) et $\Pr_{[t-b;t]\times\overline{VD}_{obs}}(ObsB(cs, F, t_n)) \notin ObsBeh^{Bd(b)}_{Cons}(F')$ pour tout autre défaut $F' \in \Gamma \setminus \{F\}$.

Les points (a) sont à chaque fois clairement satisfaits car nous avons imposé que l'occurrence t_n de F soit après l'instant de temps $t = t_0$ que nous avons supposé supérieur à b. Donc pour tout instant $t \geq t_n$, nous avons toujours $[t-b; t] \subseteq Tdom(ObsB(cs, F, t_n))$. Le point (4.b) est par ailleurs lui aussi clairement satisfait par construction de l'ensemble $ObsBeh^{Bd(b)}_{Cons}(F)$ des comportements observables bornés sous la présence de F ; ceci prouve d'ailleurs que tout défaut F est éligible avec cette caractérisation parfaite, avec un instant d'éligibilité $t_{ke} = t_n$.

Si le défaut F n'est pas diagnosticable, il est par conséquent nécessaire de revenir aux définitions de détectabilité et d'isolabilité de F. Nous n'avons pas besoin d'en analyser l'éligibilité car, comme nous venons de le voir, l'instant d'éligibilité est toujours $t_{ke} = t_n$ et vérifiera donc toujours $t_{ke} \leq t_{kd}$ pour n'importe quel instant de détectabilité $t_{ke} \in [t_n; t_n + (b-h)]$. Donc si F n'est pas diagnosticable, cela signifie :

- soit que les instants t_{kd} de détectabilité et t_{ki} d'isolablité ne sont pas compatibles avec la diagnosticabilité (i.e. : $t_{ki} \leq t_{kd}$) ;
- soit que le délai h d'isolation ou que la borne b de diagnostic sont trop courts ou longs ;
- soit qu'il n'est pas détectable, ce qui signifie que les comportements observables bornés du système en fonctionnement normal ne permettent pas de détecter le défaut ;
- soit qu'il n'est pas isolable, ce qui signifie que les comportements observables bornés du système en fonctionnement sous la présence des autres défauts ne permettent pas d'isoler le défaut.

4.4.1.5 Perfection de la caractérisation parfaite

Cette caractérisation est dite « parfaite ». En effet, les comportements observables bornés $Obs^{Bd(b)}_F$ sous la présence d'un défaut quelconque $F \in \Gamma$ sont construits en restreignant les comportements observables $ObsB(cs, F, t_n)$ uniquement sur des fenêtres temporelles. Or ces comportements observables sont les définitions nominales des comportements observables du système sous la présence d'un défaut. Il n'est par conséquent pas possible de les spécifier et les construire d'une meilleure façon. Ils contiennent donc toute l'information observable disponible sur le comportement du système en fonctionnement, normal ou sous la présence d'un défaut. Les deux seuls moyens de rajouter de l'information observable seraient :

- de considérer une période d'échantillonnage ι plus fine du temps $\mathbb{T}_\iota$, ce qui reviendrait à changer la dynamique temporelle des modèles ;
- de rajouter des informations venant du système, ce qui reviendrait à rajouter des capteurs permettant d'observer d'autres grandeurs physiques du système.

Cela signifierait donc de modifier les modèles du système, et par voie de conséquence de modifier les comportements observables du système.

Par ailleurs et pour un diagnostic en temps borné par b, à l'instant $\mathrm{t}_n + b$ après l'occurrence d'un défaut, l'enregistrement, par le diagnostiqueur, du comportement réellement observé du système sur la fenêtre temporelle $[\mathrm{t}_n; \mathrm{t}_n + b]$ représente toute l'information observable nécessaire au diagnostic. Il n'est pas possible d'avoir plus d'information durant cette fenêtre temporelle de longueur b. Ainsi les défauts conclus non-diagnosticables (de même pour les défauts non-détectables ou non-isolables), suivant cette caractérisation parfaite, ne pourraient l'être avec une autre caractérisation. Le cas contraire signifierait alors une aberration dans la construction de cette autre caractérisation, car toute autre caractérisation, basée sur cet ensemble de comportements observables du système, ne peut que « perdre » de l'information observable sur ces comportements observables. Elle ne peut donc qu'être moins précise que cette caractérisation parfaite.

Néanmoins, cette « perfection » n'est pas sans conséquence. D'une part, l'ensemble $Cons$ des instructions doit être le plus représentatif possible. Cela signifie potentiellement l'ensemble $C^{[\mathrm{t}_0;b]}$ de toutes les fonctions de l'intervalle temporel $[\mathrm{t}_0; b]$ dans l'ensemble C du domaine des valeurs de la consigne c. Cet intervalle temporel $[\mathrm{t}_0; b]$ est donné afin d'avoir toutes les informations nécessaires durant les fenêtres temporelles de longueur b. D'autre part, les ensembles d'occurrences $\Omega_{(F,cs)}$ des défauts $F \in \Gamma \setminus \{F_0\}$, pour n'importe quelles instructions $cs \in Cons$ doivent, eux aussi, être les plus représentatifs. Cela signifie potentiellement, en considérant l'ensemble d'instructions $Cons = C^{[\mathrm{t}_0;b]}$, l'instant d'occurrence $\mathrm{t}_n = \mathrm{t}_0$ pour chaque défaut $F \in \Gamma \setminus \{F_0\}$ et chaque instruction $cs \in Cons$: c'est-à-dire $\Omega_{(F,cs)} = \{\mathrm{t}_0\}$. Nous pouvons déjà remarquer, sans techniques spécifiques d'implémentation, que cette caractérisation parfaite n'est potentiellement pas applicable pour des systèmes complexes, notamment ceux utilisés en embarqué ayant des ressources limitées en capacité de calcul et espace de stockage. En effet, ils ont non seulement un domaine de fonctionnement important, mais aussi beaucoup de défauts potentiels à prendre en compte par le diagnostiqueur. Il y aura donc d'une part un nombre très important de comportements observables bornés à stocker afin de pouvoir mener les comparaisons. D'autre part et à chaque instant du temps, le nombre de comparaisons, du comportement observé du système réel aux comportements observables bornés stockés, risque lui aussi d'être très important : potentiellement et dans le pire des cas, une comparaison à tous les comportements observables bornés sous la présence de n'importe quels défauts $F \in \Gamma$.

Remarquons de plus que nous utilisons une architecture spéciale du système : le système complet constitué d'une boucle réelle et d'une boucle modèle. Or cette architecture complète n'est pas nécessaire, et donc pas optimale, pour générer un diagnostiqueur issu de l'étude de la diagnosticabilité suivant cette caractérisation parfaite. En effet, il serait possible de considérer une architecture constituée uniquement de la boucle réelle, et d'étudier alors la diagnosticabilité uniquement sur le modèle de cette boucle. Nous justifions néanmoins l'utilisation de cette caractérisation parfaite avec la boucle complète car nous souhaitons utiliser une approche de diagnostic à base de modèles, qui utilise donc un modèle embarqué du système afin de comparer son fonctionnement réel à celui du modèle, prenant en compte l'évolution temporelle du fonctionnement du système. Nous souhaitons donc nous assurer de la diagnosticabilité intrinsèque des défauts avec cette caractérisation parfaite, pour ensuite pouvoir utiliser une autre caractérisation prenant uniquement en compte les évolutions temporelles des comparaisons $|c - y|$, $|y - y^M|$ et $|u - u^M|$, construite donc suivant un autre formalisme que celui ensembliste que nous venons d'utiliser. C'est la caractérisation par formules temporelles que nous allons présenter juste après.

Notons par ailleurs qu'il est aussi possible de définir une caractérisation plus faible que la caractérisation parfaite et toujours construite suivant ce formalisme ensembliste (i.e. : des ensembles de données bornés). Au lieu de considérer les évolutions de toutes les variables observables du système, comme la caractérisation parfaite, nous pouvons alors ne considérer que les évolutions de ces comparaisons

$|c - y|$, $|y - y^M|$ et $|u - u^M|$. Néanmoins, cette caractérisation plus faible risque, là encore, d'être inexploitable car elle nécessiterait toujours un stockage et une puissance de calcul très importants.

Ainsi et bien qu'elle risque sûrement d'être inexploitable pour une utilisation en solution de diagnostic d'un système embarqué, cette caractérisation parfaite est très importante lors des phases de conception du système afin de s'assurer, et ce de manière intrinsèque, de la diagnosticabilité bornée par b des défauts potentiels répertoriés.

4.4.2 La caractérisation par formules temporelles

Pour décrire le comportement observable du système sous la présence ou non d'un défaut, nous venons d'utiliser un formalisme de logique ensembliste. Nous venons de voir que ce formalisme est bien adapté pour s'assurer de la diagnosticabilité des défauts, mais qu'il risque néanmoins de souffrir d'une complexité importante pouvant sûrement le rendre inutilisable pour des systèmes complexes. Il convient donc d'utiliser un formalisme plus adapté.

Par ailleurs, rappelons que nous souhaitons ajouter un aspect temporel dans l'utilisation des méthodologies de diagnostic à base de modèles. Nous souhaitons en effet pouvoir prendre en compte l'évolution temporelle dans la comparaison du fonctionnement du système réel à celui du modèle.

Nous allons de ce fait considérer un formalisme de logique temporelle. Nous allons pour cela utiliser une adaptation ([Rap08]) de la logique temporelle à base de mesures d'intervalles : MITL pour *Metric Interval Temporal Logic* ([AFH96]). Cette logique temporelle est spécifiquement adaptée pour exprimer des propriétés temps réel bornées sur un ensemble de variables évoluant dans un domaine temporel continu. Dans notre cas, le pouvoir d'expression de cette logique ne sera pas totalement exploité car nous ne considérons que des variables évoluant dans un domaine temporel discret (i.e. : cadencées à un pas de temps fixe). Nous pourrons alors décrire les évolutions temporelles des comparaisons $|c-y|$, $|y - y^M|$ et $|u - u^M|$ lors de différents fonctionnements du système, normal ou sous la présence de défauts.

La vérification des formules temporelles, qui caractériseront le comportement observable du système sous la présence ou non d'un défaut, se réalisera grâce à l'utilisation de l'outil ARTiMon© du CEA/List ([Rap08]). Cet outil, directement relié à l'environnement de simulation MATLAB/Simulink©, va nous permettre non seulement de définir certaines formules temporelles mais aussi d'une part d'en étudier la diagnosticabilité et d'autre part d'être le diagnostiqueur embarqué, ce que nous verrons au chapitre 5 suivant concernant la génération du diagnostiqueur.

4.4.2.1 Syntaxe

Les formules temporelles considérées sont construites de manière usuelle par induction. Nous construisons d'abord l'ensemble des termes, représentant des formules arithmétiques, qui nous permettra ensuite de construire l'ensemble des formules atomiques représentant des comparaisons de formules arithmétiques. Enfin l'ensemble des formules temporelles se construira par induction à partir de ces formules atomiques.

L'ensemble des termes Les termes sont des formules arithmétiques construites grâce aux opérateurs arithmétiques classiques (l'addition, la soustraction, la multiplication, la division et la valeur absolue) sur un ensemble de constantes et les variables observables du système.

Nous considérons un ensemble K de constantes (l'ensemble $\mathbb{Q}$ des nombres rationnels par exemple) et l'ensemble $\overline{\mathrm{V}}_{obs} = \{c; u; u^M; y; y^M\}$ des variables observables du système. Nous considérons de plus les opérateurs arithmétiques usuels ($+$, $-$, $\times$, $\div$ et $|\ |$) ainsi qu'un opérateur temporel $\mathrm{V}_{[\alpha]}$, où α est une constante positive ou négative du temps.

L'ensemble des termes est construit inductivement par les règles suivantes :
- Toute constante $k \in \mathrm{K}$ est un terme.
- Toute variable $v \in \overline{\mathrm{V}}_{obs}$ est un terme.
- Pour deux termes te_1 et te_2 et pour un opérateur arithmétique $* \in \{+; -; \times; \div; |\ |\}$, $te_1 * te_2$ est un terme.
- Pour un terme te et une constante de temps α positive ou négative, $\mathrm{V}_{[\alpha]}te$ est un terme.

Exemple L'expression suivante est un terme :

$$|c - \mathrm{V}_{[-0.01]}c|$$

À un instant $\mathrm{t} \in \mathbb{T}_{0.01}$, elle exprime la différence absolue entre la variable c et cette même variable à l'instant précédent du temps. Ce type de terme nous sera utile lorsque nous souhaiterons vérifier si une variable a évolué dans le temps.

L'ensemble des formules atomiques Les formules atomiques sont des comparaisons (i.e. : égalités ou inégalités) de formules arithmétiques (i.e. : des termes). En considérant les opérateurs usuels de comparaison ($=$, $<$, $\leq$, $>$ et $\geq$), l'ensemble des formules atomiques est alors construit inductivement par la règle suivante :
- Pour deux termes te_1 et te_2 et pour un opérateur de comparaison $\bowtie \in \{=; <; \leq; >; \geq\}$, alors $te_1 \bowtie te_2$ est une formule atomique.

Exemple L'expression suivante est une formule atomique :

$$|c - \mathrm{V}_{[-0.01]}c| = 0$$

À un instant $\mathrm{t} \in \mathbb{T}_{0.01}$, elle exprime le fait que la variable c n'a pas évolué par rapport à l'instant précédent.

L'ensemble des formules temporelles L'ensemble des formules temporelles est construit inductivement par les règles suivantes :
Opérateurs de la logique classique :
- Négation : pour une formule temporelle φ, alors $\neg\varphi$ est une formule temporelle.
- Disjonction : pour deux formules temporelles φ et ψ, alors $\varphi \vee \psi$ est une formule temporelle.
- Conjonction : pour deux formules temporelles φ et ψ, alors $\varphi \wedge \psi$ est une formule temporelle.
- Implication : pour deux formules temporelles φ et ψ, alors $\varphi \Rightarrow \psi$ est une formule temporelle.
- Équivalence : pour deux formules temporelles φ et ψ, alors $\varphi \Leftrightarrow \psi$ est une formule temporelle.

Opérateurs temporels :
- Globalité bornée sur une fenêtre temporelle : pour une formule temporelle φ et deux constantes de temps α et β positives ou négatives telles que $\alpha \leq \beta$, alors $\mathrm{G}_{[\alpha;\beta]}\varphi$ est une formule temporelle.
- Existencialité bornée sur une fenêtre temporelle : pour une formule temporelle φ et deux constantes de temps α et β positives ou négatives telles que $\alpha \leq \beta$, alors $\mathrm{E}_{[\alpha;\beta]}\varphi$ est une formule temporelle.
- « Jusqu'à » borné sur une fenêtre temporelle : pour deux formules temporelles φ et ψ et deux constantes de temps α et β positives ou négatives telles que $\alpha \leq \beta$, alors $\varphi \mathrm{U}_{[\alpha;\beta]}\psi$ est une formule temporelle.

Exemple L'expression suivante est une formule temporelle :

$$\varphi_{ex} \ : \ [(0 \leq c) \wedge (c \leq 10) \wedge \mathrm{G}_{[-3;0]}(|c - \mathrm{V}_{[-0.01]}c| = 0)] \Rightarrow [(0.1 \leq |y - y^M|) \wedge (|y - y^M| \leq 0.2)]$$

À un instant $\mathrm{t} \in \mathbb{T}_{0.01}$, elle exprime le fait que si la variable c est entre la valeur 0 et la valeur 10 et qu'elle n'a pas évolué depuis 3 unités de temps, alors la différence absolue $\varepsilon_{(y,y^M)} = |y - y^M|$ entre la variable y et la variable y^M est comprise entre les valeurs 0.1 et 0.2.

4.4.2.2 Sémantique

Les formules temporelles sont évaluées à chaque instant du temps par le comportement observé du système : c'est-à-dire le flux de données constitué des vecteurs des valeurs des variables observables du système à chaque instant du temps. La relation de satisfaction, notée $\models$, est donc définie pour n'importe quel comportement observé et n'importe quel instant du temps par induction sur l'ensemble des formules temporelles (i.e. : pour chaque opérateur). Pour une formule temporelle φ et un instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps, nous notons $(Obs, \mathrm{t}) \models \varphi$ la satisfaction de φ par le comportement observé Obs du système à l'instant t. Le couple (Obs, t) désigne le vecteur v_{Obs} des variables observables du système à l'instant t (i.e. : $v_{Obs}(\mathrm{t}) = (c(\mathrm{t}), u(\mathrm{t}), u^M(\mathrm{t}), y(\mathrm{t}), y^M(\mathrm{t}))$).

Comme les opérateurs temporels considèrent des décalages temporels dans le temps, la relation de satisfaction doit donc prendre en compte ces décalages. Cela signifie qu'à un instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps il y a satisfaction d'une formule φ par le comportement observé Obs du système uniquement si ce comportement observé est bien défini dans la fenêtre temporelle considérée par l'opérateur principal de la formule temporelle évaluée. Si par exemple l'opérateur principal fait référence à des instants passés qui n'étaient pas définis car le système vient d'être mis en route, la formule temporelle ne peut donc pas être évaluée. Le domaine temporel d'une formule φ, que nous notons $TDom(\varphi)$ et qui est défini par induction, est donc l'intervalle temporelle dont dépend la validité de la formule φ.

Sémantique des formules atomiques La satisfaction d'une formule atomique est définie suivant que cette formule contienne ou non un terme de la forme $\mathrm{V}_{[\alpha]}te$: c'est-à-dire l'opérateur temporel $\mathrm{V}_{[\alpha]}$ défini sur les termes. Pour ces formules atomiques, nous allons considérer le domaine temporel $TDom(Obs)$ du comportement observé Obs ; il va s'agir de la fenêtre temporelle débutant à l'instant initial t_{init} quand le système a été mis en marche et n'ayant potentiellement pas de fin.

- En considérant une formule atomique at ne contenant pas de terme de la forme $\mathrm{V}_{[\alpha]}te$ et pour un instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps : $(Obs, \mathrm{t}) \models at$ si et seulement si $\mathrm{t} \in TDom(at) = TDom(Obs)$ et la valeur de at est vraie lorsque toutes les variables de at sont substituées par leurs valeurs provenant du vecteur v_{Obs} à cet instant t du temps.
- En considérant une formule atomique at contenant un terme de la forme $\mathrm{V}_{[\alpha]}te$ et pour un instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps : $(Obs, \mathrm{t}) \models at$ si et seulement si $(\mathrm{t}+\alpha) \in TDom(at) = TDom(Obs)$ et la valeur de at est vraie lorsque d'une part toutes les variables du terme te apparaissant dans $\mathrm{V}_{[\alpha]}te$ sont substituées par leurs valeurs provenant du vecteur v_{Obs} à l'instant $(\mathrm{t} + \alpha)$ du temps, d'autre part toutes les variables de at n'apparaissant pas dans le terme $\mathrm{V}_{[\alpha]}te$ sont substituées par leurs valeurs provenant du vecteur v_{Obs} à l'instant t du temps.

Sémantique des opérateurs de la logique classique Pour les opérateurs de la logique classique, la relation de satisfaction est définie de manière classique suivant les différents opérateurs. Les domaines de validité des formules ne seront pas impactés par ces opérateurs classiques.

- Négation : pour une formule temporelle φ et pour un instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps, $(Obs, \mathrm{t}) \models \neg\varphi$ si et seulement si $\mathrm{t} \in TDom(\varphi)$ et $(Obs, \mathrm{t}) \not\models \varphi$.
- Disjonction : pour deux formules temporelles φ et ψ et pour un instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps, $(Obs, \mathrm{t}) \models \varphi \vee \psi$ si et seulement si $\mathrm{t} \in TDom(\varphi) \cap TDom(\psi)$ et $(Obs, \mathrm{t}) \models \varphi$ ou $(Obs, \mathrm{t}) \models \psi$.
- Conjonction : pour deux formules temporelles φ et ψ et pour un instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps, $(Obs, \mathrm{t}) \models \varphi \wedge \psi$ si et seulement si $\mathrm{t} \in TDom(\varphi) \cap TDom(\psi)$ et $(Obs, \mathrm{t}) \models \varphi$ et $(Obs, \mathrm{t}) \models \psi$.

- Implication : pour deux formules temporelles φ et ψ et pour un instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps, $(Obs, \mathrm{t}) \models \varphi \Rightarrow \psi$ si et seulement si $\mathrm{t} \in TDom(\varphi) \cap TDom(\psi)$ et si $(Obs, \mathrm{t}) \models \varphi$ alors $(Obs, \mathrm{t}) \models \psi$.
- Équivalence : pour deux formules temporelles φ et ψ et pour un instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps, $(Obs, \mathrm{t}) \models \varphi \Leftrightarrow \psi$ si et seulement si $\mathrm{t} \in TDom(\varphi) \cap TDom(\psi)$ et $(Obs, \mathrm{t}) \models \varphi$ si et seulement si $(Obs, \mathrm{t}) \models \psi$.

Sémantique des opérateurs temporels Pour les opérateurs temporels, la relation de satisfaction est définie en considérant l'intervalle temporel de l'opérateur et la satisfaction de la formule sous-jacente à différents instants de temps suivant l'opérateur considéré.

- Globalité bornée sur une fenêtre temporelle : pour une formule temporelle φ, pour deux constantes de temps α et β positives ou négatives et pour un instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps, $(Obs, \mathrm{t}) \models \mathrm{G}_{[\alpha;\beta]}\varphi$ si et seulement si $[\mathrm{t}+\alpha; \mathrm{t}+\beta] \subseteq TDom(\varphi)$ et pour tout instant $\mathrm{t}' \in [\mathrm{t}+\alpha; \mathrm{t}+\beta]$: $(Obs, \mathrm{t}') \models \varphi$.
- Existencialité bornée sur une fenêtre temporelle : pour une formule temporelle φ, pour deux constantes de temps α et β positives ou négatives et pour un instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps, $(Obs, \mathrm{t}) \models \mathrm{E}_{[\alpha;\beta]}\varphi$ si et seulement si $[\mathrm{t}+\alpha; \mathrm{t}+\beta] \subseteq TDom(\varphi)$ et il existe un instant $\mathrm{t}' \in [\mathrm{t}+\alpha; \mathrm{t}+\beta]$ tel que $(Obs, \mathrm{t}') \models \varphi$.
- « Jusqu'à » borné sur une fenêtre temporelle : pour deux formules temporelles φ et ψ, pour deux constantes de temps α et β positives ou négatives et pour un instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps, $(Obs, \mathrm{t}) \models \varphi \mathrm{U}_{[\alpha;\beta]}\psi$ si et seulement si $[\mathrm{t}+\alpha; \mathrm{t}+\beta] \subseteq TDom(\varphi)$ et il existe un instant $\mathrm{t}' \in [\mathrm{t}+\alpha; \mathrm{t}+\beta]$ tel que pour tout instant $\mathrm{t}'' \in [\mathrm{t}+\alpha; \mathrm{t}'[$: $(Obs, \mathrm{t}'') \models \varphi$, et pour tout instant $\mathrm{t}'' \in [\mathrm{t}'; \mathrm{t}+\beta]$: $(Obs, \mathrm{t}'') \models \psi$.

Il nous semble important d'expliquer la sémantique de ces opérateurs temporels, d'autant plus pour l'opérateur « Jusqu'à » borné sur une fenêtre temporelle. L'opérateur de globalité bornée sur une fenêtre temporelle signifie que la sous-formule considérée doit être valide à chaque instant du temps dans la fenêtre temporelle exprimée par l'opérateur. L'opérateur d'existencialité bornée sur une fenêtre temporelle signifie que la sous-formule considérée doit être valide au moins à un instant du temps dans la fenêtre temporelle exprimée par l'opérateur. L'opérateur « Jusqu'à » borné sur une fenêtre temporelle, considéré pour deux sous-formules φ et ψ, signifie que durant la fenêtre temporelle exprimée par l'opérateur : la sous-formule φ doit toujours être vraie jusqu'à ce que la sous-formule ψ soit vraie. Notons que ce « jusqu'à » peut être à l'instant initial de la fenêtre temporelle exprimée par l'opérateur.

Exemple Comme nous l'avons vu, la formule suivante :

$$\varphi_{ex} \ : \ [(0 \leq c) \wedge (c \leq 10) \wedge \mathrm{G}_{[-3;0]}(|c - \mathrm{V}_{[-0.01]}c| = 0)] \Rightarrow [(0.1 \leq |y - y^M|) \wedge (|y - y^M| \leq 0.2)]$$

exprime que si la variable c est dans l'intervalle $[0; 10]$ et qu'elle n'a pas évolué depuis 3 unités de temps ; alors la différence absolue entre la mesure réelle y et la mesure modèle y^M est dans l'intervalle $[0.1; 0.2]$.

4.4.2.3 Les propriétés des défauts de la caractérisation par formules temporelles

Pour chacun des défauts $F \in \Gamma$, nous allons définir une formule temporelle φ_F caractérisant, si possible, le comportement observable du système sous la présence du défaut F considéré. Pour le cas normal $F_0 \in \Gamma$, nous allons donner une procédure permettant de l'obtenir alors que pour les autres défauts $F \in \Gamma \setminus \{F_0\}$ nous serons obligé de les définir au cas pas cas sans méthode prédéfinie.

Rappelons que nous souhaitons utiliser une approche de diagnostic à base de modèles prenant en compte l'évolution temporelle du fonctionnement du système. Pour le cas normal, nous allons donc

décrire les évolutions temporelles des comparaisons $|c-y|$, $|y-y^M|$ et $|u-u^M|$ en fonction de l'évolution de la consigne.

Les propriétés caractéristiques des défauts Lorsque chacune des formules φ_F est construite pour chacun des défauts $F \in \Gamma$, les propriétés P_F sont définies de la manière suivante : pour un instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps, $\mathrm{P}_F(Obs, \mathrm{t}) = \texttt{True}$ si et seulement si Obs satisfait la formule temporelle φ_F à cet instant t (i.e. : si $(Obs, \mathrm{t}) \models \varphi_F$).

Comme pour l'étude de la diagnosticabilité, nous considérons l'ensemble $ObsBeh_{Cons}$ des comportements observables du système obtenus grâce aux modèles de bon fonctionnement et de défauts. Ces propriétés peuvent donc être évaluées par ces comportements observables. C'est-à-dire formellement :

Définition 4.9 *Pour un défaut $F \in \Gamma$, un comportement observable $Obs \in ObsBeh_{Cons}$ et un instant $\mathrm{t} \in \mathrm{I}$ du temps, $\mathrm{P}_F(Obs, \mathrm{t}) = \texttt{True}$ si et seulement si $(Obs, \mathrm{t}) \models \varphi_F$.*

Identiquement au cas de la caractérisation parfaite, remarquons que la validité de la propriété P_F, évaluée par un comportement observable $Obs \in ObsBeh_{Cons}$ à un instant $\mathrm{t} \in \mathrm{I}$, dépend aussi du fait que l'intervalle temporel, dépendant de l'opérateur temporel principal de φ_F, est bien inclus dans le domaine temporel du comportement observable Obs (i.e. : dans l'intervalle temporel I).

La formule du cas normal Pour la formule φ_{F_0} du cas normal $F_0 \in \Gamma$, nous allons comparer le fonctionnement réel du système et le fonctionnement du modèle embarqué : c'est-à-dire, pour l'étude de la diagnosticabilité, que nous allons comparer le fonctionnement du modèle parfait M_{parf} au fonctionnement du modèle embarqué M_{emb}. Nous allons de ce fait exprimer les évolutions des comparaisons $|c-y|$, $|y-y^M|$ et $|u-u^M|$, ceci suivant la dynamique d'évolution de la consigne c. Nous partons du constat qu'en fonctionnement, la consigne c de l'opérateur évolue dans le domaine C et que ces évolutions peuvent être plus ou moins fortes. Le système, ayant une certaine dynamique de réaction, ne va donc pas « répondre » immédiatement suivant cette consigne lorsqu'elle évolue fortement. Lors de faibles changements de la consigne, nous dirons que le système est en fonctionnement statique, alors que lors de forts changements de la consigne, nous dirons que le système est en fonctionnement dynamique.

L'utilisation d'un modèle embarqué M_{emb}, permettant de comparer le fonctionnement du système réel au fonctionnement de ce modèle, prend ici toute son importance. Le but principal étant en effet que le système réel fonctionne normalement suivant la consigne de l'opérateur, nous pouvons donc estimer qu'il fonctionne normalement lorsque la comparaison $|c-y|$ entre la consigne et les mesures est inférieure à un certain seuil. Or ce seuil risque normalement d'être violé en fonctionnement dynamique, ce qui implique qu'il n'est pas possible de surveiller le système lors de fortes évolutions de la consigne. La comparaison entre le fonctionnement réel du système et le fonctionnement du modèle (i.e. : la comparaison $|y-y^M|$ entre la sortie réellement observée y et la sortie prédite y^M, ainsi que la comparaison $|u-u^M|$ entre la commande réelle u et la commande prédite u^M) va ainsi nous permettre de faire cette surveillance lors, entre autres, du fonctionnement dynamique du système.

Les différents types de modèles que nous considérons, un modèle parfait M_{parf} et un modèle embarqué M_{emb} dans notre cas, sont liés aux contraintes imposées par leurs utilisations. Le modèle embarqué M_{emb}, qui sera exploité en temps réel par le diagnostiqueur avec potentiellement des limites de capacité mémoire et de puissance de calcul, sera généralement linéarisé autour d'un point de fonctionnement. Le modèle parfait M_{parf}, qui sera quant à lui utilisé uniquement lors des phases de conception avec donc une capacité de mémoire et une puissance de calcul importantes, sera supposé représenter fidèlement le système physique et pourra donc être non-linéaire. Il risque ainsi de ne pas y avoir les mêmes seuils de comparaison entre ces deux modèles, donc entre le modèle embarqué M_{emb} et le système réel, suivant l'endroit où se trouve la consigne dans son domaine C, et ceci même en fonctionnement statique. Il

est donc nécessaire d'aussi prendre en compte ces différentes parties du domaine C de la consigne c où les valeurs des seuils des comparaisons seraient fortement différentes.

La formule normale φ_{F_0} va donc représenter ce que nous venons de décrire. D'une part la prise en compte des changements faibles ou forts de la consigne, et d'autre part les différentes parties du domaine C de la consigne c où les valeurs des seuils des comparaisons seraient fortement différentes.

La figure 4.9 de la page 135 représente ces deux observations que nous venons de présenter. Le trait discontinu rouge représente une instruction de l'opérateur évoluant arbitrairement et les traits continus bleu et vert représentent respectivement les mesures réelle et modèle. Nous pouvons remarquer que lorsque la consigne est inférieure à la valeur 27, les mesures réelle et modèle sont quasiment identiques et suivent cette consigne sauf lors de forts changements. Par contre lorsque la consigne est supérieure à la valeur 27, alors les mesures réelle et modèle sont fortement différentes. Par ailleurs, lorsque la consigne évolue fortement aux instants de temps t = 10 et t = 15, nous remarquons que les mesures réelle et modèle prennent un certain temps avant de suivre la consigne. Il s'agit de la dynamique de réponse du système.

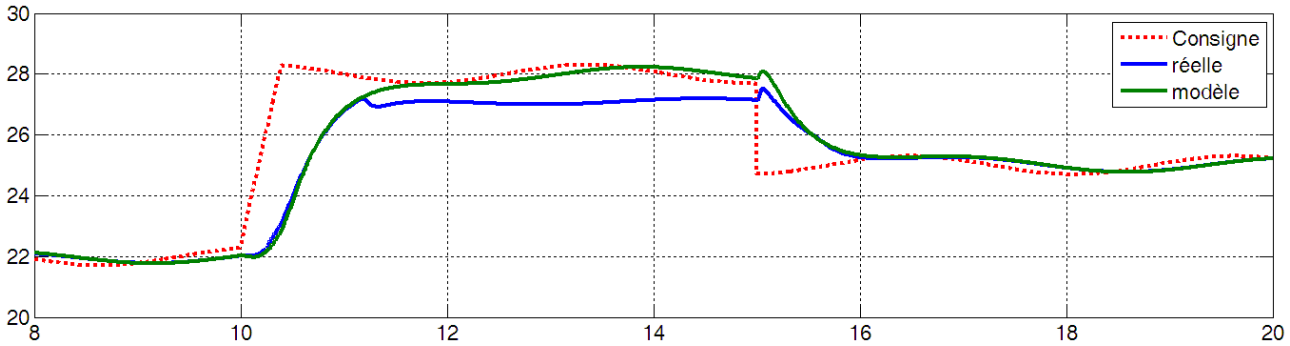

Figure 4.9 – Évolutions des mesures réelle et modèle suivant l'évolution de la consigne.

En fonctionnement statique, c'est-à-dire lors de changements de la consigne inférieurs à une certaine constante α, nous vérifierons non seulement que le système réel fonctionne conformément à cette consigne mais aussi qu'il fonctionne conformément au modèle embarqué. Nous vérifierons donc non seulement que la mesure réelle y suit la consigne c mais aussi d'une part que cette mesure réelle y concorde avec la mesure modèle y^M et d'autre part que la commande réelle u concorde avec la commande modèle u^M. En fonctionnement dynamique, c'est-à-dire lors des changements de la consigne supérieurs à cette constante α, nous vérifierons uniquement la conformité entre le fonctionnement du système et le fonctionnement du modèle, et ceci durant la dynamique de réponse β du système (que nous supposons identique entre le système et le modèle). Nous ne vérifierons donc que la concordance entre la mesure réelle y et la mesure modèle y^M.

Les différentes parties du domaine C de la consigne sont à déterminer. Suivant l'appartenance de la valeur de la consigne à une de ces parties, les seuils des comparaisons $|c - y|$, $|y - y^M|$ et $|u - u^M|$ vont être différents ; ceci en prenant aussi en compte les fonctionnements statiques ou dynamiques du système. Il est donc nécessaire d'obtenir un partitionnement du domaine de la consigne $C = \dot{\bigcup}_{i\in\iota} C^i$. Nous faisons pour cela évoluer la consigne c dans son domaine C puis nous traçons les nuages de points $(c, |c - y|)$, $(c, |y - y^M|)$ ainsi que $(c, |u - u^M|)$. Ces trois nuages de points doivent nous permettre d'obtenir un partitionnement $C = \dot{\bigcup}_{i\in\iota} C^i$ du domaine de la consigne.

La formule normale φ_{F_0} va donc prendre en compte des seuils adaptatifs de comparaison entre la consigne c et la mesure réelle y, entre les mesures réelle y et modèle y^M, ainsi qu'entre les commandes réelle u et modèle u^M.

Nous obtenons ainsi la formule temporelle normale φ_{F_0} par la conjonction de sous-formules $\varphi^i_{F_0}$:

$$\varphi_{F_0} : \bigwedge_{i \in I} \varphi^i_{F_0}$$

où chacune des sous-formules $\varphi^i_{F_0}$, pour $i \in I$, est donnée par :

$$\varphi^i_{F_0} : \left[(c \in C^i \wedge stat_c) \Rightarrow Stat_{C^i}\right] \wedge \left[(c \in C^i \wedge dyn_c) \Rightarrow Dyn_{C^i}\right]$$

$c \in C^i$ correspond à la formule $\left[(min(C^i) \leq c) \wedge (c \leq max(C^i))\right]$, la sous-formule $stat_c$ représente le fait que la variable de consigne c n'a pas fortement évolué depuis au moins β unités de temps et la sous-formule dyn_c représente le fait qu'elle a fortement évolué. La sous-formule $stat_c$ est formellement donnée par $\mathrm{G}_{[-\beta,0]}(|c - \mathrm{V}_{[-0.01]}c| < \alpha)$ et décrit bien que la variable de consigne c n'a pas fortement évolué depuis au moins β unités de temps. La formule dyn_c est formellement donnée par la formule $\mathrm{G}_{[-\beta,0]}(|c - \mathrm{V}_{[-0.01]}c| \geq \alpha)$ et décrit bien que la variable de consigne c a fortement évolué depuis β unités de temps. Remarquons qu'il s'agit encore d'abréviations dans le sens où la variable de consigne c est généralement un vecteur $c = (c_1, \ldots, c_r)$ et qu'il faut alors considérer les fortes ou faibles évolutions de chacun des termes du vecteur.

Pour chaque $i \in I$, la sous-formule $Stat_{C^i}$ représente les seuils adaptatifs statiques. Elle vérifie donc que les différences absolues $|c - y|$, $|y - y^M|$ et $|u - u^M|$ doivent être comprises entre des seuils adaptatifs minimums $\underline{\varepsilon}^{(stat,i)}$ et maximums $\overline{\varepsilon}^{(stat,i)}$ suivant la partie C^i. C'est-à-dire formellement :

$$\begin{aligned} Stat_{C^i} \; : \; [& \\ & \left(\underline{\varepsilon}^{(stat,i)}_{(c,y)} \leq |c - y|\right) \wedge \left(|c - y| \leq \overline{\varepsilon}^{(stat,i)}_{(c,y)}\right) \\ & \wedge \left(\underline{\varepsilon}^{(stat,i)}_{(y,y^M)} \leq |y - y^M|\right) \wedge \left(|y - y^M| \leq \overline{\varepsilon}^{(stat,i)}_{(y,y^M)}\right) \\ & \wedge \left(\underline{\varepsilon}^{(stat,i)}_{(u,u^M)} \leq |u - u^M|\right) \wedge \left(|u - u^M| \leq \overline{\varepsilon}^{(stat,i)}_{(u,u^M)}\right) \\ ] & \end{aligned}$$

De même pour chaque $i \in I$, la sous-formule Dyn_{C^i} représente les seuils adaptatifs dynamiques. Elle vérifie donc que la différence absolue $|y - y^M|$ doit être comprise entre des seuils adaptatifs minimums $\underline{\varepsilon}^{(dyn,i)}$ et maximums $\overline{\varepsilon}^{(dyn,i)}$ suivant la partie C^i. C'est-à-dire formellement :

$$Dyn_{C^i} : \left[\left(\underline{\varepsilon}^{(dyn,i)}_{(y,y^M)} \leq |y - y^M|\right) \wedge \left(|y - y^M| \leq \overline{\varepsilon}^{(dyn,i)}_{(y,y^M)}\right)\right]$$

Nous expliquerons plus en détail comment obtenir cette formule normale φ_{F_0} à la fin de ce chapitre lorsque nous traiterons la diagnosticabilité du cas d'étude. Indiquons néanmoins que nous utiliserons tous les comportements observables normaux $ObsB(cs, F_0, 0)$ de l'ensemble $ObsBeh_{Cons}(F_0)$ pour toute instruction $cs \in Cons$.

Les formules des défauts Toutes les formules des défauts $F \in \Gamma \setminus \{F_0\}$ sont élaborées en prenant en compte le comportement flt_F du défaut ainsi que son effet sur le système. Ces caractéristiques sont traduites, si possible, en propriétés temporelles décrivant comment le défaut perturbe les différences absolues $|c - y|$, $|y - y^M|$ et $|u - u^M|$, ainsi que les variables observables y, y^M, u et u^M.

Nous n'avons malheureusement pas trouvé de méthode générique permettant de définir une formule temporelle d'un défaut. Nous verrons néanmoins, lors de la mise en œuvre sur le cas d'étude à la fin de ce chapitre, que certains défauts seront facilement traduisibles en formules temporelles avec d'ailleurs de bons résultats lors de l'étude de la diagnosticabilité.

4.4.2.4 Traduction des notions de diagnosticabilité en formules temporelles

En considérant l'ensemble des formules temporelles pour chacun des défauts répertoriés, l'ensemble $\{\varphi_F / F \in \Gamma\}$, il est possible de traduire les différentes notions d'éligibilité, de détectabilité, d'isolabilité et de diagnosticabilité par des formules temporelles prenant en compte ces formules temporelles φ_F comme sous-formules. Il va falloir pour cela considérer de nouvelles ressources : un signal indiquant l'occurrence du défaut ainsi que de nouveaux opérateurs temporels permettant d'observer des fronts montants ou descendants de signaux.

Signal d'occurrence des défauts Considérons d'abord une nouvelle variable $sf_{\{\mathrm{t}_n\}}$ décrivant la validité temporelle de n'importe qu'elle occurrence t_n de n'importe quel défaut $F \in \Gamma \setminus \{F_0\}$. Pour une occurrence quelconque $\mathrm{t}_n \in \Omega_{(F,cs)}$ de n'importe quel défaut $F \in \Gamma$ suivant n'importe quelle instruction $cs \in Cons$, nous posons :

$$\begin{array}{rcccl} sf_{\{\mathrm{t}_n\}} & : & \mathrm{I} & \longrightarrow & \{0;1\} \\ & & \mathrm{t} & \longmapsto & \left\{ \begin{array}{l} 0, \text{ si } \mathrm{t} < \mathrm{t}_n \\ 1, \text{ si } \mathrm{t} \geq \mathrm{t}_n \end{array} \right. \end{array}$$

Remarquons que ce signal est totalement différent du signal flt_F d'un défaut F représentant le comportement de ce défaut. En effet, flt_F est à valeurs dans l'intervalle $[0;1]$ alors que $sf_{\{\mathrm{t}_n\}}$ est à valeurs dans l'ensemble $\{0;1\}$ composé uniquement des deux éléments 0 et 1. Ce signal doit nous fournir l'occurrence d'un défaut et nous l'interpréterons, dans une formule temporelle, comme un signal booléen.

Opérateurs temporels de fronts Considérons deux nouveaux opérateurs temporels Top et Bot permettant d'observer des fronts montants ou descendants. Bien que ces deux nouveaux opérateurs vont être donnés autant pour des termes que pour des formules temporelles, ils n'en auront néanmoins pas la même interprétation : ils fourniront des valeurs pour les termes alors qu'ils fourniront des validités pour les formules temporelles.

Pour un terme te, les formules $\mathrm{Top}(te)$ et $\mathrm{Bot}(te)$ sont aussi des termes dont la sémantique est donnée, toujours pour un comportement $Obs \in ObsBeh_{Cons}$ et un instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps, par :

- $(Obs, \mathrm{t}) \models \mathrm{Top}(te)$ si et seulement si $\mathrm{t} \in TDom(Obs)^{\lfloor}$ (l'ouverture à droite de $TDom(Obs)$) et te change de valeur à cet instant t ; la valeur de te est donc la nouvelle valeur à laquelle il passe à cet instant t.
- $(Obs, \mathrm{t}) \models \mathrm{Bot}(te)$ si et seulement si $\mathrm{t} \in TDom(Obs)^{\lfloor}$ et te change de valeur à cet instant t ; la valeur de te est donc l'ancienne valeur à laquelle il était jusque cet instant t.

Pour une formule temporelle φ, les formules $\mathrm{Top}(\varphi)$ et $\mathrm{Bot}(\varphi)$ sont aussi des formules temporelles dont la sémantique est donnée, pour un comportement $Obs \in ObsBeh_{Cons}$ et un instant $\mathrm{t} \in \mathbb{T}_\iota$ du temps, par :

- $(Obs, \mathrm{t}) \models \mathrm{Top}(\varphi)$ si et seulement si $\mathrm{t} \in TDom(Obs)^{\lfloor}$ et à cet instant t, φ passe de invalide à valide.
- $(Obs, \mathrm{t}) \models \mathrm{Bot}(\varphi)$ si et seulement si $\mathrm{t} \in TDom(Obs)^{\lfloor}$ et à cet instant t, φ passe de valide à invalide.

La figure 4.10 ci-dessous représente un flux de données que nous pouvons considérer comme un comportement observable Obs représenté par l'évolution d'une variable v sur la fenêtre temporelle $[0;10]$.

Le tableau 4.1 suivant indique les différentes valeurs de différentes formules évaluées par Obs aux instants de temps $\mathrm{t} = 2$, $\mathrm{t} = 4$, $\mathrm{t} = 6$ et $\mathrm{t} = 8$. Nous pouvons remarquer, grâce à cette figure et ce tableau, que la formule $\varphi_1 := (\mathrm{Top}(v) - \mathrm{Bot}(v) > 0)$ est valide aux instants de temps $\mathrm{t} = 2$ et $\mathrm{t} = 8$, que la formule $\varphi_2 := (\mathrm{Top}(v) - \mathrm{Bot}(v) \leq 0)$ est valide aux instants de temps $\mathrm{t} = 4$ et $\mathrm{t} = 6$, qu'enfin la

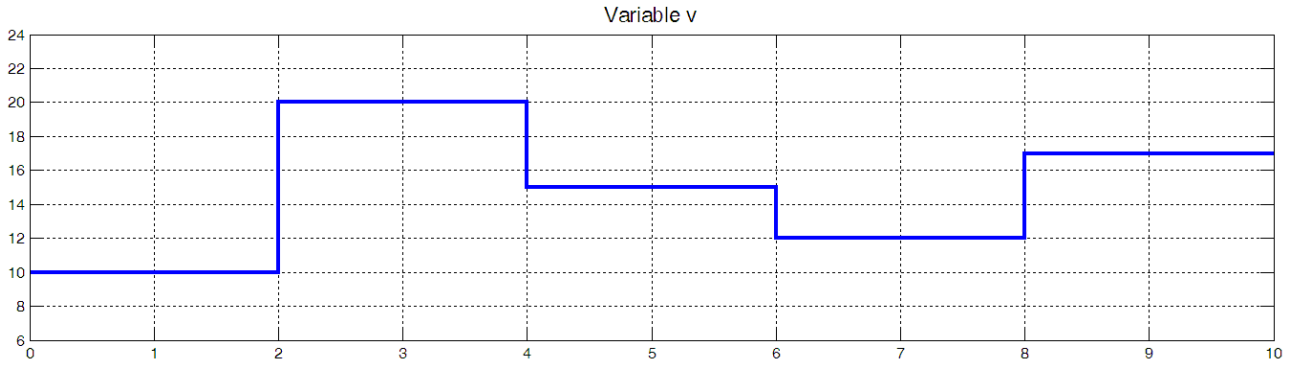

Figure 4.10 – Simulation d'une variable quelconque pour la sémantique des opérateurs temporels Top et Bot.

formule $\varphi_3 := \text{Top}(v > 16)$, où l'opérateur Top est appliqué à une formule atomique (donc temporelle), est valide aux instants de temps t = 2 et t = 8 car la formule $(v > 16)$ passe de invalide à valide en ces deux instants de temps.

	Instants			
Formules	t = 2	t = 4	t = 6	t = 8
Top(v)	20	15	12	17
Bot(v)	10	20	15	12
Top(v) − Bot(v)	10	−5	−3	5

Tableau 4.1 – Sémantique de différentes formules avec les opérateurs temporels Top et Bot.

La notion de diagnosticabilité Pour rappel, cette notion de diagnosticabilité est définie différemment selon le défaut considéré : c'est-à-dire le cas normal $F_0 \in \Gamma$ ou un « vrai » défaut $F \in \Gamma \setminus \{F_0\}$.

Pour le cas normal $F_0 \in \Gamma$, il faut que la formule normale φ_{F_0} soit toujours validée lorsqu'il n'y a pas de défaut. Nous vérifions pour cela que tout comportement observable normal $ObsB(cs, F_0, 0) \in ObsBeh_{Cons}(F_0)$, suivant une instruction quelconque cs de $Cons$, valide sur l'intervalle $[\lambda; max(\text{I})]$ la formule Ψ_{F_0} suivante :

$$\Psi_{F_0} \ := \ \varphi_{F_0}$$

Pour un défaut $F \in \Gamma \setminus \{F_0\}$, il faut qu'après son occurrence et dans un temps borné par b, la formule normale φ_{F_0} devienne invalide, puis que seule la formule φ_F du défaut reste valide suite à cette invalidité et jusqu'au délai δ de confiance. Cela signifie qu'il faut (et qu'il suffit) donc que tout comportement observable $ObsB(cs, F, \text{t}_n) \in ObsBeh_{Cons}(F)$, suivant une instruction quelconque cs de $Cons$ et sous la présence de ce défaut F à une occurrence quelconque $\text{t}_n \in \Omega_{(F,cs)}$, valide la propriété Ψ_F suivante :

$$\Psi_F \ := \ \text{Top}(sf_{\{\text{t}_n\}}) \Rightarrow [\varphi_{F_0} \text{U}_{[0;b-h]} (\text{Bot}(\varphi_{F_0}) \wedge \text{G}_{[h,h+\delta]} [\varphi_F \wedge \bigwedge_{F' \in \Gamma \setminus \{F\}} \neg \varphi_{F'}])]$$

Nous pouvons remarquer que si la propriété Ψ_F est validée, elle l'est alors forcément à l'instant de temps t_n, instant étant d'ailleurs l'occurrence du défaut. En effet, cette formule Ψ_F contient deux sous-formules liées par l'opérateur d'implication $\Rightarrow$ et dont la première sous-formule $\text{Top}(sf_{\{\text{t}_n\}})$ ne peut être validée qu'à l'instant de temps t_n où la variable $sf_{\{\text{t}_n\}}$ passe de 0 à 1.

Par ailleurs la seconde sous-formule, qui ne peut donc être valide qu'à l'instant de temps t_n, à cause de la validité de la sous-formule $\text{Top}(sf_{\{\text{t}_n\}})$, exprime que la formule normale φ_{F_0} doit être

valide jusqu'à ce qu'elle passe à invalide à un instant t_k de détection dans la fenêtre temporelle $[t_n; t_n + (b-h)]$ (ce qu'exprime le morceau $\varphi_{F_0} U_{[0;b-h]} \text{Bot}(\varphi_{F_0})$), et que suite à cette invalidité, seule la formule φ_F du défaut doit être valide sur la fenêtre temporelle $[t_k + h, t_k + (h+\delta)]$ (ce qu'exprime le morceau $G_{[h,h+\delta]}[\varphi_F \wedge \bigwedge_{F' \in \Gamma \setminus \{F\}} \neg \varphi_{F'}]$).

La notion d'éligibilité Comme pour la diagnosticabilité, cette notion d'éligibilité est définie différemment selon le défaut considéré : c'est-à-dire le cas normal $F_0 \in \Gamma$ ou un « vrai » défaut $F \in \Gamma \setminus \{F_0\}$. Pour le cas normal $F_0 \in \Gamma$ il faut, comme pour la diagnosticabilité, que la formule normale φ_{F_0} soit toujours validée lorsqu'il n'y a pas de défaut. Pour un « vrai » défaut $F \in \Gamma \setminus \{F_0\}$ il faut que sa formule φ_F soit validée, dans un temps borné par b, après son occurrence.

Pour le cas normal $F_0 \in \Gamma$, il faut donc vérifier que tout comportement observable normal $ObsB(cs, F_0, 0) \in ObsBeh_{Cons}(F_0)$, suivant une instruction quelconque cs de $Cons$, valide sur l'intervalle $[\lambda; max(\mathrm{I})]$ la formule $\Psi^e_{F_0}$ suivante :

$$\Psi^e_{F_0} \ := \ \varphi_{F_0}$$

Pour un défaut $F \in \Gamma \setminus \{F_0\}$: il faut vérifier que tout comportement observable $ObsB(cs, F, t_n) \in ObsBeh_{Cons}(F)$, suivant une instruction quelconque cs de $Cons$ et sous la présence de ce défaut F à une occurrence quelconque $t_n \in \Omega_{(F,cs)}$, valide la propriété Ψ^e_F suivante :

$$\Psi^e_F \ := \ \text{Top}(sf_{\{t_n\}}) \Rightarrow E_{[0;b-h]}(G_{[h,h+\delta]}\varphi_F)$$

Comme pour la diagnosticabilité, si cette propriété Ψ^e_F est validée, elle ne l'est alors forcément qu'à l'instant de temps t_n. Par ailleurs la sous-formule $E_{[0;b-h]}(G_{[h,h+\delta]}\varphi_F)$, qui ne peut être valide qu'à l'instant t_n à cause de la validité de la sous-formule $\text{Top}(sf_{\{t_n\}})$, exprime bien qu'il existe un instant de temps t_k dans la fenêtre temporelle $[t_n; t_n + (b-h)]$ (ce qu'exprime l'opérateur $E_{[0;b-h]}$) tel que la formule φ_F de ce défaut est valide sur toute la fenêtre temporelle $[t_k + h; t_k + (h+\delta)]$ (ce qu'exprime le morceau $G_{[h,h+\delta]}\varphi_F$).

La notion de détectabilité Cette notion n'est définie que pour les « vrais » défauts : c'est-à-dire les défauts $F \in \Gamma$ autres que le cas cas normal F_0. Un défaut $F \in \Gamma \setminus \{F_0\}$ est détectable si la formule normale φ_{F_0} passe de valide à invalide, dans un temps borné par b, après son occurrence. Il faut donc vérifier que tout comportement observable $ObsB(cs, F, t_n) \in ObsBeh_{Cons}(F)$, suivant une instruction quelconque cs de $Cons$ et sous la présence de ce défaut F à une occurrence quelconque $t_n \in \Omega_{(F,cs)}$, valide la propriété Ψ^d suivante :

$$\Psi^d \ := \ \text{Top}(sf_{\{t_n\}}) \Rightarrow [\varphi_{F_0} U_{[0;b-h]}(\text{Bot}(\varphi_{F_0}))]$$

Encore une fois, cette propriété Ψ^d ne peut être validée qu'à l'instant de temps t_n. Remarquons de plus que cette propriété est définie indépendamment du défaut $F \in \Gamma \setminus \{F_0\}$ considéré : c'est-à-dire que ce sera toujours cette formule qui sera utilisée pour vérifier la détectabilité de n'importe quel défaut $F \in \Gamma \setminus \{F_0\}$. Par ailleurs la sous-formule $\varphi_{F_0} U_{[0;b-h]}(\text{Bot}(\varphi_{F_0}))$, qui ne peut être valide qu'à l'instant t_n à cause de la validité de la sous-formule $\text{Top}(sf_{\{t_n\}})$, exprime bien que la formule normale φ_{F_0} doit être valide jusqu'à ce qu'elle passe à invalide à un instant t_k de détection dans la fenêtre temporelle $[t_n; t_n + (b-h)]$.

La notion d'isolabilité Cette notion n'est, elle aussi, définie que pour les « vrais » défauts : c'est-à-dire les défauts $F \in \Gamma$ autres que le cas cas normal F_0. Un défaut $F \in \Gamma \setminus \{F_0\}$ est isolable si toutes les formules $\varphi_{F'}$ des autres défauts $F' \in \Gamma \setminus \{F\}$ sont invalides, dans un temps borné par b après son occurrence et jusqu'au délai δ de confiance. Il faut donc vérifier que tout comportement observable

$ObsB(cs, F, \mathrm{t}_n) \in ObsBeh_{Cons}(F)$, suivant une instruction quelconque cs de $Cons$ et sous la présence de ce défaut F à une occurrence quelconque $\mathrm{t}_n \in \Omega_{(F,cs)}$, valide la propriété Ψ^i_F suivante :

$$\Psi^i_F \ := \ \mathrm{Top}(sf_{\{\mathrm{t}_n\}}) \Rightarrow [\mathrm{E}_{[0;b-h]}(\mathrm{G}_{[h,h+\delta]}[\bigwedge_{F' \in \Gamma \setminus \{F\}} \neg\varphi_{F'}])]$$

Comme pour les notions précédentes, cette propriété Ψ^i_F ne peut être validée qu'à l'instant de temps t_n. Par ailleurs la sous-formule $\mathrm{E}_{[0;b-h]}(\mathrm{G}_{[h,h+\delta]}[\bigwedge_{F' \in \Gamma \setminus \{F\}} \neg\varphi_{F'}])$, qui ne peut être valide qu'à l'instant t_n à cause de la validité de la sous-formule $\mathrm{Top}(sf_{\{\mathrm{t}_n\}})$, exprime bien qu'il existe un instant de temps t_k dans la fenêtre temporelle $[\mathrm{t}_n; \mathrm{t}_n+(b-h)]$ (ce qu'exprime l'opérateur $\mathrm{E}_{[0;b-h]}$) tel que toutes les formules $\varphi_{F'}$ des autres défauts $F' \in \Gamma \setminus \{F\}$ sont invalides sur toute la fenêtre temporelle $[\mathrm{t}_k + h; \mathrm{t}_k + (h + \delta)]$ (ce qu'exprime le morceau $\mathrm{G}_{[h,h+\delta]}[\bigwedge_{F' \in \Gamma \setminus \{F\}} \neg\varphi_{F'}]$).

4.4.2.5 Étude de la diagnosticabilité du système par utilisation de la caractérisation par formules temporelles

En ayant traduit, par des formules temporelles, ces quatre notions d'éligibilité, de détectabilité, d'isolabilité et de diagnosticabilité, il est ainsi facile de procéder à l'étude de la diagnosticabilité du système suivant cette caractérisation par formules temporelles.

Diagnosticabilité du cas normal La première étape consiste à vérifier la diagnosticabilité du cas normal $F_0 \in \Gamma$. Comme nous l'avons dit, il faut que la formule normale φ_{F_0} soit toujours valide lorsqu'il n'y a pas de défaut. Pour cela nous vérifions que tout comportement observable normal $ObsB(cs, F_0, 0) \in ObsBeh_{Cons}(F_0)$, suivant une instruction quelconque cs de $Cons$, valide toujours la formule $\Psi^e_{F_0}$ $(:= \varphi_{F_0})$ sur la fenêtre temporelle $[\mathrm{t}_0; max(\mathrm{I})]$.

Dans le cas où ce cas normal F_0 n'est pas diagnosticable : c'est-à-dire qu'au moins un comportement observable $ObsB(cs, F_0, 0) \in ObsBeh_{Cons}(F_0)$ n'a pas validé la formule $\Psi^e_{F_0}$ (i.e. : n'a donc pas validé la formule normale φ_{F_0}) à un instant t_i de la fenêtre temporelle $[\mathrm{t}_0; max(\mathrm{I})]$, cela signifie soit que la phase d'apprentissage des différents seuils de φ_{F_0} ne s'est pas bien déroulée, soit que cette formule normale φ_{F_0} n'est pas assez explicite. Dans les deux cas, il faut d'abord affiner cette phase d'apprentissage en prenant en compte l'instruction $cs \in Cons$ pour laquelle le comportement observable normal $ObsB(cs, F_0, 0)$ suivant cette instruction cs n'a pas satisfait cette formule normale φ_{F_0}. Si après ré-étude de la diagnosticabilité, ce cas normal F_0 n'est toujours pas diagnosticable, il faudra augmenter cette formule temporelle normale φ_{F_0} par des sous-formules tenant compte des spécificités l'ayant rendu non-diagnosticable.

Comme nous l'avons indiqué lors de la construction de cette formule normale φ_{F_0}, nous verrons à la fin de ce chapitre lorsque nous traiterons la diagnosticabilité du cas d'étude comment obtenir cette formule normale φ_{F_0}. Nous avions précisé que cette construction se fera en prenant en compte tous les comportements observables normaux $ObsB(cs, F_0, cs)$ de l'ensemble $ObsBeh_{Cons}(F_0)$ pour toute instruction $cs \in Cons$. Par conséquent ce cas normal F_0 doit donc être diagnosticable car les phases d'apprentissage de la formule normale φ_{F_0} et d'étude de la diagnosticabilité utilisent les mêmes comportements observables.

Diagnosticabilité des défauts Les étapes suivantes consistent à étudier la diagnosticabilité de chacun des défauts $F \in \Gamma \setminus \{F_0\}$. Dans le cas formel, cela consisterait à vérifier la satisfaction des formules Ψ_F de diagnosticabilité, pour chacun des défauts $F \in \Gamma \setminus \{F_0\}$, par tout comportement observable $ObsB(cs, F, \mathrm{t}_n) \in ObsBeh_{Cons}(F)$ suivant n'importe quelle instruction cs de $Cons$ et sous la présence du défaut F à toute occurrence $\mathrm{t}_n \in \Omega_{(F,cs)}$. Néanmoins, cela risque d'être long car dans le cas où un défaut n'est pas diagnosticable, il faudrait vérifier la satisfaction des formules Ψ^e_F d'éligibilité, Ψ^d de détectabilité et Ψ^i_F d'isolabilité afin de déterminer laquelle n'est pas satisfaite.

La solution plus rapide serait donc de mener la vérification de la satisfaction des formules Ψ_F de diagnosticabilité en même temps que la vérification de la satisfaction des formules Ψ^e_F d'éligibilité, Ψ^d de détectabilité et Ψ^i_F d'isolabilité. Remarquons par ailleurs que dans la pratique, la vérification de la diagnosticabilité de défauts non préalablement isolables n'est pas nécessaire. Cela signifie que l'isolabilité d'un défaut ne se considère que s'il est détectable et qu'avec les autres défauts détectables. En effet comme l'étude de la diagnosticabilité est inhérente au processus d'analyse du diagnostiqueur, si un défaut n'est pas détectable, il ne sera alors pas détecté par le diagnostiqueur. Ainsi, le comportement observable du système sous la présence du défaut sera donc analysé par le diagnostiqueur comme normal. Il est donc inutile, en pratique, d'en étudier son isolabilité.

Par conséquent, dans le cas où un défaut $F \in \Gamma \setminus \{F_0\}$ n'est pas diagnosticable, cela signifie qu'au moins un comportement observable $ObsB(cs, F, \mathrm{t}_n) \in ObsBeh_{Cons}(F)$ n'a pas satisfait la formule Ψ_F de diagnosticabilité. Comme nous avons en même temps vérifié les formules Ψ^e_F d'éligibilité, Ψ_F de détectabilité et Ψ^i_F d'isolabilité, soit ces formules sont toutes satisfaites soit elles ne le sont pas toutes. Dans le cas où elles le sont toutes, cela implique que les instants t_{ke} d'éligibilité, t_{kd} de détectabilité et t_{ki} d'isolabilité ne sont pas tels que $\mathrm{t}_{ke} \leq \mathrm{t}_{kd}$ et $\mathrm{t}_{ki} \leq \mathrm{t}_{kd}$, et cela signifie donc que soit la borne b de diagnostic, soit le délai h d'isolation, ou soit l'information disponible ne sont pas adéquats. Dans le cas où elles ne sont pas toutes satisfaites, cela nous permet de vérifier que soit l'éligibilité, soit la détectabilité ou soit l'isolabilité de F ne sont pas vérifiées.

4.4.2.6 Justification de la caractérisation par formules temporelles

Cette caractérisation par formules temporelles nous est apparue adéquate pour introduire une notion temporelle dans la comparaison entre le fonctionnement réel du système et celui du modèle embarqué. En effet et comme nous l'avions souligné dans l'introduction générale et durant la présentation des méthodologies de diagnostic au chapitre 1, les méthodologies de diagnostic à base de modèles, notamment celles utilisant une modélisation continue du système, réalisent une comparaison entre le fonctionnement réel du système et le fonctionnement obtenu par un modèle embarqué. Or, cette comparaison se réalise à chaque instant du temps sans prendre en compte cet aspect temporel du fonctionnement du système. Les seuils de comparaison sont donc identiques quel que soit le fonctionnement du système. Nous avons donc intégré cet aspect temporel afin de rendre les seuils de comparaison adaptatifs au fonctionnement du système : c'est-à-dire prendre en compte les changements de la consigne du système. Nous avons par ailleurs ajouté d'autres caractéristiques, comme les différences de points de fonctionnement entre le modèle et le système, afin de rendre cette caractérisation par formules temporelles plus robuste.

4.5 Application sur le cas d'étude

Nous allons dans cette partie appliquer l'ensemble du cadre théorique présenté dans ce chapitre sur le cas d'étude introduit au chapitre 2 : la ligne d'air d'un système pile à combustible. Rappelons qu'au chapitre 3 de typologie des défauts, nous avons introduit différents défauts de cette ligne d'air :

- un blocage du compresseur $F_{LockCmpr}$,
- un encrassement du compresseur $F_{DirtCmpr}$,
- un blocage de l'électrovanne F_{LockEV},
- un encrassement de l'électrovanne F_{DirtEV},
- une fuite d'air entre le compresseur et l'humidificateur due à une usure normale $F_{LeakAir}$,
- un défaut de mesure du capteur de débit F_{SenQ},
- un défaut de mesure du capteur de pression F_{SenP}.

En notant F_{Norm} cette ligne d'air sans défaut, nous considérons donc l'ensemble suivant de défauts de ligne d'air :

$$\Gamma_{AirLine} = \{F_{norm}; F_{LockCmpr}; F_{DirtCmpr}; F_{LockEV}; F_{DirtEV}; F_{LeakAir}; F_{SenQ}; F_{SenP}\}$$

Après avoir indiqué les différents paramètres de la ligne d'air et suite à la définition des comportements observables de cette ligne d'air sous la présence de tous ces défauts $F \in \Gamma_{AirLine}$, nous allons obtenir les propriétés des défauts qui vont nous permettre d'étudier la diagnosticabilité de cette ligne d'air.

4.5.1 Préliminaires

Rappelons au préalable les différentes variables observables de cette ligne d'air, et présentons de plus les différents paramètres en permettant une étude de diagnosticabilité.

4.5.1.1 Variables observables de la ligne d'air

Le rôle de cette ligne d'air est de fournir la pile en air, dont l'oxygène en est consommé, à un débit et une pression ciblés. L'ensemble $\overline{\mathrm{V}}_{obs} = \{c_Q; c_P; u_\omega; u_x; y_Q; y_P; u_\omega^M; u_x^M; y_Q^M; y_P^M\}$ des variables observables de cette ligne d'air est récapitulé dans le tableau 4.2 ci-dessous, avec leurs différents domaines de valeurs. Rappelons de plus que le produit de tous les domaines des variables observables de cette ligne d'air est $\overline{\mathrm{VD}}_{obs} = C_Q \times C_P \times U_\omega \times U_x \times Y_Q \times Y_P \times U_\omega^M \times U_x^M \times Y_Q^M \times Y_P^M$.

Nom	Notation	Notation du domaine de valeurs
Consigne de débit	c_Q	C_Q
Consigne de pression	c_P	C_Q
Mesure du débit de la boucle réelle	y_Q	Y_Q
Mesure de la pression de la boucle réelle	y_P	Y_Q
Commande du compresseur de la boucle réelle	u_ω	U_ω
Commande de l'électrovanne de la boucle réelle	u_x	U_x
Mesure du débit de la boucle modèle	y_Q^M	Y_Q^M
Mesure de la pression de la boucle modèle	y_P^M	Y_P^M
Commande du compresseur de la boucle modèle	u_ω^M	U_ω^M
Commande de l'électrovanne de la boucle modèle	u_x^M	U_x^M

Tableau 4.2 – Variables observables de la ligne d'air.

4.5.1.2 Paramètres de la ligne d'air

Rappelons que la période d'échantillonnage de cette ligne d'air est $\iota = 0.01$ seconde et que son temps d'exécution est $\mathbb{T}_{0.01}$. Par ailleurs, différents paramètres permettant une étude de diagnosticabilité ont été décrits au début de ce chapitre : la dynamique de réponse β du système, la borne b de diagnostic, la longueur λ d'enregistrement du diagnostiqueur, le délai h d'isolation ainsi que le délai δ de confiance des validités des propriétés. Concernant la ligne d'air nous allons donner ces différents paramètres.

La dynamique de réponse β de la ligne d'air vaut 3 secondes. Ce paramètre provient des exigences du cahier des charges du système pile à combustible et fut par la suite vérifié lors des tests expérimentaux.

La borne b de diagnostic vaut 6 secondes. Nous l'avons estimée suivant la dynamique de réponse β de la ligne d'air : comme nous cherchons à rapidement diagnostiquer un défaut suite à son occurrence

mais que la contre-réaction de la commande du système peut influencer l'impact visible du défaut, nous avons donc estimé que la borne b de diagnostic devait être supérieure à cette dynamique de réponse β de la ligne d'air. Notons que nous avons choisi arbitrairement cette borne comme égale à 6 secondes, toute valeur supérieure à 3 secondes aurait pu correspondre et nous aurions même pu tester la borne b de diagnostic la plus petite, cela en augmentant bien sûr le temps de conception.

La longueur d'enregistrement λ du diagnostiqueur va valoir deux valeurs différentes suivant la caractérisation considérée. Concernant la caractérisation parfaite, nous allons considérer des morceaux de comportements observables de longueur b et la longueur d'enregistrement λ va donc valoir 6 secondes. Concernant la caractérisation par formules temporelles, en toute généralité cette longueur vaut le maximum des longueurs des intervalles des opérateurs temporels. Pour la ligne d'air et les défauts considérés, la formule normale φ_{F_0} va, comme nous l'avons indiqué, être construite en considérant la dynamique de réponse β de cette ligne d'air. Les autres formules vont, pour leur part, être construites suivant l'effet du défaut sur les variables observables du système et nous verrons, lorsqu'il sera possible de les construire, qu'elles ne seront pas aussi complexes que la formule normale et auront besoin d'un enregistrement inférieur à cette dynamique de réponse β du système. Nous allons donc poser $\lambda = 3$ secondes pour cette caractérisation par formules temporelles.

Le délai d'isolation h vaut 2 secondes. Nous avons fixé ce délai de manière arbitraire, tout en gardant à l'esprit la remarque de [VRYK03] indiquant que la phase de détection est généralement assez rapide par rapport à la phase d'isolation.

Le délai δ de confiance des validités des propriétés vaut 3 secondes. Comme ce paramètre est fortement lié à la dynamique de réponse β du système, nous l'avons supposé égal à cette dynamique.

4.5.2 Comportements observables de la ligne d'air

Les comportements observables de la ligne d'air, suivant les défauts répertoriés, ont été obtenus par simulation de cette ligne d'air dans l'outil de simulation MATLAB/Simulink©. Ces simulations ont été réalisées suivant différentes instructions et pour différentes occurrences de défauts que nous allons au préalable définir.

4.5.2.1 Ensemble des instructions

Comme nous l'avons précédemment indiqué au chapitre 2 lors de la présentation de ce cas d'étude, la ligne d'air reçoit ses consignes de débit c_Q et pression c_P d'air de la part du système de pilotage global du système pile à combustible. La figure 4.11 suivante de la page 144 représente une simulation de ces deux consignes, la consigne du débit d'air sur le premier graphique et la consigne de pression d'air sur le second graphique, durant une fenêtre temporelle de longueur 50 secondes. Ces deux consignes reflètent une fonction aléatoire comme lors d'une utilisation urbaine du véhicule où le conducteur accélère et décélère en permanence avec des niveaux et des périodes plus ou moins élevés.

Nous remarquons que ces deux consignes sont corrélées. En effet et comme nous l'avons indiqué au chapitre 2 lors de la présentation de ce cas d'étude, le débit d'air est calculé suivant d'une part la demande de puissance électrique requise par le superviseur du véhicule et d'autre part en accord avec l'exigence de stœchiometrie. La pression d'air est ensuite déduite de ce débit et en accord avec l'exigence de pression maximale autorisée dans la pile et la ligne. Il est donc normal que la consigne de pression soit déduite de celle du débit. Pour la suite, nous considérerons d'ailleurs que cette consigne de débit est la consigne principale.

Cette consigne de débit évolue dans l'ensemble des valeurs réelles $[0; 30]$. Néanmoins, les valeurs très faibles sont difficilement atteignables du fait que le compresseur est rarement mis à l'arrêt pour éviter des consommations énergétiques supplémentaires. Il fonctionne donc toujours même à faible vitesse et produit ainsi toujours un certain débit. Pour ce cas d'étude et pour ne pas alourdir le travail, nous

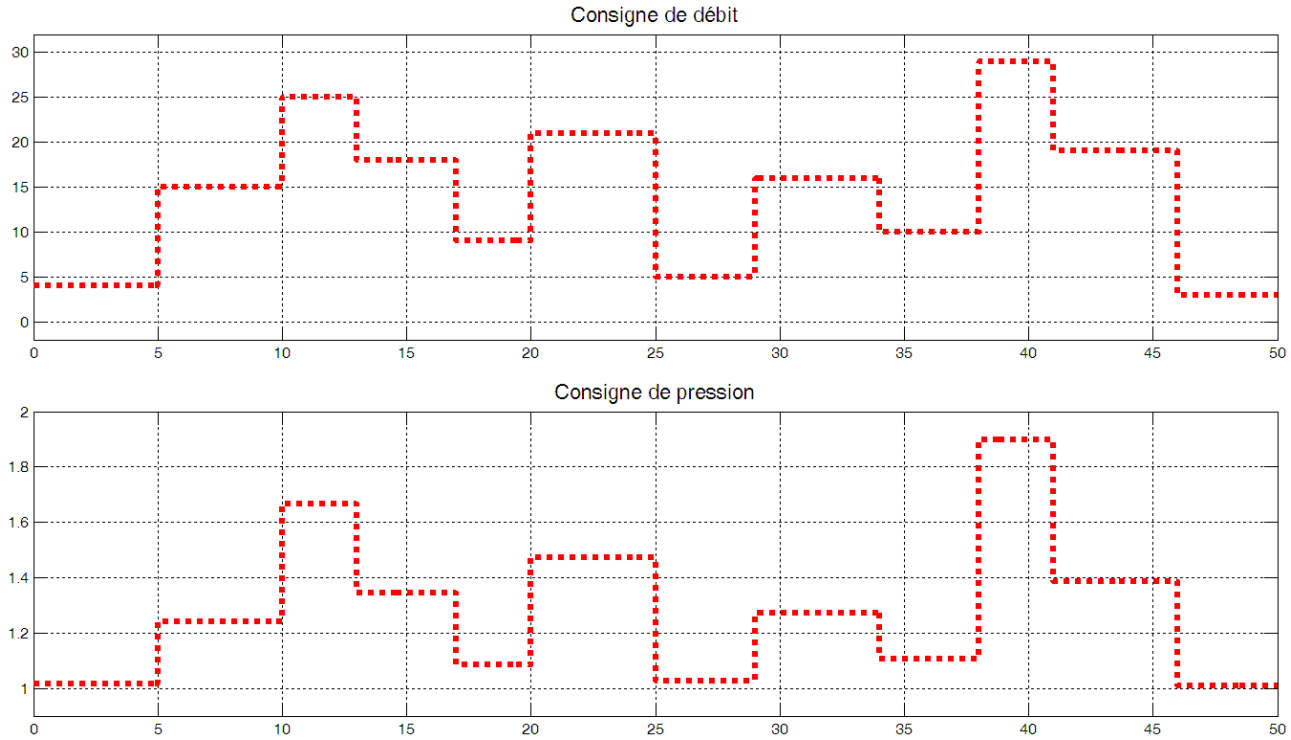

Figure 4.11 – Simulation d'une instruction de la ligne d'air.

n'avons considéré que des consignes entières et évoluant de manière constante par morceaux. L'étude est bien sûr tout à fait possible d'une part avec des consignes évoluant de manière continue, ce qui changera néanmoins l'ensemble des instructions, les ensembles d'occurrences des défauts ainsi que les formules temporelles, et d'autre part avec des valeurs plus fines (à 10^{-2} par exemple). Nous avons donc considéré l'ensemble des valeurs de la consigne de débit $C_Q = \{0; 1; 2; \ldots; 29; 30\} \subseteq \mathbb{N}$.

En supposant que la consigne soit toujours entière et évolue de manière constante par morceaux, nous avons établi un ensemble $Cons_{AirLine}$ d'instructions évoluant de la fenêtre temporelle $\mathrm{I} = [0; 50]$ vers l'ensemble des valeurs $[0; 30]$. Chaque instruction de cet ensemble est définie, pour deux constantes entières a et b de l'ensemble C_Q, par :

$$cs_{(a,b)} : [0;50] \longrightarrow [0;30]$$
$$\mathrm{t} \longmapsto \begin{cases} 10 & \text{si t} \in [0;10[\\ a & \text{si t} \in [10;30[\\ b & \text{si t} \in [30;50] \end{cases}$$

Cet ensemble d'instructions $Cons_{AirLine}$ est ainsi défini par $Cons_{AirLine} = \{cs_{(a,b)}/a, b \in C_Q\} \subseteq [0;30]^{[0;50]}$. Chaque instruction démarre à la valeur 10 pour nous permettre d'initialiser le système. Nous fixons par ailleurs la durée t_0 d'initialisation de manière à ce que le système soit stabilisé sur la valeur a de la consigne : c'est-à-dire à l'instant $\mathrm{t}_0 = 20$ secondes. En effet, comme le système a une dynamique de réponse β maximum de 3 secondes, nous posons donc $\mathrm{t}_0 = 20$ secondes afin d'être sûr de ne pas subir les effets du changement de la consigne à l'instant $\mathrm{t} = 10$ secondes (i.e. : au passage de la valeur 10 à la valeur a). Nous aurions pu considérer $\mathrm{t}_0 = 13$ secondes en accord avec cette dynamique de réponse, mais nous avons préféré attendre plus longtemps afin d'avoir des valeurs temporelles plus simples à analyser (en considérant qu'une valeur telle que 20 secondes est plus simple à considérer qu'une valeur telle que 13 secondes). Cet ensemble $Cons_{AirLine}$ d'instructions est donc composé de 961 instructions.

Comme nous l'avons indiqué, nous cherchons à travers cet ensemble $Cons_{AirLine}$ d'instructions à être représentatif de tous les comportements potentiels du système. Ces instructions $cs_{(a,b)}$ permettent donc de refléter les fonctionnements statique et dynamique de la ligne d'air. Ainsi suivant que les constantes entières a et b de l'ensemble C_Q sont différentes ou égales, les « formes » des instructions ne sont donc pas identiques : lorsque ces deux valeurs sont égales, cela nous permet de refléter un fonctionnement totalement statique, alors que lorsqu'elles sont différentes, cela nous permet de refléter une alternance entre un fonctionnement dynamique et un fonctionnement statique. Le fonctionnement statique correspond aux fenêtres temporelles pour lesquelles la consigne n'a pas changé alors que le fonctionnement dynamique correspond aux fenêtres temporelles $[t_c; t_c + 3[$ après chacun des instants t_c pour lesquels la consigne a changé. Pour une instruction $cs_{(a,a)} \in Cons_{AirLine}$, pour $a \in C_Q$, le fonctionnement statique correspond à la fenêtre temporelle $[20; 50]$ complète. Pour une instruction $cs_{(a,b)} \in Cons_{AirLine}$, avec $a, b \in C_Q$ où $a \neq b$ et en considérant l'instant $t_c = 30$ secondes où la consigne change, le fonctionnement dynamique correspond à la fenêtre temporelle $[30; 33[$ et le fonctionnement statique correspond aux fenêtres temporelles $[20; 30[$ et $[33; 50]$.

La figure 4.12 suivante de la page 145 représente l'instruction $cs_{(25,14)}$ avec les mesures réelle y_Q et modèle y_Q^M. Dans le graphique, le trait discontinu rouge représente la consigne et les traits continus bleu et vert représentent respectivement ces mesures réelle et modèle. Cette instruction $cs_{(25,14)}$ est définie de la fenêtre temporelle $I = [0; 50]$ vers l'ensemble des valeurs $[0; 30]$ et vaut 10 sur la fenêtre temporelle $[0; 10[$, 25 sur la fenêtre temporelle $[10; 30[$ et 14 sur la fenêtre temporelle $[30; 50]$. Nous remarquons bien les différents fonctionnements statiques et dynamiques durant la fenêtre temporelle $[20; 50]$, ce que nous avons représenté en violet et orange en dessous du graphique. À l'instant $t_c = 30$ secondes, la consigne change de la valeur 25 à la valeur 14 et ainsi durant la fenêtre temporelle $[30; 33[$ la ligne d'air est en fonctionnement dynamique : les mesures réelle et modèle ne suivent plus cette consigne. Durant les fenêtres temporelles $[20; 30[$ et $[33; 50]$, la consigne n'a pas évolué depuis au moins 3 secondes et ainsi la ligne d'air est en fonctionnement statique : les mesures réelle et modèle suivent cette consigne.

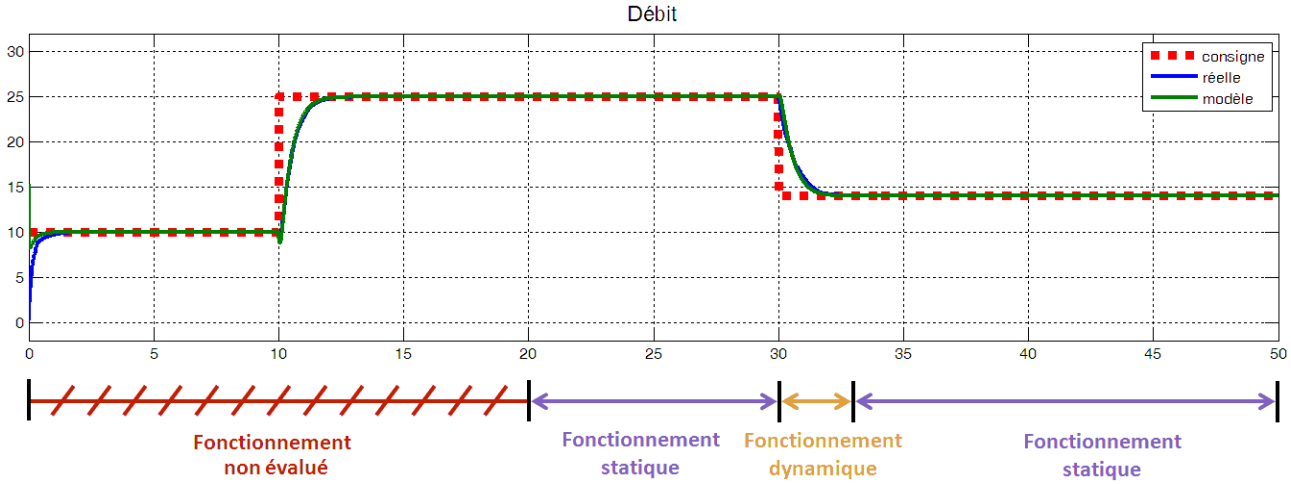

Figure 4.12 – Fonctionnements statiques et dynamiques de la ligne d'air suivant l'instruction $cs_{(25,14)}$.

4.5.2.2 Ensembles d'occurrences des défauts

Pour rappel, l'occurrence d'un défaut désigne l'instant t_n du temps où il apparaît : le système fonctionne normalement avant cet instant t_n et sous la présence du défaut après. Nous allons définir les ensembles d'occurrences $\Omega_{(F,cs_{(a,b)})}$ pour tous les défauts F de l'ensemble $\Gamma_{AirLine} \setminus \{F_{Norm}\}$ de la ligne d'air et toutes les instructions $cs_{(a,b)}$ de l'ensemble $Cons_{AirLine}$.

Comme nous venons de le voir, suivant que les constantes entières a et b de l'ensemble C_Q sont différentes ou égales, les « formes » des instructions ne sont donc pas identiques. Lorsque ces deux valeurs sont égales, cela nous permet de refléter un fonctionnement totalement statique du système, alors que lorsqu'elles sont différentes, cela nous permet de refléter une alternance entre un fonctionnement dynamique et un fonctionnement statique. Comme les variables observables n'évoluent pas lorsque la ligne est en fonctionnement statique, sous réserve d'un modèle représentatif et de la non présence de perturbations, il n'est donc pas nécessaire d'étudier un nombre important d'occurrences dans les fenêtres temporelles liées au fonctionnement statique de la ligne d'air. Pour n'importe quel défaut $F \in \Gamma_{AirLine} \setminus \{F_{Norm}\}$, nous avons par conséquent déterminé deux types d'occurrences : les occurrences statiques et les occurrences dynamiques suivant les instructions $cs_{(a,b)}$.

Occurrences pour les instructions statiques Pour les instructions dites « statiques », c'est-à-dire les instructions du type $cs_{(a,a)}$ avec $a \in C_Q$, il n'y a qu'une seule occurrence considérée, dite « occurrence statique », qui est l'instant $\mathrm{t}_n = 30$ secondes. Ainsi, pour n'importe quelle instruction « statique » $cs_{(a,a)}$ de l'ensemble $Cons_{AirLine}$ et pour n'importe quel défaut $F \in \Gamma_{AirLine} \setminus \{F_{Norm}\}$, l'ensemble d'occurrences est donné par :

$$\Omega_{(F,cs_{(a,a)})} = \{30\}$$

Occurrences pour les instructions dynamiques Pour les instructions dites « dynamiques », c'est-à-dire les instructions du type $cs_{(a,b)}$ avec $a, b \in C_Q$ tels que $a \neq b$, les « occurrences dynamiques » considérées sont l'ensemble des instants entiers compris dans la fenêtre temporelle $[25; 35]$. Ainsi, pour n'importe quelle instruction « dynamique » $cs_{(a,b)}$ de l'ensemble $Cons_{AirLine}$ et pour n'importe quel défaut $F \in \Gamma_{AirLine} \setminus \{F_{Norm}\}$, l'ensemble d'occurrences est donné par :

$$\Omega_{(F,cs_{(a,b)})} = \{25; 26; \ldots; 34; 35\}$$

Occurrences des défauts progressifs Bien entendu pour les défauts $F_{DirtCmpr}$, F_{DirtEV} et $F_{LeakAir}$, qui représentent des usures du système et qui sont donc faiblement progressifs, les ensembles d'occurrences sont identiques mais nous allons considérer les comportements de ces défauts comme étant constants.

4.5.2.3 Ensemble des comportements observables

La ligne d'air a été simulée, dans l'outil de simulation MATLAB/Simulink©, suivant chaque instruction de l'ensemble $cs_{(a,b)} \in Cons_{AirLine}$ et sous la présence de chacun des défauts $F \in \Gamma_{AirLine}$ à chacune des occurrences $\mathrm{t}_n \in \Omega_{(F,cs_{(a,b)})}$ (il n'y a bien sûr aucune occurrence à considérer pour le cas normal $F_{Norm} \in \Gamma_{AirLine}$). Ces simulations ont été ensuite enregistrées et ces enregistrements représentent donc les comportements observables de la ligne d'air. Nous avons en même temps enregistré chacun des signaux $sf_{\{\mathrm{t}_n\}}$ décrivant la validité temporelle de présence de n'importe quel défaut suivant les occurrences t_n. Rappelons que pour le cas normal F_{Norm} chacun de ces signaux est toujours nul (égal à 0) alors que pour une occurrence quelconque t_n de n'importe quel défaut $F \in \Gamma_{AirLine}$, le signal $sf_{\{\mathrm{t}_n\}}$ est donné par :

$$\begin{array}{rccl} sf_{\{\mathrm{t}_n\}} & : & [0;50] & \longrightarrow & \{0;1\} \\ & & \mathrm{t} & \longmapsto & \begin{cases} 0, \text{ si } \mathrm{t} \in [0;\mathrm{t}_n[\\ 1, \text{ si } \mathrm{t} \in [\mathrm{t}_n;50] \end{cases} \end{array}$$

Comportements observables normaux Pour le cas normal F_{Norm}, nous avons donc simulé cette ligne d'air sans défaut et pour chaque instruction $cs_{(a,b)} \in Cons_{AirLine}$. Nous avons ainsi obtenu chaque comportement observable normal $ObsB(cs_{(a,b)}, F_{Norm}, 0)$ en enregistrant ces simulations. En considérant la réunion de ces 930 comportements observables normaux pour toutes les instructions $cs_{(a,b)} \in Cons_{AirLine}$ nous avons obtenu l'ensemble des comportements observables normaux suivant l'ensemble $Cons_{AirLine}$ des instructions :

$$ObsBeh_{Cons_{AirLine}}(F_{Norm}) = \bigcup_{cs_{(a,b)} \in Cons_{AirLine}} \{ObsB(cs_{(a,b)}, F_{Norm}, 0)\}$$

La figure 4.13 suivante de la page 147 représente la simulation normale, sous la présence du cas normal F_{Norm}, de la ligne d'air suivant l'instruction $cs_{(25,14)}$. Les quatre premiers graphiques représentent les évolutions des variables observables de la ligne d'air alors que le cinquième graphique représente la validité temporelle de présence de n'importe quel défaut qui est nulle car il n'y a pas présence de défaut.

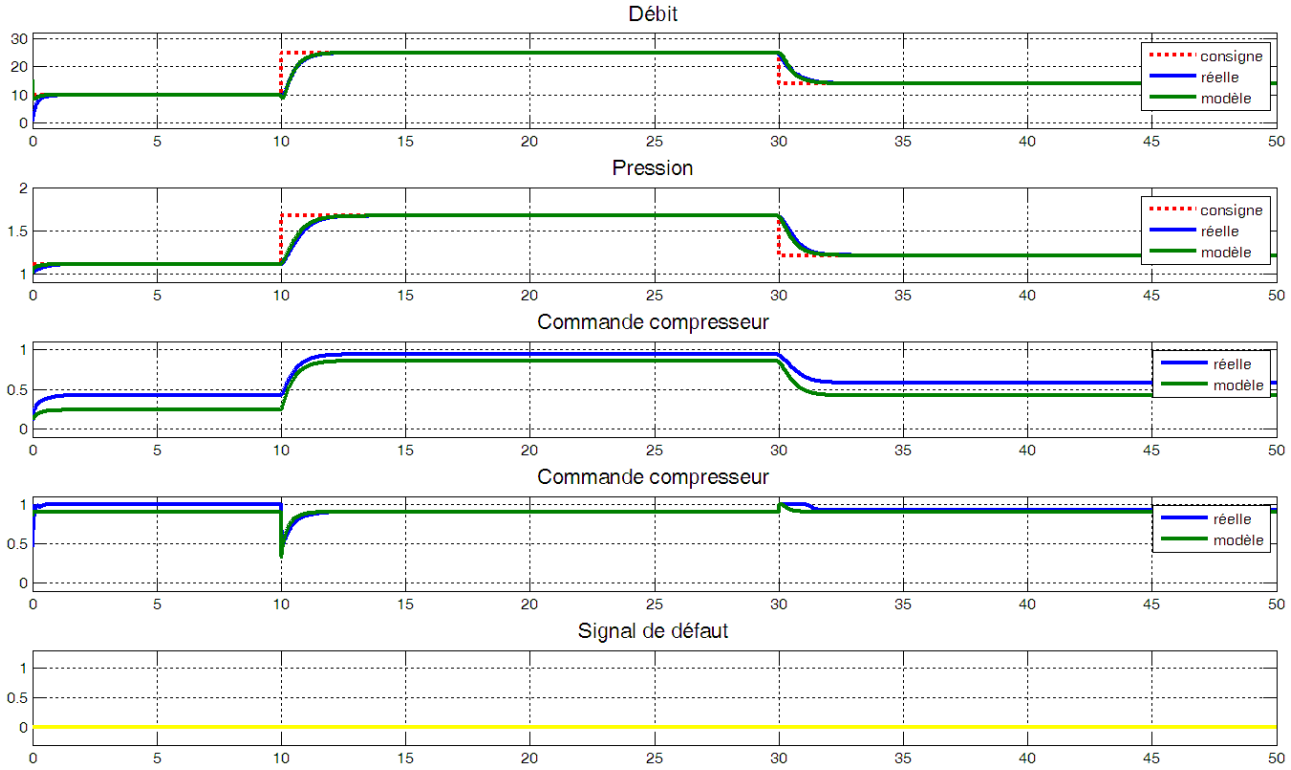

Figure 4.13 – Simulation normale de la ligne d'air suivant l'instruction $cs_{(25,14)}$.

Comportements observables sous la présence des défauts L'ensemble des défauts $\Gamma_{AirLine} \setminus \{F_{Norm}\}$ considéré pour la ligne d'air peut être partitionné en deux sous-ensembles : les défauts brusques ($F_{LockCmpr}$, F_{LockEV}, F_{SenQ} et F_{SenP}) et les défauts faiblement progressifs ($F_{DirtCmpr}$, F_{DirtEV} et $F_{LeakAir}$).

Pour les défauts brusques, les comportements de chacun de ces défauts sont identiques. De même pour les défauts faiblement progressifs où nous avons considéré un comportement constant durant toute la durée des simulations. Le tableau 4.3 suivant reprend ces comportements et les effets de tous ces défauts $F \in \Gamma_{AirLine}$ pour n'importe quelle instruction $cs_{(a,b)} \in Cons_{AirLine}$ et pour toute occurrence $t_n \in \Omega_{(F,cs_{(a,b)})}$.

Défaut	Comportement	Effet
$F_{LockCmpr}$	$dft_{(F_{LockCmpr},\mathrm{t}_n)}(\mathrm{t}) = \begin{cases} 0, & \text{si t} \in [0;\mathrm{t}_n[\\ 1, & \text{si t} \in [\mathrm{t}_n;50] \end{cases}$	$Q_{cmpr_{F_{LockCmpr}}}(\mathrm{t}) = Q_{cmpr}(\mathrm{t}) \cdot (1 - dft_{(F_{LockCmpr},\mathrm{t}_n)}(\mathrm{t}))$
F_{LockEV}	$dft_{(F_{LockEV},\mathrm{t}_n)}(\mathrm{t}) = \begin{cases} 0, & \text{si t} \in [0;\mathrm{t}_n[\\ 1, & \text{si t} \in [\mathrm{t}_n;50] \end{cases}$	$u_{x_{F_{LockEV}}}(\mathrm{t}) = \begin{cases} u_x(\mathrm{t}), & \text{si } dft_{(F_{LockEV},\mathrm{t}_n)}(\mathrm{t}) = 0 \\ u_x(\mathrm{t}_n), & \text{si } dft_{(F_{LockEV},\mathrm{t}_n)}(\mathrm{t}) = 1 \end{cases}$
F_{SenQ}	$dft_{(F_{SenQ},\mathrm{t}_n)}(\mathrm{t}) = \begin{cases} 0, & \text{si t} \in [0;\mathrm{t}_n[\\ 1, & \text{si t} \in [\mathrm{t}_n;50] \end{cases}$	$y_{Q_{F_{SenQ}}}(\mathrm{t}) = y_Q(\mathrm{t}) \cdot (1 - dft_{(F_{SenQ},\mathrm{t}_n)}(\mathrm{t}))$
F_{SenP}	$dft_{(F_{SenP},\mathrm{t}_n)}(\mathrm{t}) = \begin{cases} 0, & \text{si t} \in [0;\mathrm{t}_n[\\ 1, & \text{si t} \in [\mathrm{t}_n;50] \end{cases}$	$y_{P_{F_{SenP}}}(\mathrm{t}) = y_P(\mathrm{t}) \cdot (1 - dft_{(F_{SenP},\mathrm{t}_n)}(\mathrm{t}))$
$F_{DirtCmpr}$	$dft_{(F_{DirtCmpr},0)}(\mathrm{t}) = 1$	$u_{\omega_{F_{DirtCmpr}}}(\mathrm{t}) = u_\omega(\mathrm{t}) \cdot (0.75 \cdot dft_{(F_{DirtCmpr},0)}(\mathrm{t}))$
F_{DirtEV}	$dft_{(F_{DirtEV},0)}(\mathrm{t}) = 1$	$S_{EV_{F_{DirtEV}}}(\mathrm{t}) = S_{EV}(\mathrm{t}) \cdot (0.8 \cdot dft_{(F_{DirtEV},0)}(\mathrm{t}))$
$F_{LeakAir}$	$dft_{(F_{LeakAir},0)}(\mathrm{t}) = 1$	Défaut pour lequel nous n'avons pas décrit l'effet sous forme de perturbations de variables mais par ajout de composants.

Tableau 4.3 – Comportements et effets des défauts de la ligne d'air.

Pour chacun des défauts $F \in \Gamma_{AirLine} \setminus \{F_{Norm}\}$ de la ligne d'air, nous avons simulé cette ligne d'air suivant chacune des instructions $cs_{(a,b)} \in Cons_{AirLine}$ et sous la présence de chacun de ces défauts :

- Pour les défauts brusques $F \in \{F_{LockCmpr}; F_{LockEV}; F_{SenQ}; F_{SenP}\}$, nous l'avons simulée avec chacune des occurrences $\mathrm{t}_n \in \{25; 26; \ldots; 35\}$ pour les instructions dynamiques (i.e. : les instructions de la forme $cs_{(a,b)} \in Cons_{AirLine}$ avec $a \neq b$), et avec l'occurrence $\mathrm{t}_n = 30$ pour les instructions statiques (i.e. : les instructions de la forme $cs_{(a,b)} \in Cons_{AirLine}$ avec $a = b$).
- Pour les défauts faiblement progressifs $F \in \{F_{DirtCmpr}; F_{DirtEV}; F_{LeakAir}\}$ nous n'avons effectué qu'une seule simulation par instruction. Pour les instructions dynamiques et comme nous avons considéré un comportement constant du défaut durant toute la durée des simulations, cette simulation est donc équivalente pour toutes les occurrences $\mathrm{t}_n \in \{25; 26; \ldots; 35\}$; cependant et pour chacune des instructions dynamiques, tous ces comportements observables $ObsB(cs_{(a,b)}, F, \mathrm{t}_n)$ sont différents pour chacune des occurrences $\mathrm{t}_n \in \{25; 26; \ldots; 35\}$. Pour les instructions statiques, nous avons simulé ce modèle en considérant l'occurrence $\mathrm{t}_n = 30$.

En enregistrement ces simulations pour chacun des défauts $F \in \Gamma_{AirLine} \setminus \{F_{Norm}\}$, nous avons ainsi obtenu chacun des 10231 comportements observables $ObsB(cs_{(a,b)}, F, \mathrm{t}_n)$ sous la présence du défaut F. En considérant leur réunion, nous avons obtenu l'ensemble des comportements observables suivant l'ensemble $Cons_{AirLine}$ des instructions et sous la présence de F :

$$ObsBeh_{Cons_{AirLine}}(F) = \bigcup_{cs_{(a,b)} \in Cons_{AirLine}} \bigcup_{\mathrm{t}_n \in \Omega_{(F,cs_{(a,b)})}} \{ObsB(cs_{(a,b)}, F, \mathrm{t}_n)\}$$

La figure 4.14 de la page 149 représente la simulation de la ligne d'air suivant l'instruction $cs_{(25,14)}$ et sous la présence du défaut $F_{LockCmpr}$ de blocage du compresseur à l'occurrence $\mathrm{t}_n = 28$ secondes. Les quatre premiers graphiques représentent les évolutions des variables observables de la ligne d'air alors que le cinquième graphique représente la validité temporelle de présence de ce défaut $F_{LockCmpr}$ à l'occurrence $\mathrm{t}_n = 28$ secondes.

4.5.3 Étude de la diagnosticabilité de la ligne d'air

L'étude de la diagnosticabilité de la ligne d'air va être menée suivant les deux caractérisations de défauts introduites dans la partie théorique précédente : la caractérisation parfaite, qui nous permettra de nous assurer de manière intrinsèque de la diagnosticabilité des différents défauts, et la caractérisation par formules temporelles que nous souhaitons par la suite utiliser comme solution d'outil de diagnostic. Du fait que la caractérisation parfaite est beaucoup plus facile à construire, nous n'aurons

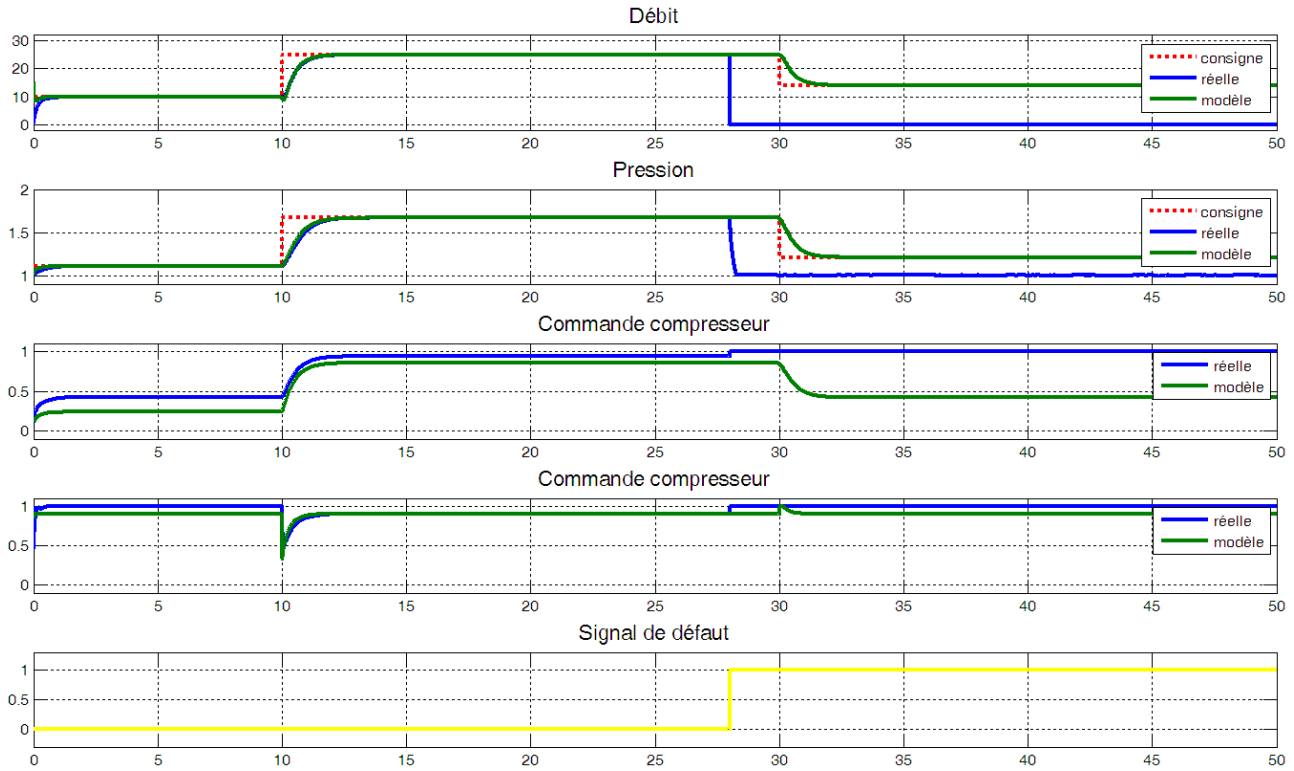

Figure 4.14 – Simulation de la ligne d'air suivant l'instruction $cs_{(25,14)}$ et sous la présence du blocage compresseur à l'occurrence $t_n = 28$.

pas la même approche de construction de ces deux caractérisations : c'est-à-dire que nous construirons complètement la caractérisation parfaite alors que pour la caractérisation par formules temporelles, nous ne construirons que les formules des défauts détectables. L'étude de la diagnosticabilité se fera par contre en suivant la même approche tirant partie du résultat du corollaire 4.1 : nous étudierons en premier lieu la détectabilité de tous les défauts répertoriés, puis nous étudierons ensuite la diagnosticabilité des défauts qui auront été conclus comme détectables. Comme nous l'avons expliqué, nous justifions ce choix de ne pas étudier la diagnosticabilité de tous les défauts car si un défaut n'est au préalable pas détectable, cela signifie que du point de vue de la caractérisation de défauts considérée, le comportement observable du système sous la présence de ce défaut est identique au comportement observable du système en fonctionnement normal. Le diagnostiqueur, basé sur cette caractérisation de défauts, ne détectera donc pas un comportement anormal du système, et ce comportement du système sous la présence de ce défaut observé par le diagnostiqueur sera donc analysé comme étant normal.

4.5.3.1 Étude suivant la caractérisation parfaite

La caractérisation parfaite consiste en une base de données de morceaux de comportements observables de longueur temporelle $b = 6$ secondes (la borne de diagnostic) sous la présence de n'importe quel défaut (y compris le cas normal). L'étude se réalise donc par comparaison entre eux de ces différents morceaux de comportements observables.

Construction de la caractérisation parfaite La base de données, définissant la caractérisation parfaite, va être rapidement construite car elle consiste juste à considérer les comportements observables, que nous venons de construire, restreints sur des fenêtres temporelles de longueur $b = 6$ secondes.

Pour le cas normal F_{Norm}, chaque comportement observable normal $ObsB(cs, F_{Norm}, 0)$, suivant n'importe quelle instruction $cs_{(a,b)} \in Cons_{AirLine}$, est restreint à chaque instant $\mathrm{t} \in [20; 50]$ du temps sur la fenêtre temporelle $[\mathrm{t}-6; \mathrm{t}]$:

$$\mathrm{Pr}_{[\mathrm{t}-6;\mathrm{t}]\times\overline{\mathrm{VD}}_{obs}}(ObsB(cs, F_{Norm}, 0))$$

L'ensemble $ObsBeh^{Bd(6)}_{Cons_{AirLine}}(F_{Norm})$ des comportements observables bornés normaux est donc défini par l'union, pour toutes les instructions $cs_{(a,b)} \in Cons_{AirLine}$ et tous les instants $\mathrm{t} \in [20; 50]$ du temps, de toutes les projections de $ObsB(cs, F_{Norm}, 0)$ sur tous les produits d'ensembles $[\mathrm{t}-6; \mathrm{t}] \times \overline{\mathrm{VD}}_{obs}$:

$$ObsBeh^{Bd(6)}_{Cons_{AirLine}}(F_{Norm}) = \bigcup_{cs_{(a,b)}\in Cons_{AirLine}} \Big(\bigcup_{\mathrm{t}\in[20;50]} \{\mathrm{Pr}_{[\mathrm{t}-6;\mathrm{t}]\times\overline{\mathrm{VD}}_{obs}}(ObsB(cs, F_{Norm}, 0))\}\Big)$$

Pour chacun des défauts $F \in \Gamma_{AirLine} \setminus \{F_{Norm}\}$, chaque comportement observable $ObsB(cs, F, \mathrm{t}_n)$, suivant n'importe quelle instruction $cs_{(a,b)} \in Cons_{AirLine}$ et sous la présence de ce défaut F à l'occurrence $\mathrm{t}_n \in \Omega_{(F,cs_{(a,b)})}$, est restreint à chaque instant $\mathrm{t} \in [\mathrm{t}_n; \mathrm{t}_n + 9]$ du temps (la valeur 9 est obtenue par $b = 6$ et $\delta = 3$) sur la fenêtre temporelle $[\mathrm{t}-6; \mathrm{t}]$:

$$\mathrm{Pr}_{[\mathrm{t}-6;\mathrm{t}]\times\overline{\mathrm{VD}}_{obs}}(ObsB(cs, F, \mathrm{t}_n))$$

L'ensemble $ObsBeh^{Bd(6)}_{Cons_{AirLine}}(F)$ des comportements observables bornés sous la présence de F est donc défini par l'union, pour toutes les instructions $cs_{(a,b)} \in Cons_{AirLine}$, pour toutes les occurrences $\mathrm{t}_n \in \Omega_{(F,cs_{(a,b)})}$ de F et tous les instants $\mathrm{t} \in [\mathrm{t}_n; \mathrm{t}_n + 9]$ du temps, de toutes les projections de $ObsB(cs_{(a,b)}, F, \mathrm{t}_n)$ sur tous les produits d'ensembles $[\mathrm{t}-6; \mathrm{t}] \times \overline{\mathrm{VD}}_{obs}$:

$$ObsBeh^{Bd(6)}_{Cons_{AirLine}}(F) = \bigcup_{cs_{(a,b)}\in Cons_{AirLine}} \Big(\bigcup_{\mathrm{t}_n\in\Omega_{(F,cs_{(a,b)})}} \Big(\bigcup_{\mathrm{t}\in[\mathrm{t}_n;\mathrm{t}_n+9]} \{\mathrm{Pr}_{[\mathrm{t}-6;\mathrm{t}]\times\overline{\mathrm{VD}}_{obs}}(ObsB(cs_{(a,b)}, F, \mathrm{t}_n))\}\Big)\Big)$$

Diagnosticabilité de la ligne d'air suivant la caractérisation parfaite Comme nous l'avons indiqué, nous avons mené cette étude en recherchant en premier lieu les défauts détectables, puis en étudiant la diagnosticabilité des défauts détectables. Rappelons que l'étude d'éligibilité est acquise, comme nous l'avons remarqué dans la partie présentant cette caractérisation parfaite. Nous ne nous sommes pas intéressé, pour cette caractérisation parfaite, à l'étude de l'isolabilité des défauts ; nous sommes directement passé de l'étude de la détectabilité à l'étude de la diagnosticabilité. Le tableau 4.4 ci-dessous reprend les résultats obtenus lors de cette étude.

Défaut	Détectabilité	Diagnosticabilité
$F_{LockCmpr}$	Détectable	Diagnosticable
F_{LockEV}	Non détectable	⁄
F_{SenQ}	Détectable	Diagnosticable
F_{SenP}	Détectable	Diagnosticable
$F_{DirtCmpr}$	Non détectable	⁄
F_{DirtEV}	Non détectable	⁄
$F_{LeakAir}$	Non détectable	⁄

Tableau 4.4 – Étude de la diagnosticabilité de la ligne d'air suivant la caractérisation parfaite.

Pour la détectabilité de chaque défaut $F \in \Gamma_{AirLine} \setminus \{F_{Norm}\}$, il a fallu comparer chaque comportement observable $ObsB(cs_{(a,b)}, F, \mathrm{t}_n)$, suivant une instruction quelconque $cs_{(a,b)} \in Cons_{AirLine}$ et sous

la présence de F à une occurrence $\mathrm{t}_n \in \Omega_{(F,cs_{(a,b)})}$, à la base de données $ObsBeh^{Bd(6)}_{Cons_{AirLine}}(F_{Norm})$ des comportements observables bornés normaux. C'est-à-dire en reprenant la définition de la détectabilité qu'à partir de l'occurrence t_n du défaut F (valant 30 secondes lorsque $a = b$ ou un instant entre 25 et 35 secondes lorsque $a \neq b$), il faut vérifier qu'il existe un instant t_k dans la fenêtre temporelle $[\mathrm{t}_n; \mathrm{t}_n+4]$ (où $\mathrm{t}_n + 4$ est obtenu par $\mathrm{t}_n + (b - h)$ avec $b = 6$ et $h = 2$) tel que :

- d'une part, chaque morceau de $ObsB(cs_{(a,b)}, F, \mathrm{t}_n)$, restreint sur les fenêtres temporelles $[\mathrm{t} - 6; \mathrm{t}]$ pour chaque instant $\mathrm{t} < \mathrm{t}_k$, correspond à un élément de $ObsBeh^{Bd(6)}_{Cons_{AirLine}}(F_{Norm})$;
- d'autre part que le morceau de $ObsB(cs_{(a,b)}, F, \mathrm{t}_n)$, restreint sur la fenêtre temporelle $[\mathrm{t}_k - 6; \mathrm{t}_k]$, ne correspond à aucun élément de $ObsBeh^{Bd(6)}_{Cons_{AirLine}}(F_{Norm})$.

Seuls les défauts $F_{LockCmpr}$ de blocage du compresseur ainsi que F_{SenQ} et F_{SenP} de mesure des capteurs de débit et de pression sont détectables. Nous obtenons donc l'ensemble $\Gamma^{Detect}_{AirLine} = \{F_{LockCmpr}; F_{SenQ}; F_{SenP}\}$ de défauts détectables suivant cette caractérisation parfaite. Le défaut F_{LockEV} de blocage de l'électrovanne n'est pas détectable notamment pour des blocages en fonctionnement statique, c'est-à-dire les cas où l'occurrence du défaut est dans une période de fonctionnement statique. Comme aucun changement de la consigne n'est demandé, le système est donc stationnaire et la commande u_x de l'électrovanne n'est pas modifiée. Il n'est donc pas possible de déterminer qu'elle est bloquée. Enfin, concernant les défauts $F_{DirtCmpr}$ et F_{DirtEV} d'encrassement du compresseur et de l'électrovanne ainsi que le défaut $F_{LeakAir}$ de fuite d'air, ils ne sont pas détectables suivant la remarque que nous avons indiquée précédemment : même avant l'occurrence du défaut, les morceaux de comportements observables étudiés (sur les fenêtre temporelles de longueur 6 secondes) ne sont pas identiques à des morceaux de comportements observables normaux. Il n'y a donc pas chute de la propriété normale $\mathrm{P}_{F_{Norm}}$.

Tous ces résultats de détectabilité, sauf pour le blocage de l'électrovanne, sont du type tout ou rien : soit le défaut est détectable pour les 10261 comportements observables sous sa présence, soit il n'est pas détectable pour aucun de ces 10261 comportements observables. Concernant le blocage de l'électrovanne et comme nous venons de le dire, que l'instruction soit statique (i.e. : du type $cs_{(a,b)} \in Cons_{AirLine}$ avec $a = b$) ou dynamique (i.e. : du type $cs_{(a,b)} \in Cons_{AirLine}$ avec $a \neq b$) mais que l'occurrence t_n du défaut se situe après la dynamique de réponse du système (i.e. : $\mathrm{t}_n \in \{43; 44; 45\}$), ce blocage n'est pas détectable car aucun changement de consigne n'est demandé : le système est stationnaire.

Suite à cette étude de la détectabilité, nous nous sommes donc attelé à l'étude de la diagnosticabilité des trois défauts détectables : le défaut $F_{LockCmpr}$ de blocage du compresseur ainsi que les défauts F_{SenQ} et F_{SenP} de mesure des capteurs de débit et de pression.

Pour la diagnosticabilité de chaque défaut détectable $F \in \{F_{LockCmpr}; F_{SenQ}; F_{SenP}\}$, il a fallu comparer chaque comportement observable $ObsB(cs_{(a,b)}, F, \mathrm{t}_n)$, suivant une instruction quelconque $cs_{(a,b)} \in Cons_{AirLine}$ et sous la présence de F à une occurrence $\mathrm{t}_n \in \Omega_{(F,cs_{(a,b)})}$, à la base de données $ObsBeh^{Bd(6)}_{Cons_{AirLine}}(F_{Norm})$ des comportements observables bornés normaux ainsi qu'aux bases de données $ObsBeh^{Bd(6)}_{Cons_{AirLine}}(F')$ des comportements observables bornés sous la présence des autres défauts détectables $F' \neq F$. En reprenant la définition de la diagnosticabilité, cela signifie qu'à partir de l'occurrence t_n du défaut F, il faut vérifier qu'il existe un instant t_k dans la fenêtre temporelle $[\mathrm{t}_n; \mathrm{t}_n + 4]$ tel que :

- non seulement chaque morceau de $ObsB(cs_{(a,b)}, F, \mathrm{t}_n)$, restreint sur les fenêtres temporelles $[\mathrm{t} - 6; \mathrm{t}]$ pour chaque instant $\mathrm{t} < \mathrm{t}_k$, correspond à un élément de $ObsBeh^{Bd(6)}_{Cons_{AirLine}}(F_{Norm})$,
- mais aussi que le morceau de $ObsB(cs_{(a,b)}, F, \mathrm{t}_n)$, restreint sur la fenêtre temporelle $[\mathrm{t}_k - 6; \mathrm{t}_k]$, ne correspond à aucun élément de $ObsBeh^{Bd(6)}_{Cons_{AirLine}}(F_{Norm})$,
- et enfin que chaque morceau de $ObsB(cs_{(a,b)}, F, \mathrm{t}_n)$, restreint sur les fenêtres temporelles $[\mathrm{t} - 6; \mathrm{t}]$ pour chaque instant $\mathrm{t} \in [\mathrm{t}_k + 2; \mathrm{t}_k + 5]$ (intervalle obtenu par $\mathrm{t}_k + h$ avec $h = 2$ et $\mathrm{t}_k + (h + \delta)$

avec $h = 2$ et $\delta = 3$), correspond à un élément de $ObsBeh^{Bd(6)}_{Cons_{AirLine}}(F)$ et à aucun élément de $ObsBeh^{Bd(6)}_{Cons_{AirLine}}(F')$ pour les autres défauts détectables $F' \neq F$.
Ces trois défauts sont diagnosticables, tous du type tout ou rien.

4.5.3.2 Étude suivant la caractérisation par formules temporelles

À l'inverse de l'étude suivant la caractérisation parfaite, où nous avons complètement construit cette caractérisation pour tous les défauts répertoriés, nous ne construirons, pour la caractérisation par formules temporelles, que les formules temporelles des défauts détectables. Nous construirons donc en premier lieu la formule temporelle normale qui nous permettra d'étudier la détectabilité des différents défauts, puis nous construirons les formules temporelles des défauts détectables afin d'en étudier leur diagnosticabilité. La formule normale va être obtenue suivant la méthode générique présentée, alors que les autres formules des défauts détectables seront, quant à elles, déterminées au cas par cas en analysant les différents comportements observables sous la présence des défauts considérés.

Construction de la formule temporelle normale Comme nous l'avons indiqué dans la partie théorique, la formule normale φ_{Norm} va prendre en compte des seuils adaptatifs de comparaison non seulement entre la consigne c et la mesure réelle y, mais aussi entre les mesures réelle y et modèle y^M, ainsi qu'entre les commandes réelle u et modèle u^M. Ces seuils adaptatifs vont être déterminés afin de rendre compte de l'évolution de la consigne c_Q de débit : c'est-à-dire où elle se trouve dans l'ensemble C_Q des valeurs ainsi que ses changements entre les différentes valeurs de C_Q.

En considérant où se trouve la consigne de débit c_Q dans le domaine C_Q des valeurs, nous obtenons les différents points de fonctionnement du système et du modèle. Cela nous fournit donc un partitionnement du domaine de la consigne $C_Q = \dot{\bigcup}_i C^i_Q$. Pour obtenir ce partitionnement, nous avons fait évoluer la consigne c_Q dans son domaine de valeurs C_Q puis, selon où se trouve c_Q, nous avons analysé les différences $|c_Q - y_Q|$ et $|c_P - y_P|$ entre les consignes et les mesures réelles, ainsi que les différences $|y_Q - y^M_Q|$, $|y_P - y^M_P|$ entre les mesures réelle et les mesures modèle. Le but étant de trouver les zones dans C_Q où ces différences sont similaires : c'est-à-dire des parties disjointes C_i de C_Q où quelle que soit la valeur de la consigne c_Q, les différences sont proches.

La figure 4.15 de la page 153 représente les nuages de points des différences $|c_Q - y_Q|$, $|c_P - y_P|$, $|y_Q - y^M_Q|$ et $|y_P - y^M_P|$ en fonction de la valeur de la variable c_Q dans son domaine C_Q, représentée par l'axe des abscisses de chacun des quatre graphiques. Les différents traits verticaux rouges représentent, dans chacun des graphiques, les coupures entre les partitionnements de C_Q suivant les différences considérées. Le premier graphique représente le nuage de points de la différence $|c_Q - y_Q|$ pour lequel nous observons le partitionnement $C_Q = [0; 2]\dot{\cup}[3; 27]\dot{\cup}[28; 30]$. Sur le deuxième graphique, représentant le nuage de points de la différence $|c_P - y_P|$, le partitionnement est $C_Q = [0; 27]\dot{\cup}[28; 30]$. Le troisième graphique représente le nuage de points de la différence $|y_Q - y^M_Q|$ dont le partitionnement est $C_Q = [0; 7]\dot{\cup}[8; 27]\dot{\cup}[28; 30]$. Enfin le quatrième graphique représente le nuage de points de la différence $|y_P - y^M_P|$ pour lequel nous observons le partitionnement $C_Q = [0; 6]\dot{\cup}[7; 27]\dot{\cup}[28; 30]$. En « fusionnant » ces différents partitionnements, nous avons obtenu les parties suivantes :

$$\begin{array}{rcl} C^1_Q & = & [0; 2] \\ C^2_Q & = & [3; 7] \\ C^3_Q & = & [8; 27] \\ C^4_Q & = & [28; 30] \end{array}$$

Nous n'avons pas utilisé les différences $|u_\omega - u^M_\omega|$ et $|u_x - u^M_x|$, entre les commandes réelles et modèles, pour établir ce partitionnement de C_Q. Du fait que les points de fonctionnement entre les deux modèles M_{parf} et M_{emb} sont différents, ces deux différences risquent donc d'être trop importantes

Figure 4.15 – Nuages de points des différences $|c_Q - y_Q|$, $|c_P - y_P|$, $|y_Q - y_Q^M|$ et $|y_P - y_P^M|$ en fonction de la valeur de la variable c_Q.

et/ou hachées suivant l'évolution de la variable c_Q dans son domaine C_Q. Leur utilisation aurait conduit à un partitionnement beaucoup trop fin ce qui aurait donc augmenté la complexité de la formule normale φ_{Norm}.

En considérant les changements de la consigne de débit c_Q entre les différentes valeurs de C_Q, nous obtenons les fonctionnements statiques et dynamiques à l'intérieur des différentes parties C_Q^i de C_Q pour $i \in \{1; 2; 3; 4\}$. C'est-à-dire que lorsque la consigne vient d'évoluer depuis moins de 3

secondes, correspondant à la dynamique de réponse du système, le système est dit en « fonctionnement dynamique » alors que si elle n'a pas évolué depuis au moins 3 secondes, le système est dit en « fonctionnement statique ». La formule normale φ_{Norm} est donc de la forme :

$$\varphi_{Norm} : \bigwedge_{i\in\{1;...;4\}} ([(c_Q \in C_Q^i \wedge stat_{c_Q}) \Rightarrow Stat_{C_Q^i}] \wedge [(c_Q \in C_Q^i \wedge dyn_{c_Q}) \Rightarrow Dyn_{C_Q^i}])$$

avec $stat_{c_Q}$ exprimant l'évolution statique de la variable de consigne c_Q et dyn_{c_Q} exprimant son évolution dynamique. Comme nous ne considérons que des instructions constantes par morceaux, ces sous-formules $stat_{c_Q}$ et dyn_{c_Q} sont donc de la forme :

$$\begin{array}{rcl} stat_{c_Q} & : & \mathrm{G}_{[-3,0]}(|c_Q - \mathrm{V}_{[-0.01]}c_Q| = 0) \\ dyn_{c_Q} & : & \mathrm{E}_{[-3,0]}(|c_Q - \mathrm{V}_{[-0.01]}c_Q| \neq 0) \end{array}$$

avec $\mathrm{G}_{[-3,0]}(|c-\mathrm{V}_{[-0.01]}c| = 0)$ exprimant le fait que la consigne n'a pas évolué pour tout instant depuis 3 secondes, et $\mathrm{E}_{[-3,0]}(|c - \mathrm{V}_{[-0.01]}c| \neq 0)$ exprimant le fait qu'il existe un instant depuis 3 secondes où la consigne a évolué.

Pour chacun des $i \in \{1; \ldots; 4\}$, exprimant les différentes parties du domaine C_Q de la consigne c_Q, la sous-formule $Stat_{C_Q^i}$ vérifie les seuils adaptatifs statiques supérieurs $\overline{\varepsilon}^{(stat,i)}$ et inférieurs $\underline{\varepsilon}^{(stat,i)}$, alors que la sous-formule $Dyn_{C_Q^i}$ vérifie les seuils adaptatifs dynamiques supérieurs $\overline{\varepsilon}^{(dyn,i)}$ et inférieurs $\underline{\varepsilon}^{(dyn,i)}$. Ces sous-formules $Stat_{C_Q^i}$ et $Dyn_{C_Q^i}$ sont alors de la forme :

$$\begin{array}{rcl} Stat_{C_Q^i} & : & [\\ & & \quad (\underline{\varepsilon}^{(stat,i)}_{(c_Q,y_Q)} \leq |c_Q - y_Q| \leq \overline{\varepsilon}^{(stat,i)}_{(c_Q,y_Q)}) \wedge (\underline{\varepsilon}^{(stat,i)}_{(c_P,y_P)} \leq |c_P - y_P| \leq \overline{\varepsilon}^{(stat,i)}_{(c_P,y_P)}) \\ & & \wedge (\underline{\varepsilon}^{(stat,i)}_{(y_Q,y_Q^M)} \leq |y_Q - y_Q^M| \leq \overline{\varepsilon}^{(stat,i)}_{(y_Q,y_Q^M)}) \wedge (\underline{\varepsilon}^{(stat,i)}_{(y_P,y_P^M)} \leq |y_P - y_P^M| \leq \overline{\varepsilon}^{(stat,i)}_{(y_P,y_P^M)}) \\ & & \wedge (\underline{\varepsilon}^{(stat,i)}_{(u_\omega,u_\omega^M)} \leq |u_\omega - u_\omega^M| \leq \overline{\varepsilon}^{(stat,i)}_{(u_\omega,u_\omega^M)}) \wedge (\underline{\varepsilon}^{(stat,i)}_{(u_x,u_x^M)} \leq |u_x - u_x^M| \leq \overline{\varepsilon}^{(stat,i)}_{(u_x,u_x^M)}) \\ & &] \end{array}$$

$$Dyn_{C_Q^i} : [(\underline{\varepsilon}^{(dyn,i)}_{(y_Q,y_Q^M)} \leq |y_Q - y_Q^M| \leq \overline{\varepsilon}^{(dyn,i)}_{(y_Q,y_Q^M)}) \wedge (\underline{\varepsilon}^{(dyn,i)}_{(y_P,y_P^M)} \leq |y_P - y_P^M| \leq \overline{\varepsilon}^{(dyn,i)}_{(y_P,y_P^M)})]$$

L'apprentissage de ces différents seuils adaptatifs inférieurs $\underline{\varepsilon}$ et supérieurs $\overline{\varepsilon}$ a été accompli par analyse de tous les comportements observables normaux de l'ensemble $ObsBeh_{Cons_{AirLine}}(F_{Norm})$. Pour chaque comportement observable normal $ObsB(cs_{(a,b)}, F_{Norm}, 0) \in ObsBeh_{Cons_{AirLine}}(F_{Norm})$, suivant une instruction $cs_{(a,b)} \in Cons_{AirLine}$, nous avons déterminé les seuils minimum et maximum de chacune des comparaisons $|c_Q - y_Q|$, $|c_P - y_P|$, $|u_\omega - u_\omega^M|$ et $|u_x - u_x^M|$ pour le fonctionnement statique, ainsi que de chacune des comparaisons $|y_Q - y_Q^M|$ et $|y_Q - y_Q^M|$ pour les fonctionnements statique et dynamique. Chacun de ces 961 comportements observables normaux nous a donc fourni ses seuils minimum et maximum. En prenant le minimum des seuils minimum et le maximum des seuils maximum, nous avons donc obtenu les différents seuils.

Utilisation de l'outil ARTiMon© pour la vérification de formules temporelles Afin d'étudier la diagnosticabilité des défauts, mais aussi leur éligibilité, détectabilité et isolabilité, nous allons avoir besoin de vérifier la validité de formules temporelles : soit directement les formules des défauts, soit des formules temporelles combinant ces formules de défauts. Cela signifie que nous allons vérifier que les comportements observables sous la présence des défauts satisfont, ou non, des formules temporelles.

Pour réaliser ces vérifications, nous avons utilisé l'outil ARTiMon© du CEA/List ([Rap08]). Il s'agit d'un outil de surveillance qui permet donc de surveiller un processus en vue de détecter la violation de spécifications portant sur celui-ci ; cette surveillance se faisant sur un flux de données du processus

arrivant en continu ou échantillonné et à une période fixe ou non. Une spécification est, quant à elle, exprimée sous la forme d'une formule temporelle élaborée grâce aux opérateurs que nous avons présentés dans la partie théorique précédente. L'outil peut être interfacé à MATLAB/Simulink© afin d'analyser un flux de données lors d'une simulation d'un modèle, c'est la manière dont nous l'avons utilisé. Le flux de données du processus est un comportement observable sous la présence d'un défaut $F \in \Gamma_{AirLine}$, et les spécifications sont les formules de défauts ou des combinaisons de formules de défauts.

Nous avons donc considéré tous les comportements observables $Obs \in ObsBeh_{Cons_{AirLine}}$, que nous avions enregistrés sous forme de données simulables dans MATLAB/Simulink© (i.e. : des fichiers de données du type .mat de MATLAB©). Avec l'outil ARTiMon©, nous avons donc vérifié que ces comportements observables satisfont, ou non, les formules temporelles des défauts ou des combinaisons temporelles des formules de défauts.

Éligibilité du cas normal Nous avons étudié l'éligibilité du cas normal F_{Norm} en utilisant pour cela la formule *ad-hoc* $\Psi^e_{F_{Norm}}$ suivante :

$$\Psi^e_{F_{Norm}} \ := \ \varphi_{Norm}$$

Il a donc fallu vérifier que chaque comportement observable normal $ObsB(cs_{(a,b)}, F_{Norm}, 0) \in ObsBeh_{Cons_{AirLine}}(F_{Norm})$ satisfait cette formule normale $\Psi^e_{F_{Norm}}$. En ayant fixé la durée t_0 d'initialisation à l'instant 20 secondes, il a fallu vérifier que pour toute instruction $cs_{(a,b)} \in Cons_{AirLine}$ et pour tout instant $\mathrm{t} \in [20; 50]$, nous avons bien $(ObsB(cs_{(a,b)}, F_{Norm}, 0), \mathrm{t}) \models \Psi^e_{F_{Norm}}$.

Pour qu'un comportement observable normal $ObsB(cs_{(a,b)}, F_{Norm}, 0)$ satisfasse la formule normale φ_{Norm} à un instant $\mathrm{t} \in [20; 50]$ du temps, il faut d'abord que la fenêtre temporelle $[\mathrm{t} - 3; \mathrm{t}]$ soit incluse dans le domaine $[20; 50]$, ce qui est clairement établi. Ensuite et à un instant quelconque $\mathrm{t} \in [20; 50]$ du temps, la variable c_Q se trouve obligatoirement dans une unique partie C^i_Q, et soit elle n'a pas évolué depuis 3 secondes, soit elle vient d'évoluer depuis moins de 3 secondes. Ce qui signifie donc de vérifier la satisfaction de la sous-formule statique $Stat_{C^i_Q}$ dans le premier cas et la satisfaction de la sous-formule dynamique $Dyn_{C^i_Q}$ dans le second.

Chacun des 961 comportements observables normaux $ObsB(cs_{(a,b)}, F_{Norm}, 0)$, de l'ensemble $ObsBeh_{Cons_{AirLine}}(F_{Norm})$, a satisfait cette formule d'éligibilité $\Psi^e_{F_{Norm}}$, ce qui rend donc ce cas normal F_{Norm} éligible. Ce résultat d'éligibilité du cas normal F_{Norm} n'est pas surprenant car l'apprentissage de cette formule normale φ_{Norm} s'est réalisé en utilisant tous ces comportements observables normaux. Il est par conséquent normal qu'ils la satisfassent. Le contraire aurait par contre été « surprenant ».

Étude de la détectabilité des défauts Comme pour l'étude suivant la caractérisation parfaite, nous avons d'abord étudié la détectabilité de tous les défauts $F \in \Gamma_{AirLine}$ afin de n'étudier, par la suite, que la diagnosticabilité des défauts détectables. Cela est d'autant plus justifié avec cette caractérisation par formules temporelles que la construction des formules temporelles des défauts n'est pas simple. En effet, bien que nous ayons proposé une méthode permettant d'obtenir automatiquement la formule normale φ_{Norm}, nous n'avons pas trouvé de méthode générique permettant d'obtenir les formules des défauts. Il faut faire une analyse au cas par cas pour chacun des défauts.

Pour mener cette étude, nous avons utilisé la formule temporelle *ad-hoc* Ψ^d de détectabilité pour chacun des défauts $F \in \Gamma_{AirLine}$. Rappelons que cette formule s'écrit :

$$\Psi^d \ : \ \mathrm{Top}(sf_{\{\mathrm{t}_n\}}) \Rightarrow [\varphi_{Norm} \mathrm{U}_{[0;4]} (\mathrm{Bot}(\varphi_{Norm}))]$$

et signifie que lorsque le signal $sf_{\{\mathrm{t}_n\}}$ d'occurrence du défaut $F \in \Gamma_{AirLine}$ considéré passe à vrai (i.e. : à chaque occurrence $\mathrm{t}_n \in \Omega_{(F, cs_{(a,b)})}$ de F suivant l'instruction $cs_{(a,b)}$) alors dans un délai maximum de

4 secondes (obtenu par $b - h$ avec $b = 6$ et $h = 2$), la formule normale φ_{Norm} doit passer de satisfaite à non-satisfaite : c'est-à-dire que sa validité doit passer de vraie à fausse dans un délai maximum de 4 secondes et qu'avant ce passage, elle doit être vraie.

En vérifiant, pour chacun des défauts $F \in \Gamma_{AirLine}$, que chacun des comportements observables $ObsB(cs_{(a,b)}, F, \mathrm{t}_n)$ sous la présence de ce défaut satisfait cette formule Ψ^d, nous avons ainsi vérifié sa détectabilité. Or lorsque cette formule n'est pas satisfaite, il est intéressant de savoir pourquoi :

- est-ce l'intervalle de détection qui n'est pas assez long ?
- ou est-ce la validité de la formule normale φ_{Norm} qui est en cause ? est-elle toujours ou jamais valide ?

Nous avons donc en même temps vérifié la formule normale φ_{Norm} afin de connaître sa validité dans le domaine temporel $[20; 50]$.

Le tableau 4.5 de la page 156 indique les défauts détectables suivant cette caractérisation par formules temporelles. Nous avons mené cette étude de détectabilité pour tous les défauts, malgré le fait que nous aurions pu nous limiter aux défauts conclus détectables par l'étude suivant la caractérisation parfaite. Nous avons souhaité déterminer la robustesse de cette formule normale.

Défaut	Détectabilité
$F_{LockCmpr}$	Détectable
F_{LockEV}	Non détectable
F_{SenQ}	Détectable
F_{SenP}	Détectable
$F_{DirtCmpr}$	Non détectable
F_{DirtEV}	Non détectable
$F_{LeakAir}$	Non détectable

Tableau 4.5 – Étude de la détectabilité des défauts de la ligne d'air suivant la caractérisation par formules temporelles.

Les défauts faiblement progressifs $F_{DirtCmpr}$ et F_{DirtEV} d'encrassement du compresseur et de l'électrovanne ainsi que $F_{LeakAir}$ de fuite d'air ne sont pas détectables. Or nous n'obtenons pas cette non-détectabilité de manière complète comme pour la caractérisation parfaite : c'est-à-dire que certains comportements observables sous la présence de ces défauts sont détectables. Il s'agit des comportements pour lesquels la consigne évolue dans ou entre (i.e. : de l'une vers l'autre) les parties C_Q^1, C_Q^2 et C_Q^4 ainsi que pour des occurrences $\mathrm{t}_n \in \{30; 31; \ldots; 35\}$. Ces résultats s'expliquent d'une part par le fait que les seuils sont assez larges dans ces trois parties, et d'autre part que ces occurrences se situent durant la dynamique de réponse du système, où seules les différences entre les mesures réelles et celles du modèle sont évaluées avec des seuils ici aussi beaucoup plus larges. Pour les comportements observables où le défaut n'est pas détectable, il s'agit du cas que nous avons remarqué et qui est vérifié avec la caractérisation parfaite : même avant l'occurrence du défaut, la formule normale φ_{Norm} n'est pas satisfaite par les comportements observables sous la présence de ces défauts, ce que nous avons observé en analysant la validité de cette formule normale φ_{Norm}.

Par ailleurs et comme pour la caractérisation parfaite, le défaut F_{LockEV} de blocage de l'électrovanne n'est pas détectable pour des blocages en fonctionnement statique : lorsque la consigne est du type $cs_{(a,b)}$ avec $a = b$ ou lorsqu'elle est du type $cs_{(a,b)}$ avec $a \neq b$ et avec des occurrences $\mathrm{t}_n \in \{33; 34; 35\}$.

Enfin les défauts $F_{LockCmpr}$ de blocage du compresseur ainsi que F_{SenQ} et F_{SenP} de mesures des capteurs de débit et de pression sont, comme avec la caractérisation parfaite, détectables. L'ensemble des défauts détectables suivant cette caractérisation par formules temporelles est donc $\Gamma_{AirLine}^{Detect} = \{F_{LockCmpr}; F_{SenQ}; F_{SenP}\}$.

Construction des formules temporelles des défauts Concernant les défauts, nous n'avons pas trouvé de méthode générique et automatique permettant d'obtenir des formules temporelles les décrivant. Pour chaque défaut détectable $F \in \Gamma^{Detect}_{AirLine}$, il a par conséquent fallu non seulement analyser le comportement et l'effet du défaut sur le système, mais en plus analyser les 10261 différents comportements observables sous la présence du défaut considéré afin de générer la formule temporelle φ_F adéquate. Remarquons que cette phase de construction doit être réalisée par des experts du système. Nous concernant et comme nous n'avions que peu de défauts détectables à étudier, nous avons considéré des formules relativement basiques, avec néanmoins de bons résultats d'étude de la diagnosticabilité. Pour une étude ayant beaucoup plus de défauts détectables à étudier, il faudrait sûrement définir des formules plus complexes.

Pour le défaut $F_{LockCmpr}$ de blocage du compresseur, nous avons vu que ce défaut provoque une chute brutale à nulle du débit d'air en sortie du compresseur (i.e. : une chute à 0 gramme par seconde). Ce blocage du compresseur va donc mettre la mesure y_Q de débit à nulle, donc égale à 0 gramme par seconde. La mesure y_P de pression va elle aussi être à nulle (i.e. : donc égale à 1 bar) et la commande u_ω de régime du compresseur va par ailleurs être mise au maximum (i.e. : à la valeur 1) afin de compenser cette perte de débit. En analysant par la suite les comportements observables sous la présence de ce défaut $F_{LockCmpr}$ de blocage du compresseur, les éléments de $ObsBeh_{Cons_{AirLine}}(F_{LockCmpr})$, nous avons repéré d'une part que la commande u_x d'ouverture de l'électrovanne est toujours au maximum (i.e. : à la valeur 1), et de plus que lorsque la consigne c_Q demandée par l'opérateur est nulle (i.e. : égale à 0 gramme par seconde), alors les commandes u_ω de régime du compresseur et u_x d'ouverture de l'électrovanne ne sont pas mises au maximum (i.e. : à la valeur 1). La formule $\varphi_{LockCmpr}$ de ce défaut $F_{LockCmpr}$ de blocage du compresseur est donc donnée par :

$$\varphi_{LockCmpr} \quad : \quad \begin{array}{l} [(c_Q = 0) \wedge (y_Q = 0) \wedge (y_P = 1)] \\ \vee \\ [(c_Q \neq 0) \wedge (y_Q = 0) \wedge (y_P = 1) \wedge (u_\omega = 1) \wedge (u_x = 1)] \end{array}$$

Pour le défaut F_{senQ} de mesure du capteur de débit, nous avons vu que ce défaut entraîne une brusque chute de la mesure du débit à nulle (i.e. : 0 gramme par seconde). Ainsi ce défaut va donc mettre la mesure y_Q du débit à nulle et la commande u_ω de régime du compresseur au maximum (i.e. : à la valeur 1) afin de compenser cette perte de débit ce qui impactera la mesure y_P de pression qui va être elle aussi au maximum (i.e. : comprise entre 1,77 et 2 bar, valeurs obtenues par simulation). Nous aurions pu ne considérer que la valeur minimale 1,77 bar de la mesure y_P de pression, mais le fait qu'elle doit être inférieure à 2 bar a été rajouté car il se pourrait que d'autres défauts non pris en compte dans cette étude rendent cette mesure supérieure à 2 bar. Par la suite en analysant les comportements observables sous la présence de ce défaut F_{senQ} de mesure du capteur de débit, les éléments de $ObsBeh_{Cons_{AirLine}}(F_{senQ})$, nous avons repéré que la commande u_x d'ouverture de l'électrovanne est toujours au maximum (i.e. : à la valeur 1). La formule φ_{senQ} de ce défaut F_{senQ} de mesure du capteur de débit est donc donnée par :

$$\varphi_{senQ} \quad : \quad [(y_Q = 0) \wedge (1{,}77 \leq y_P \leq 2) \wedge (u_\omega = 1) \wedge (u_x = 1)]$$

Pour le défaut F_{senP} de mesure du capteur de pression, nous avons vu que ce défaut entraîne une brusque chute de la mesure de pression à nulle (i.e. : à 0 bar). Ainsi ce défaut va donc mettre la variable observable y_P de mesure de pression à nulle, donc égale à 0 bar, et nous ne présumons au préalable rien sur les effets appliqués aux autres variables observables. Par la suite en analysant les comportements observables sous la présence de ce défaut F_{senP} de mesure du capteur de pression, les éléments de $ObsBeh_{Cons_{AirLine}}(F_{senP})$, nous n'avons uniquement repéré que la mesure y_Q du débit

n'est jamais à nulle (i.e. : à 0 gramme par seconde) ; aucune information représentative n'a pu être donnée concernant les variables de commandes. La formule φ_{senP} de ce défaut F_{senQ} de mesure du capteur de débit est donc donnée par :

$$\varphi_{senP} \quad : \quad [(y_P = 0) \wedge (y_Q \neq 0)]$$

Étude de la diagnosticabilité de la ligne d'air suivant la caractérisation par formules temporelles Pour l'étude de la diagnosticabilité des défauts détectables suivant cette caractérisation par formules temporelles, nous avons utilisé les formules temporelles *ad-hoc* Ψ_F de diagnosticabilité de chacun des défauts $F \in \Gamma^{Detect}_{AirLine}$ ainsi que celles Ψ^e_F d'éligibilité et Ψ^i_F d'isolabilité. Nous avons par ailleurs utilisé la formule normale φ_{Norm} et celles des défauts $\varphi_{LockCmpr}$, φ_{SenQ} et φ_{SenP} afin de connaître leurs validités dans le domaine temporel $[20; 50]$. Le tableau 4.6 de la page 158 reprend ces différentes formules.

Formules		
φ_{Norm}	:=	$\bigwedge_{i\in\{1;\ldots;4\}} \big([(c_Q \in C^i_Q \wedge stat_{c_Q}) \Rightarrow Stat_{C^i_Q}] \wedge [(c_Q \in C^i_Q \wedge dyn_{c_Q}) \Rightarrow Dyn_{C^i_Q}]\big)$
$\varphi_{LockCmpr}$	:=	$\big[(c_Q = 0) \wedge (y_Q = 0) \wedge (y_P = 1)\big] \vee \big[(c_Q \neq 0) \wedge (y_Q = 0) \wedge (y_P = 1) \wedge (u_\omega = 1) \wedge (u_x = 1)\big]$
φ_{SenQ}	:=	$\big[(y_Q = 0) \wedge (1{,}77 \leq y_P \leq 2) \wedge (u_\omega = 1) \wedge (u_x = 1)\big]$
φ_{SenP}	:=	$\big[(y_P = 0) \wedge (y_Q \neq 0)\big]$
$\Psi_{LockCmpr}$	:=	$\mathrm{Top}(sf_{\{t_n\}}) \Rightarrow \big[\varphi_{Norm} \mathrm{U}_{[0;4]} \big(\mathrm{Bot}(\varphi_{Norm}) \wedge \mathrm{G}_{[2,5]}[\varphi_{LockCmpr} \wedge \neg\varphi_{SenQ} \wedge \neg\varphi_{SenP} \wedge \neg\varphi_{Norm}]\big)\big]$
Ψ_{SenQ}	:=	$\mathrm{Top}(sf_{\{t_n\}}) \Rightarrow \big[\varphi_{Norm} \mathrm{U}_{[0;4]} \big(\mathrm{Bot}(\varphi_{Norm}) \wedge \mathrm{G}_{[2,5]}[\varphi_{SenQ} \wedge \neg\varphi_{LockCmpr} \wedge \neg\varphi_{SenP} \wedge \neg\varphi_{Norm}]\big)\big]$
Ψ_{SenP}	:=	$\mathrm{Top}(sf_{\{t_n\}}) \Rightarrow \big[\varphi_{Norm} \mathrm{U}_{[0;4]} \big(\mathrm{Bot}(\varphi_{Norm}) \wedge \mathrm{G}_{[2,5]}[\varphi_{SenP} \wedge \neg\varphi_{LockCmpr} \wedge \neg\varphi_{SenQ} \wedge \neg\varphi_{Norm}]\big)\big]$
$\Psi^e_{LockCmpr}$	:=	$\mathrm{Top}(sf_{\{t_n\}}) \Rightarrow \mathrm{E}_{[0;4]} \big(\mathrm{G}_{[2,5]} \varphi_{LockCmpr}\big)$
Ψ^e_{SenQ}	:=	$\mathrm{Top}(sf_{\{t_n\}}) \Rightarrow \mathrm{E}_{[0;4]} \big(\mathrm{G}_{[2,5]} \varphi_{SenQ}\big)$
Ψ^e_{SenP}	:=	$\mathrm{Top}(sf_{\{t_n\}}) \Rightarrow \mathrm{E}_{[0;4]} \big(\mathrm{G}_{[2,5]} \varphi_{SenP}\big)$
$\Psi^i_{LockCmpr}$	:=	$\mathrm{Top}(sf_{\{t_n\}}) \Rightarrow \big[\mathrm{E}_{[0;4]} \big(\mathrm{G}_{[2,5]}[\neg\varphi_{SenQ} \wedge \neg\varphi_{SenP} \wedge \neg\varphi_{Norm}]\big)\big]$
Ψ^i_{SenQ}	:=	$\mathrm{Top}(sf_{\{t_n\}}) \Rightarrow \big[\mathrm{E}_{[0;4]} \big(\mathrm{G}_{[2,5]}[\neg\varphi_{LockCmpr} \wedge \neg\varphi_{SenP} \wedge \neg\varphi_{Norm}]\big)\big]$
Ψ^i_{SenP}	:=	$\mathrm{Top}(sf_{\{t_n\}}) \Rightarrow \big[\mathrm{E}_{[0;4]} \big(\mathrm{G}_{[2,5]}[\neg\varphi_{LockCmpr} \wedge \neg\varphi_{SenQ} \wedge \neg\varphi_{Norm}]\big)\big]$

Tableau 4.6 – Formules temporelles d'étude de la diagnosticabilité pour chacun des défauts détectables.

Les résultats de cette étude montrent que seul le défaut F_{SenP} de mesure du capteur de pression est diagnosticable. Le défaut $F_{LockCmpr}$ de blocage du compresseur et le défaut F_{SenQ} de mesure du capteur de débit ne sont pas diagnosticables.

Remarquons d'abord que pour chacun de ces défauts $F \in \Gamma^{Detect}_{AirLine}$, aucune des autres formules $\varphi_{F'}$ d'un autre défaut $F' \in \Gamma^{Detect}_{AirLine} \setminus \{F\}$ n'est satisfaite à un instant t du domaine temporel $[20; 50]$. Cela signifie que ce n'est pas à cause d'un problème d'isolation vis-à-vis des autres défauts qu'un défaut n'est pas diagnosticable.

Concernant le défaut $F_{LockCmpr}$ de blocage du compresseur, il n'est pas diagnosticable à cause de la validité de sa formule de défaut $\varphi_{LockCmpr}$. Bien qu'elle soit toujours valide à partir d'un certain délai après l'occurrence du défaut, c'est néanmoins ce délai qui peut parfois être trop long par rapport au délai $h = 2$ d'isolation suite à un instant t_k de détection : c'est-à-dire que cette formule $\varphi_{LockCmpr}$ n'est pas encore valide à l'instant $t_k + 2$. Ce cas se présente uniquement avec des comportements pour lesquels la consigne évolue dans ou vers la partie C^1_Q. Cela signifie que pour de faibles demandes de débit (i.e. : des changements de la consigne dans la partie C^1_Q) avec la présence de ce défaut $F_{LockCmpr}$, la commande u_ω du compresseur évolue très lentement vers sa valeur maximale 1, alors que pour des

changements de la consigne dans les autres parties (i.e. : les parties C_Q^2, C_Q^3 et C_Q^4), cette commande u_ω évolue très rapidement vers la valeur maximale 1. De ce fait et suite au délai d'isolation $h = 2$ après la détection du défaut à un instant t_k, cette formule $\varphi_{LockCmpr}$ n'est pas encore valide ; elle ne l'est qu'à partir d'un instant légèrement supérieur $t_k + (2 + \varepsilon)$ où ε vaut entre une demi et deux secondes.

Concernant le défaut F_{SenQ} de mesure du capteur de débit, il n'est pas diagnosticable en partie pour la même raison que le défaut $F_{LockCmpr}$ de blocage du compresseur. Sa formule φ_{SenQ} n'est soit pas du tout valide après l'occurrence du défaut, soit elle l'est mais trop tard par rapport au délai $h = 2$ d'isolation. Ici encore, ces cas ne se présentent qu'avec des comportements pour lesquels la consigne évolue dans ou vers la partie C_Q^1 : c'est-à-dire lorsque le débit est faible, les commandes u_ω du compresseur et u_x de l'électrovanne ne sont pas au maximum et la mesure y_P de la pression n'est pas à son maximum (i.e. : comprise entre 1,77 et 2 bar).

Pour rendre ces deux défauts diagnosticables, il faudrait enrichir les deux formules $\varphi_{LockCmpr}$ et φ_{SenQ} pour le cas de la consigne évoluant dans ou vers cette partie C_Q^1. Nous ne l'avons pas réalisé dans ces travaux pour ne pas alourdir les formules présentées. Comme nous le verrons au chapitre suivant 5 de génération du diagnostiqueur, nous nous limiterons aux évolutions de la consigne c_Q de débit uniquement dans les parties $C_Q^2 = [3; 7]$, $C_Q^3 = [8; 27]$ et $C_Q^4 = [28; 30]$.

4.5.3.3 Analyse des résultats de l'étude de la diagnosticabilité

Nous venons de mener l'étude de la diagnosticabilité de l'ensemble $\Gamma_{AirLine}$ des défauts de la ligne d'air. Nous avons d'abord déterminé l'ensemble $Cons_{AirLine}$ des instructions ainsi que les ensembles $\Omega_{(F,cs_{(a,b)})}$ des occurrences pour chacun des défauts $F \in \Gamma_{AirLine} \setminus \{F_{Norm}\}$, qui nous ont permis de construire l'ensemble de tous les comportements observables sous la présence des défauts. Ces ensembles d'instructions et d'occurrences ont été élaborés de manière empirique afin de rendre compte, de la manière la plus représentative, de tous les comportements potentiels du système sous la présence des défauts (cas normal F_{Norm} compris). Il serait néanmoins judicieux, pour de futurs travaux, d'élaborer une méthodologie permettant de définir ces ensembles représentatifs sans qu'il y ait l'intégralité des cas possibles, ce que nous avons fait lors de cette étude.

L'étude de la diagnosticabilité a ensuite été menée suivant les deux caractérisations que nous avons présentées dans la partie théorique. Les mêmes défauts détectables ont été obtenus suivant ces deux caractérisations : le défaut $F_{LockCmpr}$ de blocage du compresseur et les défauts F_{SenQ} et F_{SenP} de mesure des capteurs de débit et de pression. Les défauts non-détectables sont classés en deux catégories : les défauts faiblement progressifs pour lesquels nous avons déjà expliqué comment résoudre ce problème de détectabilité pour de futurs travaux, le défaut F_{LockEV} de blocage de l'électrovanne.

Concernant ce défaut F_{LockEV}, sa non-détectabilité vient d'une part des occurrences considérées ainsi que du type de blocage considéré. Nous avons remarqué que pour des occurrences apparaissant durant un fonctionnement statique du système, ce défaut n'est pas détectable du fait que le système est stationnaire. Aucun changement de la consigne n'est demandé et la commande u_x de l'électrovanne n'est ainsi pas modifiée, ce qui signifie qu'il n'est donc pas possible de déterminer qu'elle est bloquée. Remarquons que nous avons considéré des blocages à n'importe quel angle d'ouverture. Or selon certains retours d'expériences, ces blocages sont généralement très difficilement isolables du fait du nombre important de cas possibles (suivant l'angle d'ouverture) ; c'est ce que nous remarquons dans nos résultats. Nous avions tenté de définir une formule temporelle caractérisant ce défaut lorsqu'il est détectable, mais sans succès du fait du nombre important de cas possibles. Il serait préférable, pour de futurs travaux, de ne considérer que certains blocages et d'étudier ainsi la diagnosticabilité de ces cas particuliers, par exemple les blocages de butée ou dus à une défaillance de la partie électronique et pour lesquels l'électrovanne revient à son angle initial d'ouverture grâce à un ressort de rappel.

Enfin les trois défauts $F_{LockCmpr}$, F_{SenQ} et F_{SenP} sont diagnosticables avec la caractérisation parfaite, alors que seul F_{SenP} est diagnosticable avec la caractérisation par formules temporelles. Pour les rendre diagnosticables, nous aurions pu considérer un délai h de détection plus important

concernant $F_{LockCmpr}$ ou ajouter certaines conditions dans la formule temporelle φ_{SenQ} concernant F_{SenQ}. Néanmoins et comme ces deux défauts ne sont pas diagnosticables que pour certaines évolutions de la consigne bien localisées (i.e. : dans la partie C_Q^1), nous avons préféré garder ces formules et ne considérer, pour le chapitre suivant 5 de génération du diagnostiqueur, des évolutions de la consigne c_Q de débit qu'uniquement dans les parties C_Q^2, C_Q^3 et C_Q^4 et pas dans la partie C_Q^1. Il faudrait bien entendu pour une réalisation réelle reprendre ces formules afin de rendre ces défauts diagnosticables, mais il ne s'agissait ici que de présenter un exemple applicatif.

4.6 Conclusion sur l'étude de la diagnosticabilité

Nous venons de présenter dans ce chapitre une méthodologie permettant l'étude de la diagnosticabilité d'un système, plus précisément de la diagnosticabilité de chacun des défauts potentiels du système qui ont été répertoriés lors de l'étude de sûreté de fonctionnement.

Suite à la description intuitive de cette notion de diagnosticabilité, nous avons défini les comportements observables du système en fonctionnement normal ou sous la présence d'un défaut. Il s'agit de l'évolution des valeurs des variables observables du système suivant différentes instructions de l'opérateur.

Nous avons ensuite défini la notion de diagnosticabilité des défauts à partir d'une famille de propriétés caractéristiques du fonctionnement du système sous la présence de chacun des défauts (y compris le fonctionnement normal du système) ; ce que nous avons nommé une *caractérisation de défauts*. Ces propriétés, évaluées par les comportements observables du système, représentent les règles de fonctionnement du diagnostiqueur. La notion de diagnosticabilité est donc définie à partir d'une telle caractérisation de défauts. Le cas normal est diagnosticable si tout comportement observable normal valide toujours la propriété normale à tout instant du temps. Un défaut est diagnosticable si dès qu'il apparaît et dans une fenêtre temporelle bornée par une certaine borne de diagnostic, la propriété normale devient invalide puis, suite à un certain délai d'isolation et durant un certain délai de confiance, seule la propriété du défaut concerné est toujours valide.

Deux caractérisations de défauts furent présentées. La première, que nous avons nommée *caractérisation parfaite* et qui est basée sur un formalisme ensembliste, nous permet de nous assurer de manière intrinsèque de la diagnosticabilité des défauts répertoriés. Elle consiste en l'ensemble de tous les morceaux de comportements observables du système sur des fenêtres temporelles de longueur b et à partir des occurrences des défauts. La seconde caractérisation, que nous avons nommée *caractérisation par formules temporelles*, est basée sur un formalisme de logique temporelle et est constituée d'un ensemble de formules temporelles liant les variables observables du système. Ces caractérisations rendent bien compte de notre souhait de considérer une approche de diagnostic à base de modèles incluant un aspect temporel. Néanmoins et comme nous l'avons vu, la caractérisation parfaite n'est d'une part pas optimisée suivant l'architecture considérée du système, et risque d'autre part de ne pas être applicable pour des systèmes complexes. À l'inverse de la caractérisation par formules temporelles qui a l'avantage, comme nous le verrons au chapitre suivant, d'être facilement exploitable pour générer un diagnostiqueur.

Enfin, tous ces concepts furent mis en application sur le cas d'étude. Les résultats obtenus de cette étude nous ont permis de bien nous rendre compte de la complémentarité des deux caractérisations. La mise en application de la caractérisation parfaite étant très simple, elle permet donc facilement d'obtenir les défauts intrinsèquement diagnosticables. La caractérisation par formules temporelles étant plus difficile à obtenir, notamment pour les formules des défauts, elle nous a néanmoins fourni des résultats probants. Indiquons cependant qu'un futur travail concernant l'obtention de ces formules de défauts serait utile.

Grâce à cette étude de la diagnosticabilité des défauts du système, il va nous être possible de

générer le diagnostiqueur associé. C'est ce que nous allons faire dans le chapitre qui va suivre.

Chapitre 5

La génération du diagnostiqueur du système

Au chapitre précédent, nous avons étudié la diagnosticabilité des défauts répertoriés du système. Nous avons ainsi vérifié qu'en fonctionnement le diagnostiqueur sera toujours capable de détecter et d'isoler sans ambiguïté un défaut préalablement répertorié lorsqu'il apparaît. Suite à cette étude, il est maintenant possible de générer ce diagnostiqueur suivant la caractérisation Λ de défauts utilisée pour cette étude : c'est-à-dire selon les propriétés caractéristiques P_F, pour chacun des défauts diagnosticables $F \in \Gamma^{Diag}$, construites lors de l'étude.

Nous allons d'abord voir comment fonctionne un diagnostiqueur. Suite à une description intuitive puis formelle de son algorithme de fonctionnement, nous verrons comment l'implanter dans le système de pilotage d'un système piloté. Nous ferons ensuite une étude de complexité sur ce fonctionnement suivant une caractérisation de défauts considérée, afin de nous assurer, avant toute implémentation, de la praticabilité du fonctionnement. Nous appliquerons enfin cette partie sur le cas d'étude.

5.1 Fonctionnement d'un diagnostiqueur

Après avoir rappelé comment fonctionne un diagnostiqueur, nous ferons un retour sur la signification d'une étude de diagnosticabilité des défauts, ceci afin de pouvoir intégrer au fonctionnement du diagnostiqueur le cas potentiel de défauts non-répertoriés. Nous pourrons alors présenter ensuite formellement son fonctionnement, puis l'implanter dans le système de pilotage du système piloté considéré.

5.1.1 Fonctionnement intuitif d'un diagnostiqueur

Nous avons déjà vu qu'en fonctionnement, un diagnostiqueur vérifie que le *comportement observé* du système satisfait une certaine propriété de bon fonctionnement P_{F_0}. Si cette propriété est satisfaite, alors le fonctionnement du système est jugé normal et le diagnostiqueur continue d'analyser le comportement observé du système toujours suivant cette propriété de bon fonctionnement. Dans le cas contraire, cette propriété de bon fonctionnement n'est donc pas satisfaite et le fonctionnement du système est jugé anormal : cela signifie qu'il y a détection d'un mauvais fonctionnement. Suite à cette détection et après un délai potentiel h d'isolation, le diagnostiqueur va alors vérifier quelle unique propriété de mauvais fonctionnement P_F, caractérisant le fonctionnement du système sous la présence d'un défaut diagnosticable $F \in \Gamma^{Diag}$, satisfait ce comportement observé du système : il s'agit de la phase d'isolation du défaut.

Cette famille $\Lambda = (\mathrm{P}_F)_{F \in \Gamma}$ de propriétés caractéristiques, que nous avons nommée une *caractérisation de défauts*, nous a permis d'étudier la diagnosticabilité de chacun des défauts $F \in \Gamma$ préalablement

répertoriés. Cette étude de la diagnosticabilité, menée suivant cette caractérisation Λ de défauts, nous a garanti que les défauts diagnosticables $F \in \Gamma^{Diag} \setminus \{F_0\}$ seront bien diagnostiqués. Cette garantie nous permet donc de construire un diagnostiqueur suivant ces propriétés caractéristiques P_F.

5.1.2 Correction et complétude entre l'étude de la diagnosticabilité et le passage au diagnostiqueur

Il est important de signaler que cette étude de diagnosticabilité ne nous assure uniquement qu'un défaut diagnosticable sera bien diagnostiqué par le diagnostiqueur lorsqu'il apparaîtra. Cela signifie d'une part que lorsque le système fonctionne normalement, alors le diagnostiqueur s'en assure sans ambiguïté, car nous avons considéré le cas normal F_0 (i.e. : le bon fonctionnement du système) dans l'étude de la diagnosticabilité. D'autre part, lorsqu'un défaut préalablement conclu comme diagnosticable lors de l'étude de la diagnosticabilité apparaîtra, il sera alors bien diagnostiqué sans ambiguïté après cette apparition.

Or cette étude de la diagnosticabilité des défauts a été menée suivant certaines hypothèses de modélisation du bon fonctionnement et des perturbations, ainsi que des choix sur les défauts à considérer. Nous avons en effet considéré d'une part que le modèle utilisé pour cette étude était « parfait », ce qui signifie exhaustivité dans la représentation du bon fonctionnement du système et de toutes les perturbations possibles, ce qui peut ne pas être possible. D'autre part, l'ensemble Γ des défauts a été élaboré lors d'une étude de sûreté de fonctionnement qui détermine que ceux à prendre en compte par un outil de diagnostic sont ceux jugés comme les plus critiques pour l'intégrité du système ou son environnement. Ceci signifie non seulement que nous avons supposé être exhaustif sur les défauts les plus critiques du système, mais que nous avons de plus écarté de l'étude tous les autres défauts potentiels peu critiques. Remarquons par ailleurs que nous nous sommes restreint aux défauts permanents ou transitoires mais dont la période de présence est supérieure à la borne b de diagnostic. Il serait par conséquent nécessaire de s'interroger sur l'impact de toutes ces hypothèses sur le fonctionnement du diagnostiqueur.

Il est donc important de prendre en compte ces hypothèses dans l'élaboration du diagnostiqueur, notamment les cas où certaines ne seraient pas valides. En effet, que doit conclure le diagnostiqueur lorsque, suite à une détection d'un fonctionnement anormal du système, aucun défaut n'est isolé après le délai h d'isolation ? De même lorsque plusieurs défauts sont isolés ? Ce fonctionnement anormal est-il par ailleurs encore présent après ce délai d'isolation ? Ces interrogations concernent autant le cas des défauts non pris en compte lors des études de sûreté de fonctionnement et de diagnosticabilité, que le cas des perturbations ou erreurs de modélisation de la représentation du bon fonctionnement du système. La deux premières sont justifiées pour des cas durables de défauts, perturbations ou erreurs de modélisation, alors que la troisième est, quant à elle, justifiée pour des cas furtifs dont la durée de présence est très courte comparée au délai d'isolation.

Les résultats de ces interrogations doivent être intégrés lors de l'élaboration du fonctionnement du diagnostiqueur. Suite à une détection et après le délai h d'isolation, plusieurs cas potentiels peuvent se présenter concernant les validités des propriétés.

1. La propriété normale est toujours invalide et une seule propriété de défauts est valide. Cela signifie que le défaut de cette propriété est apparu.
2. La propriété normale est toujours invalide mais aucune propriété de défauts n'est valide. Cela signifie qu'un défaut non-répertorié est apparu, car un défaut répertorié ne valide que sa propriété de défaut.
3. La propriété normale est toujours invalide mais plusieurs propriétés de défauts sont valides. Cela signifie aussi qu'un défaut non-répertorié est apparu, car un défaut répertorié ne valide que sa propriété de défaut.

4. La propriété normale est redevenue valide et aucune propriété de défauts n'est valide. Cela signifie alors qu'un fonctionnement anormal furtif du système est apparu et a donc disparu.
5. La propriété normale est redevenue valide mais au moins une propriété de défauts est valide. Cela signifie alors qu'un défaut non-répertorié est apparu, car un défaut répertorié ne valide que sa propriété de défaut et pas la propriété normale.

Dans le premier cas et grâce à l'étude de la diagnosticabilité, le diagnostiqueur peut conclure sans ambiguïté au défaut apparu. Dans les deuxième et quatrième cas où un défaut non-répertorié est apparu, il est nécessaire de déterminer un mode dégradé adéquat : par exemple un mode de fonctionnement où l'opérateur est informé de la présence d'un défaut non-répertorié. Enfin dans le troisième cas où un fonctionnement anormal furtif est apparu et a disparu, le diagnostiqueur peut conclure à un tel fonctionnement anormal furtif et considérer alors que le système re-fonctionne normalement. Ces solutions permettent ainsi de rendre le diagnostiqueur robuste aux cas non traités.

5.1.3 Formalisation du fonctionnement d'un diagnostiqueur issu de l'étude de la diagnosticabilité

Pour être plus précis et formel dans le fonctionnement d'un diagnostiqueur issu de l'étude de la diagnosticabilité, celui-ci analyse un flux continu de données constitué par le vecteur des valeurs des variables observables du système à chaque instant du temps : c'est-à-dire le vecteur $v_{obs}(\mathrm{t}_r) = (c(t), u(t), y(t), u^M(t), y^M(t))$. Si nécessaire et suivant la méthodologie de diagnostic utilisée, le diagnostiqueur va avoir besoin d'enregistrer ce flux de données sur une certaine durée temporelle $\lambda \in \mathbb{T}_\iota$. À chaque instant du temps, il aura donc en mémoire ce flux de données depuis λ unités de temps : c'est-à-dire qu'il aura en mémoire l'ensemble ordonné $Obs(\mathrm{t}_r) = \{v_{obs}(\mathrm{t}_r - \lambda); v_{obs}(\mathrm{t}_r - (\lambda+1)); \ldots; v_{obs}(\mathrm{t}_r)\}$ des vecteurs des valeurs des variables observables du système depuis λ unités de temps, et qui représente le *comportement observé* du système. Ce comportement observé va être analysé par la propriété normale P_{F_0} afin de vérifier que le système fonctionne normalement. Dans le cas où cette propriété normale P_{F_0} n'est pas satisfaite, ce comportement observé va alors être analysé par les propriétés P_F de chacun des défauts diagnosticables $F \in \Gamma^{Diag} \setminus \{F_0\}$ après le délai h d'isolation.

Formellement, cela signifie qu'à chaque instant $\mathrm{t}_r \in \mathbb{T}_\iota$ du temps, le diagnostiqueur réalise dans l'ordre les actions suivantes :

(A) il enregistre le vecteur $v_{obs}(\mathrm{t}_r) = (c(\mathrm{t}_r), u(\mathrm{t}_r), y(\mathrm{t}_r), u^M(\mathrm{t}_r), y^M(\mathrm{t}_r))$ des valeurs des variables observables du système à cet instant t_r du temps et l'ajoute aux vecteurs qu'il a déjà enregistrés depuis l'instant $\mathrm{t}_r - \lambda$; cela signifie qu'il supprime le vecteur $v_{obs}(\mathrm{t}_r - \lambda - 1)$ des valeurs des variables observables du système enregistré à l'instant $\mathrm{t}_r - \lambda - 1$ du temps. Il aura donc en mémoire le comportement observé $Obs(\mathrm{t}_r)$ des vecteurs $v_{obs}(\mathrm{t})$ constitué des valeurs des variables observables pour chaque instant t de la fenêtre temporelle glissante $[\mathrm{t}_r - \lambda; \mathrm{t}_r]$ (i.e. : l'ensemble $Obs(\mathrm{t}_r) = \{v_{obs}(\mathrm{t}_r - \lambda); v_{obs}(\mathrm{t}_r - (\lambda+1)); \ldots; v_{obs}(\mathrm{t}_r - 1); v_{obs}(\mathrm{t}_r)\}$).

(B) il vérifie que ce comportement observé $Obs(\mathrm{t}_r)$ satisfait la propriété normale P_{F_0} (i.e. : que $\mathrm{P}_{F_0}(Obs(\mathrm{t}_r), \mathrm{t}_r) = \texttt{True}$).

(1) si oui (i.e. : $\mathrm{P}_{F_0}(Obs(\mathrm{t}_r), \mathrm{t}_r) = \texttt{True}$), le diagnostiqueur retourne au point (A) en considérant l'instant suivant de t_r (i.e. : l'instant $\mathrm{t}_r{}^+ \in \mathbb{T}_\iota$).

(2) dans le cas contraire (i.e. : $\mathrm{P}_{F_0}(Obs(\mathrm{t}_r), \mathrm{t}_r) = \texttt{False}$) le diagnostiqueur aura donc détecté un mauvais fonctionnement et va réaliser dans l'ordre les actions suivantes :

(a) Il va continuer à enregistrer le vecteur $v_{obs}(\mathrm{t})$ des valeurs des variables observables du système jusqu'à l'instant $\mathrm{t}_r + h$, pour constituer le comportement observé $Obs(\mathrm{t}_r + h)$; tout en supprimant à chaque fois le vecteur $v_{obs}(\mathrm{t} - \lambda - 1)$.

(b) À cet instant $\mathrm{t}_r + h$, il va vérifier quelle propriété P_F, pour un défaut diagnosticable $F \in \Gamma^{Diag} \setminus \{F_0\}$, est satisfaite par ce comportement observé $Obs(\mathrm{t}_r + h)$: c'est-à-dire

pour quelle propriété P_F il y a $\mathrm{P}_F(Obs(\mathrm{t}_r + h), \mathrm{t}_r + h) = \texttt{True}$.
Quatre cas peuvent se présenter :

(i) une unique propriété P_F est satisfaite (elle est telle que $\mathrm{P}_F(Obs(\mathrm{t}_r + h), \mathrm{t}_r + h) = \texttt{True}$) et la propriété normale est toujours insatisfaite. Alors selon l'étude de la diagnosticabilité, le diagnostiqueur peut donc conclure à la présence du défaut $F \in \Gamma^{Diag} \setminus \{F_0\}$.

(ii) aucune ou au moins deux propriétés P_F sont satisfaites, pour n'importe quels défauts diagnosticables de $\Gamma^{Diag} \setminus \{F_0\}$, et la propriété normale est toujours insatisfaite. Alors un fonctionnement anormal non-répertorié est présent.

(iii) au moins une propriété P_F est satisfaite, pour n'importe quel défaut diagnosticable de $\Gamma^{Diag} \setminus \{F_0\}$, et la propriété normale est redevenue satisfaite. Alors un fonctionnement anormal non-répertorié est présent.

(iv) aucune propriété P_F n'est satisfaite pour n'importe quel défaut diagnosticable $F \in \Gamma^{Diag} \setminus \{F_0\}$ et la propriété normale est redevenue satisfaite. Alors il est possible qu'il n'y ait eu qu'un fonctionnement anormal transitoire et le diagnostiqueur retourne au point (A) en considérant l'instant suivant de $\mathrm{t}_r + h$, l'instant $(\mathrm{t}_r + h)^+ \in \mathbb{T}_\iota$.

Remarquons que dans le cas (ii) où au moins deux propriétés P_F et $\mathrm{P}_{F'}$ sont satisfaites, pour n'importe quels défauts diagnosticables de $F, F' \in \Gamma^{Diag} \setminus \{F_0\}$, nous pourrions alors être en présence des deux défauts F et F' en simultané. Ce serait donc notre hypothèse de non-occurrence de défauts simultanés qui ne serait plus valide et bien que nous concluons à un comportement anormal, le diagnostiqueur pourrait néanmoins indiquer à l'opérateur la levée de ces deux défauts.

5.1.4 Implantation d'un diagnostiqueur

Du point de vue de l'implantation du diagnostiqueur dans le système, il peut très bien être intégré au système de pilotage ou en être dissocié. L'important est qu'il soit relié d'une part aux différentes variables observables du système, afin d'effectuer sa fonction de diagnostic en surveillant ces variables observables, et d'autre part au module de décision afin de l'informer sur la présence de défauts diagnostiqués.

Nous avons considéré une architecture particulière afin de faire un diagnostic des défauts du système en utilisant une approche à base de modèles. Nous avons pour cela considéré le système réel bouclé avec son propre module de régulation et un modèle de bon fonctionnement (embarqué) du système avec, lui aussi, son propre module de régulation identique à celui du système réel. Néanmoins et comme cette structure n'est pas inhérente à tout dispositif de diagnostic, nous n'imposons pas que cette seconde boucle (i.e. : le modèle bouclé avec son module de régulation) fasse partie du diagnostiqueur. En effet, avec la méthodologie que nous présentons dans ce document, il est tout à fait possible de s'abstraire d'un modèle embarqué de bon fonctionnement et de considérer d'autres architectures du système, uniquement le système opérant sans la partie pilotage par exemple. En effet, suivant le type de défauts à diagnostiquer, certains pourraient très bien avoir une caractérisation intrinsèque ne nécessitant pas de référence à un modèle normal ; ainsi et dans le cas où tous les défauts devant être diagnostiqués seraient de ce type, il ne serait alors pas nécessaire de simuler un modèle embarqué de bon fonctionnement du système.

Néanmoins et afin de justifier d'une solution toute intégrée de contrôle-commande tolérant aux défauts, nous supposons que le diagnostiqueur et la boucle modèle sont intégrés dans le système de pilotage. Le diagnostiqueur est directement relié aux variables observables du système (i.e. : les consignes c, les commandes réelles u et les mesures réelles y), ainsi qu'à celles de la boucle modèle (i.e. : les commandes modèles u^M et les mesures modèles y^M). Le module de diagnostic fonctionne donc en analysant le comportement observé du système, constitué des variables observables réelles et celles du modèle depuis une certaine durée temporelle $\lambda \in \mathbb{T}_\iota$, et il informe le module de décision

lorsqu'un défaut est diagnostiqué. La figure 5.1 ci-dessous représente cette intégration du module de diagnostic et de la boucle modèle, entourée en rouge, directement dans le système de pilotage.

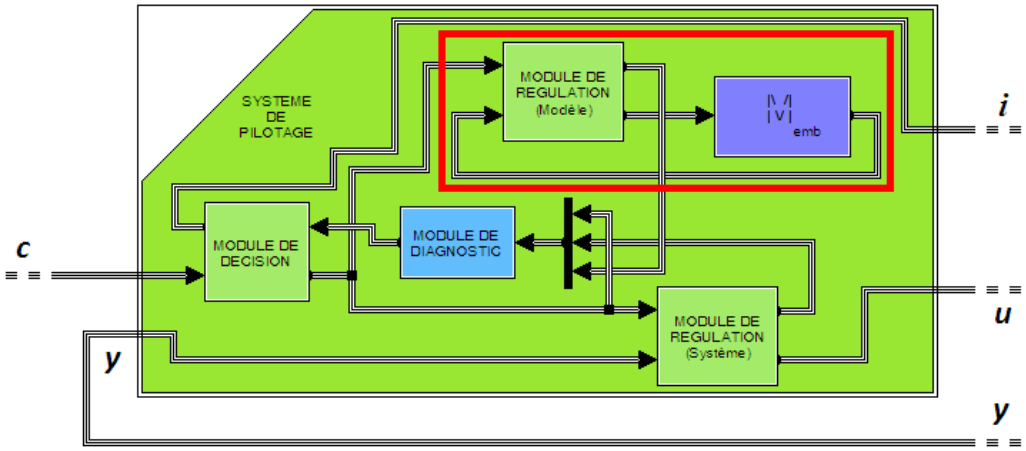

Figure 5.1 – Architecture du système de pilotage intégrant un diagnostiqueur et la boucle modèle.

5.2 Complexité de fonctionnement d'un diagnostiqueur

L'implémentation d'un diagnostiqueur est réalisée relativement à une caractérisation de défauts utilisée pour l'étude préalable de la diagnosticabilité. Or, et selon le fonctionnement du diagnostiqueur que nous venons de présenter, il va néanmoins falloir s'assurer, avant cette implémentation, que l'implantation permette la réalisation de cet algorithme. C'est-à-dire d'une part que la capacité de calcul du diagnostiqueur permette d'effectuer tous les calculs nécessaires dans les délais impartis lors du fonctionnement, et d'autre part que sa capacité mémoire permette de stocker la quantité d'informations manipulées lors du fonctionnement. Cette constatation est justifiée par le fait que durant toute l'étude de la diagnosticabilité ainsi d'ailleurs que durant la présentation du fonctionnement du diagnostiqueur, nous avons supposé que ses capacités de stockage et de calcul sont « dimensionnées » pour accomplir les différentes tâches de traitement.

Cette étude préliminaire est la *complexité de fonctionnement* du diagnostiqueur et se réalise donc relativement à une caractérisation de défauts. Il va s'agir d'obtenir l'espace mémoire et le temps de calcul nécessaires au fonctionnement du diagnostiqueur. Cela se justifie d'autant plus par le fait que les ressources disponibles, tant en puissance de calcul qu'en capacité de stockage, sont généralement fortement limitées pour les systèmes pilotés utilisés en embarqué.

Remarquons que nous avions commencé à aborder cette problématique de complexité de fonctionnement, au chapitre précédent, en indiquant que la caractérisation parfaite ne peut être praticable pour des systèmes complexes du fait de la quantité importante d'informations à sauvegarder pour effectuer les comparaisons : les comportements observables bornés normaux et ceux sous la présence de tous les défauts diagnosticables. Il s'agissait donc d'une première intuition de la complexité de fonctionnement de cette caractérisation et nous nous rendions alors déjà compte que le fonctionnement du diagnostiqueur était lié à la complexité de la caractérisation de défauts utilisée.

Pour la suite de cette partie, nous nous référons principalement à [LdR96] ou [Pap93] concernant les notions classiques de complexité. Nous ne réaliserons néanmoins pas une telle étude classique de complexité algorithmique, mais tenterons d'expliquer comment obtenir les quantités de stockage et de puissance de calcul nécessaires au fonctionnement du diagnostiqueur.

5.2.1 La complexité en temps de calcul et en espace mémoire

Ce qui nous intéresse est d'obtenir une borne de puissance de calcul, nommée *complexité en temps*, ainsi qu'une borne d'espace mémoire, nommée *complexité en espace* du fonctionnement d'un diagnostiqueur suivant une certaine caractérisation de défauts Λ.

La complexité en espace signifie l'*espace mémoire* caractérisant la taille nécessaire de la mémoire pour enregistrer les différentes valeurs, nommées *données élémentaires*, que manipulera le diagnostiqueur lors de son fonctionnement. Une *donnée élémentaire* est la valeur d'une variable ou d'une constante utilisée par le diagnostiqueur. L'espace mémoire est donc le nombre nécessaire de cases mémoires pour que le diagnostiqueur puisse enregistrer toutes les valeurs qu'il va utiliser en fonctionnement.

La complexité en temps signifie le *temps de calcul* nécessaire au fonctionnement du diagnostiqueur, qui est déterminé par le nombre de *calculs élémentaires* que devra effectuer le processeur du calculateur du diagnostiqueur à chaque instant de sa période de fonctionnement ; notons que cette période de fonctionnement est généralement plus rapide que la période d'échantillonnage ι du temps d'exécution $\mathbb{T}_\iota$ du diagnostiqueur. Un *calcul élémentaire* est soit un accès mémoire en lecture ou écriture à une donnée élémentaire, soit une opération (arithmétique ou comparaison) entre des données élémentaires.

5.2.2 La complexité dans le pire des cas

En fonctionnement, le diagnostiqueur réalise une suite de procédures accomplies à chaque instant du temps. Comme nous l'avons décrit au début de ce chapitre, à un instant t_r quelconque du temps $\mathbb{T}_\iota$, le diagnostiqueur peut être dans trois situations distinctes selon qu'il y ait eu ou non une détection et à quel instant du temps il se trouve après une détection :

1. il n'y a pas eu de détection d'un fonctionnement anormal du système et le diagnostiqueur enregistre donc le vecteur $v_{obs}(\mathrm{t}_r)$ pour créer le comportement observé $Obs(\mathrm{t}_r)$, puis vérifie uniquement la propriété normale P_{F_0} pour ce comportement observé $Obs(\mathrm{t}_r)$.
2. il y a eu détection d'un fonctionnement anormal du système à un instant t_k inférieur à l'instant t_r et le diagnostiqueur attend le délai h d'isolation. Cet instant t_r se situe donc entre l'instant t_k de détection et l'instant $\mathrm{t}_k + h$ d'isolation (i.e. : $\mathrm{t}_k < \mathrm{t}_r < \mathrm{t}_k + h$). Le diagnostiqueur enregistre donc le vecteur $v_{obs}(\mathrm{t}_r)$ pour créer le comportement observé $Obs(\mathrm{t}_r)$, mais ne vérifie aucune propriété P_F pour n'importe quel défaut diagnosticable $F \in \Gamma^{Diag}$.
3. il y a eu détection d'un fonctionnement anormal du système à l'instant $\mathrm{t}_k = \mathrm{t}_r - h$ et le diagnostiqueur enregistre donc le vecteur $v_{obs}(\mathrm{t}_r)$ pour créer le comportement observé $Obs(\mathrm{t}_r)$, puis il vérifie toutes les propriétés P_F pour ce comportement observé $Obs(\mathrm{t}_r)$ pour tous les défauts diagnosticables $F \in \Gamma^{Diag}$.

Nous pouvons donc remarquer que la quantité de « travail » à réaliser, c'est-à-dire tous les calculs élémentaires à faire, n'est pas identique dans ces trois cas ; ce qui signifie que la puissance de calcul et l'espace mémoire nécessaires ne seront donc pas identiques. Or, il nous faut considérer le pire des cas qui puisse arriver : c'est-à-dire le cas où il y ait la charge de calcul ainsi qu'un besoin de stockage de données les plus importants. Cela signifie que nous allons calculer une complexité « dans le pire des cas ». Il s'agit bien sûr de la troisième situation pour laquelle le diagnostiqueur se situe à un instant t_r d'isolation d'un défaut ; la propriété normale P_{F_0} a donc été invalidée à l'instant $\mathrm{t}_r - h$, et le diagnostiqueur d'une part enregistre le vecteur $v_{obs}(\mathrm{t}_r)$ pour créer le comportement observé $Obs(\mathrm{t}_r)$, puis d'autre part vérifie toutes les propriétés P_F pour ce comportement observé $Obs(\mathrm{t}_r)$ pour tous les défauts diagnosticables $F \in \Gamma^{Diag}$. Remarquons qu'il pourra arriver, même toujours lorsque que le système fonctionnera normalement, que les quantités de stockage et de puissance de calcul nécessaires au fonctionnement et établies dans le pire des cas, ne soient pas entièrement exploitées.

5.2.3 Complexité relativement à une caractérisation de défauts

L'étude de la complexité de fonctionnement du diagnostiqueur doit se réaliser relativement à une caractérisation de défauts Λ. Selon cette caractérisation Λ utilisée dès le début de la chaîne de conception, la solution d'implémentation de l'algorithme de fonctionnement du diagnostiqueur sera différente d'une autre solution d'implémentation suivant une autre caractérisation, et ainsi ces valeurs de complexité seront elles aussi différentes. En effet chaque caractérisation n'étant pas élaborée suivant le même formalisme, elles n'auront donc pas les mêmes besoins de stockage et de puissance de calcul.

Pour une caractérisation de défauts Λ utilisée, il faut donc déterminer le nombre n_{ce}^{Λ} de calculs élémentaires à réaliser, ainsi que le nombre n_{de}^{Λ} de données élémentaires à stocker, lors du fonctionnement du diagnostiqueur dans le pire des cas. Ces deux nombres, obtenus pour un fonctionnement dans le pire des cas à chaque instant $\mathrm{t}_r \in \mathbb{T}_\iota$ du temps du diagnostiqueur, permettent donc de déterminer les capacité d'implantation du diagnostiqueur : la période de cadencement du processeur pour accomplir les n_{ce}^{Λ} calculs élémentaires à chaque instant de fonctionnement (dans le pire des cas), ainsi que la taille de la mémoire pour stocker les n_{de}^{Λ} données élémentaires.

Concernant les deux caractérisations que nous avons présentées et comme nous n'avons pas présenté leurs implémentations, nous ne ferons donc pas explicitement leurs études de complexité. Pour la caractérisation parfaite, nous n'en avons pas tenté une implémentation car comme nous l'avons préalablement dit, elle risque d'une part de ne pas être exploitable pour des systèmes complexes et elle n'est d'autre part pas optimisée pour l'architecture considérée du système complet. Il faudrait en effet sauvegarder tous les comportements observables bornés et dans le pire des cas comparer le comportement réellement observé du système à tous ces comportements observables bornés enregistrés. Nous verrons dans la partie suivante traitant du cas d'étude que l'espace de stockage nécessaire est trop important pour pouvoir être utilisé en embarqué.

Pour la caractérisation par formules temporelles, nous utilisons l'outil ARTiMon© pour vérifier les satisfactions des formules par le comportement réellement observé du système. Sans rentrer dans les détails de fonctionnement de cet outil, notons que sa complexité de fonctionnement peut être obtenue dans [Zei09]. Indiquons juste que comme la sémantique des formules est représentée en machine par des listes d'intervalles, la seule quantité qui évolue en fonctionnement est donc le nombre d'intervalles. Pour la complexité en espace, un calcul grossier permet d'obtenir une complexité pour chaque formule de $n \cdot (h+1) \cdot 2^h$ intervalles dans le pire des cas (où n dépend de la période d'échantillonnage ι du temps $\mathbb{T}_\iota$ ainsi que de la longueur du plus grand intervalle de la formule considérée, et h représente la hauteur de l'arbre, binaire dans le pire des cas, de la formule considérée) ; néanmoins un algorithme *ad-hoc* permet un calcul du nombre d'intervalles au plus près suivant les formules considérées. Pour la complexité en temps, l'algorithme est linéaire sur l'analyse total d'un comportement observable et est constant à chaque instant $\mathrm{t}_r \in \mathbb{T}_\iota$ de temps avec un facteur de l'ordre de 1 à 2 traitements d'intervalles. Cela signifie 1 à 2 transformations d'intervalles suivant l'analyse, où une transformation d'intervalle correspond à des opérations sur les deux seuils minimum et maximum de l'intervalle et une adjonction (ajout du nouvel intervalle dans la liste ou fusion avec un intervalle de la liste).

5.3 Application sur le cas d'étude

Nous allons mettre en application ce passage au diagnostiqueur sur le cas d'étude. Suite aux rappels des variables, paramètres et défauts diagnosticables de la ligne d'air, nous allons déterminer les valeurs exactes des complexités suivant les deux caractérisations. Nous présenterons uniquement l'espace de stockage nécessaire pour la caractérisation parfaite, afin de montrer son impraticabilité pour l'embarqué, alors que nous présenterons le temps et l'espace de stockage pour la caractérisation par formules temporelles, pour laquelle nous avons utilisé l'outil ARTiMon©. Nous obtiendrons ensuite le diagnostiqueur de cette ligne d'air généré suivant cette caractérisation par formules temporelles.

5.3.1 Rappels des variables, paramètres et défauts diagnosticables de la ligne d'air

$\overline{\mathrm{V}}_{obs} = \{c_Q; c_P; u_\omega; u_x; y_Q; y_P; u_\omega^M; u_x^M; y_Q^M; y_P^M\}$ est l'ensemble des variables observables de cette ligne d'air. La période d'échantillonnage ι du temps $\mathbb{T}_\iota$ vaut 0,01 seconde. Enfin, la borne b de diagnostic vaut 6 unités de temps.

Rappelons que l'ensemble des défauts diagnosticables est différent selon la caractérisation de défauts Λ considérée durant l'étude de diagnosticabilité. Pour l'étude suivant la caractérisation parfaite, l'ensemble des défauts diagnosticables est $\Gamma_{AirLine}^{Diag} = \{F_{Norm}; F_{LockCmpr}; F_{SenQ}; F_{SenP}\}$, où F_{Norm} représente le fonctionnement normal, $F_{LockCmpr}$ représente le défaut de blocage du compresseur, F_{SenQ} et F_{SenP} représentent respectivement les défauts de mesure des capteurs de débit et de pression. Pour l'étude suivant la caractérisation par formules temporelles, l'ensemble des défauts diagnosticables est $\Gamma_{AirLine}^{Diag} = \{F_{Norm}; F_{SenP}\}$. Néanmoins et comme nous l'avons signalé lors de cette étude suivant cette caractérisation par formules temporelles, les défauts $F_{LockCmpr}$ de blocage du compresseur et F_{SenQ} de mesure du capteur de débit sont diagnosticables pour des évolutions de la consigne c_Q de débit uniquement dans les parties $C_Q^2 = [3; 7]$, $C_Q^3 = [8; 27]$ et $C_Q^4 = [28; 30]$. Nous allons donc nous restreindre, pour cette caractérisation par formules temporelles, à une consigne c_Q de débit évoluant dans le domaine $C_Q = \{3; 4; \ldots; 30\}$. Nous pouvons ainsi considérer l'ensemble suivant des défauts diagnosticables : $\Gamma_{AirLine}^{Diag} = \{F_{Norm}; F_{LockCmpr}; F_{SenQ}; F_{SenP}\}$.

5.3.2 Résultats de complexité en espace suivant la caractérisation parfaite

Concernant la caractérisation parfaite et si nous voudrions l'utiliser, il nous faudrait d'abord trouver une solution d'implémentation. Notons que nous n'avons pas tenté de l'implémenter car comme nous l'avons préalablement dit, elle risque d'une part de ne pas être exploitable pour des systèmes complexes et elle n'est d'autre part pas optimisée pour l'architecture considérée du système complet. Afin justement de rendre compte de son inexploitabilité, nous allons calculer, suivant une solution naïve d'implémentation, la capacité de stockage nécessaire en considérant néanmoins de ne sauvegarder des données que de la boucle réelle et pas celles de la boucle modèle.

Suivant cette caractérisation parfaite, le fonctionnement du diagnostiqueur se résume à rechercher un élément dans une base de données. En effet, en fonctionnement et dans le pire des cas, le diagnostiqueur doit comparer le comportement observé du système à tous les comportements observables bornés, des ensembles $ObsBeh_{Cons}^{Bd(6)}(F)$ préalablement enregistrés pour chacun des défauts diagnosticables $F \in \Gamma^{Diag}$. Il est donc nécessaire de connaitre le nombre de comportements observables bornés afin de pouvoir déterminer l'espace mémoire nécessaire, ceci car chaque comportement observable borné est constitué du même nombre de données. Bien entendu, il s'agit d'une solution d'implémentation naïve dans le sens où nous ne faisons pas de recoupement de données entre plusieurs comportements observables.

Ensemble des instructions et des occurrences des défauts diagnosticables Le domaine des valeurs de la consigne c_Q étant restreint à l'ensemble $C_Q = \{3; 4; \ldots; 30\}$, nous restreignons aussi l'ensemble $Cons_{AirLine}$ des instructions à celles évoluant uniquement dans ce domaine. Il y a donc 28 instructions « statiques » (i.e. : de la forme $cs_{(a,a)}$ avec $a \in \{3; 4; \ldots; 30\}$) et 756 instructions « dynamiques » (i.e. : de la forme $cs_{(a,b)}$ avec $a, b \in \{3; 4; \ldots; 30\}$ tels que $a \neq b$).

Pour chacun les défauts diagnosticables $F \in \Gamma_{AirLine}^{Diag} \setminus \{F_{Norm}\}$ et pour les 28 instructions statiques $cs_{(a,a)} \in Cons_{AirLine}$, l'ensemble d'occurrences $\Omega_{(F,cs_{(a,a)})}$ est de cardinal 1, car ne contenant que l'unique occurrence $\mathrm{t}_n = 40$, alors que pour les 756 instructions dynamiques $cs_{(a,b)} \in Cons_{AirLine}$, $\Omega_{(F,cs_{(a,b)})}$ est de cardinal 11, car contenant les instants $\mathrm{t}_n = 35$ à $\mathrm{t}_n = 45$.

Nombre de comportements observables bornés Nous allons calculer le nombre de comportements observables bornés normaux ou sous la présence d'un défaut diagnosticable.

Pour chacun des trois défauts diagnosticables $F \in \{F_{LockCmpr}; F_{SenQ}; F_{SenP}\}$, chaque ensemble $ObsBeh_{Cons}^{Bd(6)}(F)$ contient au total 5 006 400 comportements observables bornées sous la présence de F. En effet, il y a d'une part 28 instructions statiques ayant chacune une seule occurrence potentielle de F, ce qui fait 28 comportements observables statiques, et d'autre part 756 instructions dynamiques ayant chacune 11 occurrences potentielles de F, ce qui fait 8 316 comportements observables dynamiques. Cela fait donc 8 344 comportements observables sous la présence de F ayant chacun $\frac{6}{0.01}$ restrictions temporelles potentielles (il faut enregistrer les restrictions de l'occurrence t_n de F jusqu'à la borne $b = 6$ de diagnostic).

Pour le cas normal F_{Norm}, l'ensemble $ObsBeh_{Cons}^{Bd(6)}(F_{Norm})$ contient au total 756 028 comportements observables bornées normaux. Il y a d'une part 28 instructions statiques ayant chacune une seule restrictions temporelles potentielles, car le système est stationnaire durant tout le domaine temporelle du comportement observable, ce qui fait 28 comportements observables bornés normaux statiques. Il y a d'autre part 756 instructions dynamiques ayant chacune $\frac{10}{0.01}$ restrictions temporelles potentielles, qui commencent à partir de l'instant $t_c = 30$ où la consigne évolue de a vers b et termine suite à la dynamique de réponse $\beta = 3$ du système et de la borne $b = 6$ de diagnostic, ce qui représente donc 756 000 comportements observables bornés normaux dynamiques.

En faisant le total pour le cas normal F_{Norm} ainsi que les trois défauts $F_{LockCmpr}$, F_{SenQ} et F_{SenP}, il y a donc 5 762 428 comportements observables bornés.

Complexité en espace Rappelons qu'ils faut calculer le nombre de données élémentaire à sauvegarder pour constituer la base de données des comportements observables bornés.

Un comportement observable borné est constitué des valeurs des variables observables de la ligne d'air sur une longueur temporelle $b = 6$. Il y a 6 variables observables, car nous ne considérons que la boucle réelle et pas la boucle modèle, à multiplier par $\frac{6}{0.01}$ instants de temps, ce qui fait 3 600 données élémentaires par comportement observable borné. Comme nous venons de calculer qu'il y a 5 762 428 comportements observables bornés sous la présence d'un défaut diagnosticable $F \in \Gamma_{AirLine}^{Diag}$, il y a donc 20 744 740 800 données élémentaires à sauvegarder pour constituer la base de données.

Afin de rendre compte de l'énormité du nombre de données élémentaires, nous allons faire une conversion de cette valeurs. Comme ces 20 744 740 800 données élémentaires représentent des nombres et en supposant, par exemple, qu'ils soient codés sur 4 octets, cela représente au total 82 978 963 200 octets ; soit environ 78 gigaoctets de données à stocker, arrondi au gigaoctet supérieur et en considérant que 1 gigaoctet correspond à 1 073 741 824 octets. Bien qu'il s'agisse d'une solution naïve d'implémentation et qu'elle pourrait ainsi être optimisée, remarquons néanmoins que nous avons fait certaines hypothèses sur les profils d'instructions (i.e. : nous n'avons considéré que des instructions constantes par morceaux et évoluant dans des nombres entiers entre 3 et 30).

5.3.3 Résultats de complexité suivant la caractérisation par formules temporelles

Pour obtenir ces résultats de complexité suivant cette caractérisation par formules temporelles, il faut non seulement obtenir les valeurs pour le fonctionnement de l'outil ARTiMon©, mais aussi celles provenant de la simulation de la boucle embarquée.

5.3.3.1 Complexité de fonctionnement de l'outil ARTiMon©

Comme nous l'avons signalé lors de la partie théorique, la valeur du nombre d'intervalles peut être obtenu de manière fine par un algorithme *ad-hoc*. Cet algorithme nous a fourni 3008 intervalles pour la vérification des 4 formules φ_F des défauts diagnosticables $F \in \Gamma_{AirLine}^{Diag}$. Or un intervalle est donné par ses deux seuils minimum et maximum, implémentés par des variables de type double, et par les ouvertures ou fermetures de ces deux seuils, implémentés par des variables de type booléen. En considérant qu'une variable de type double est représentée sur 64 bits et que les deux variables de

type booléen peuvent être représentées sur un même mot de longueur 16 bits suivant une architecture 16 bits, il faut donc 144 bits par intervalle, donc une capacité mémoire d'environ 55 kilo-octets dans le pire des cas.

Par ailleurs, la complexité en temps est constante à chaque pas de temps avec un facteur de l'ordre de 1 à 2 traitements d'intervalles. Sans rentrer dans les détails et en supposant qu'il faille par exemple réaliser environs 50 calculs élémentaires pour chacune des formules $F \in \Gamma^{Diag}_{AirLine}$ à chaque instant du temps, cela signifie 200 calculs élémentaires car nous avons 4 formules. En considérant un processeur cadencé à 1 gigahertz, il effectue un calcul élémentaire toutes les 1 nanoseconde (car 1 gigahertz correspond à la fréquence d'horloge du processeur et que sa période, obtenue par l'inverse de la fréquence, correspond au temps nécessaire pour effectuer un calcul élémentaire) et il lui faut donc au total $0,002$ centiseconde pour effectuer ces 200 calculs élémentaires (en considérant la période d'échantillonnage du diagnostiqueur à 1 centiseconde, car nous avons considéré $\iota = 0,01$ seconde).

5.3.3.2 Complexité de fonctionnement de la simulation de la boucle embarquée

Pour obtenir les valeurs de complexité de fonctionnement de la simulation de la boucle modèle, nous avons eu une approche empirique qui a consisté à estimer la taille mémoire et le temps de calcul nécessaires suivant différentes approximations.

Concernant l'espace mémoire, nous avons utilisé l'outil de compilation RTW (Real-Time Workshop©) de MATLAB© pour générer du code C à partir du modèle embarqué Simulink©, puis pour compiler ce code C et fournir une application suivant la cible désirée, des applications en temps réel nous concernant. Pour la boucle modèle constitué de la régulation et du modèle embarqué, nous avons donc obtenu une application d'environ 50 kilo-octets.

Concernant la puissance de calcul, notre approche fut très « expérimentale » et demanderait d'être consolidée par la suite par une approche beaucoup plus rigoureuse. Nous avons effectué plusieurs simulations, sur un pc fixe, de cette boucle modèle puis fait une moyenne du temps de calcul sur un ensemble de pire cas : les 100 pires cas de 1000 simulations. Au final, notre simulateur a mis, en moyenne pour ces pires cas, $0,0005$ centiseconde pour simuler le modèle à chaque instant de temps $\mathbb{T}_{0.01}$. Il s'agissait d'un processeur Intel© Core 2 Duo cadencé à $2,4$ gigahertz qui met environ $0,42$ nanoseconde, c'est-à-dire $0,00000417$ centiseconde, pour effectuer un calcul élémentaire. Nous pouvons donc estimer qu'il y a en moyenne 120 calculs élémentaires à réaliser à chaque instant du temps $\mathbb{T}_{0.01}$. Pour revenir à un processeur cadencé à 1 gigahertz, comme pour calculer la complexité de l'outil ARTiMon©, nous pouvons estimer qu'il faudrait alors $0,0012$ centiseconde pour simuler cette boucle modèle à chaque instant due temps $\mathbb{T}_{0.01}$.

5.3.3.3 Complexité de fonctionnement globale

Au global et faisant les sommes des temps et espaces mémoire, cette caractérisation nécessiterait environ 105 kilo-octets d'espace mémoire et environ $0,0032$ centiseconde pour simuler le modèle et réaliser le traitement des formules (par ARTiMon©). Cette solution de diagnostic embarquant l'outil ARTiMon© peut donc être estimée comme une solution viable car ne consommant que peu d'espace mémoire et de puissance de calcul. Notons de plus que ces bornes en espace et temps ne sont que rarement atteintes car elles ne reflètent que le cas où le diagnostiqueur doit isoler un défaut, c'est-à-dire le pire des cas de fonctionnement.

5.3.4 Génération d'un diagnostiqueur suivant la caractérisation par formules temporelles

Nous avons paramétré l'outil ARTiMon© avec les différentes formules temporelles des défauts diagnosticables et l'avons intégré dans la partie pilotage de cette ligne d'air selon l'architecture présentée

à la figure 5.1 de la page 167. ARTiMon© reçoit donc le flux de données composé des valeurs des variables observables de la boucle réelle et de celles de la boucle modèle. Il surveille ce flux de données en vérifiant à chaque instant t du temps $\mathbb{T}_{0.01}$ que la formule normale φ_{Norm} soit valide. Lorsqu'elle devient invalide à un instant t_k du temps $\mathbb{T}_{0.01}$, il attend le délai h d'isolation, qui vaut 2 secondes, pour conclure sur le défaut apparu. À l'instant d'isolation $t_k + 2$ du temps $\mathbb{T}_{0.01}$, il analyse donc les validités de la formule normale φ_{Norm} et des formules des défauts $\varphi_{LockCmpr}$, φ_{SenQ} et φ_{SenP}.

- Si aucune de ces trois formules de défaut n'est valide et que la formule normale est toujours invalide, il conclut à une perturbation ou à un défaut non-répertorié et envoie cette information au module de décision.
- Si au moins deux de ces formules de défaut sont valident et que la formule normale est toujours invalide, il conclut aussi à une perturbation ou à un défaut non-répertorié et envoie cette information au module de décision.
- Si aucune de ces trois formules de défaut n'est valide mais que la formule normale est valide, il conclut à une perturbation ou à un défaut transitoire et recommence à vérifier la formule normale.
- Si aucune une de ces trois formules de défaut est valide mais que la formule normale est aussi valide, il conclut à une perturbation ou à un défaut non-répertorié et envoie cette information au module de décision.
- Si une unique formule φ_F de défaut est valide et que la formule normale est toujours invalide, il conclut donc au diagnostic du défaut F et envoie cette information au module de décision.

5.3.5 Tests du diagnostiqueur de la ligne d'air

Pour tester le diagnostiqueur de la ligne d'air, nous avons élaboré une instruction aléatoire puis déterminé des occurrences aléatoires des trois défauts diagnosticables $F_{LockCmpr}$, F_{SenQ} et F_{SenP}. À chacun des tests, le défaut fut diagnostiqué sans ambiguïté. Nous avons par ailleurs fait apparaître le défaut F_{LockEV} de blocage de l'électrovanne afin de vérifier que lors de défauts non répertoriés (i.e. : non pris en compte par le diagnostiqueur car non diagnosticables), le diagnostiqueur concluait bien à une perturbation ou un défaut non-répertorié.

La figure 5.2 de la page 174 représente une simulation de cette ligne d'air avec l'occurrence du défaut de blocage du compresseur à l'instant du temps $t_n = 42$. Les quatre premiers graphiques représentent, comme à chaque fois que nous avons présenté la simulation d'un comportement observable, les évolutions des variables observables de cette ligne d'air. Le cinquième graphique représente l'occurrence $t_n = 42$ de ce défaut de blocage du compresseur. Les quatre derniers graphiques représentent les validités des formules des défauts : le cas normal F_{Norm} pour le sixième graphique, le défaut $F_{LockCmpr}$ de blocage du compresseur pour le septième graphique, le défaut F_{SenQ} de mesure du capteur de débit pour le huitième graphique et le défaut F_{SenP} de mesure du capteur de pression pour le neuvième graphique. Notons que pour le cas normal F_{Norm}, il s'agit de la négation de la formule normale : la validité de la formule $\neg\varphi_{Norm}$. En effet, ARTiMon© fut à la base élaboré pour détecter la violation d'exigences (élaborées sous forme de formules temporelles) d'un système par analyse du flux de données observables de ce système. Il s'agit donc de définir une exigence et de vérifier qu'elle ne soit jamais violée. Ainsi dans notre cas, nous recherchons d'abord à ce que la formule normale φ_{Norm}, exprimant le bon fonctionnement du système, ne soit jamais violée et nous vérifions ainsi la négation de cette formule.

5.4 Conclusion sur la génération du diagnostiqueur

Dans ce chapitre, nous venons de présenter comment générer un diagnostiqueur d'un système piloté. Nous avons indiqué que ce diagnostiqueur, issu d'une étude préalable de diagnosticabilité suivant une

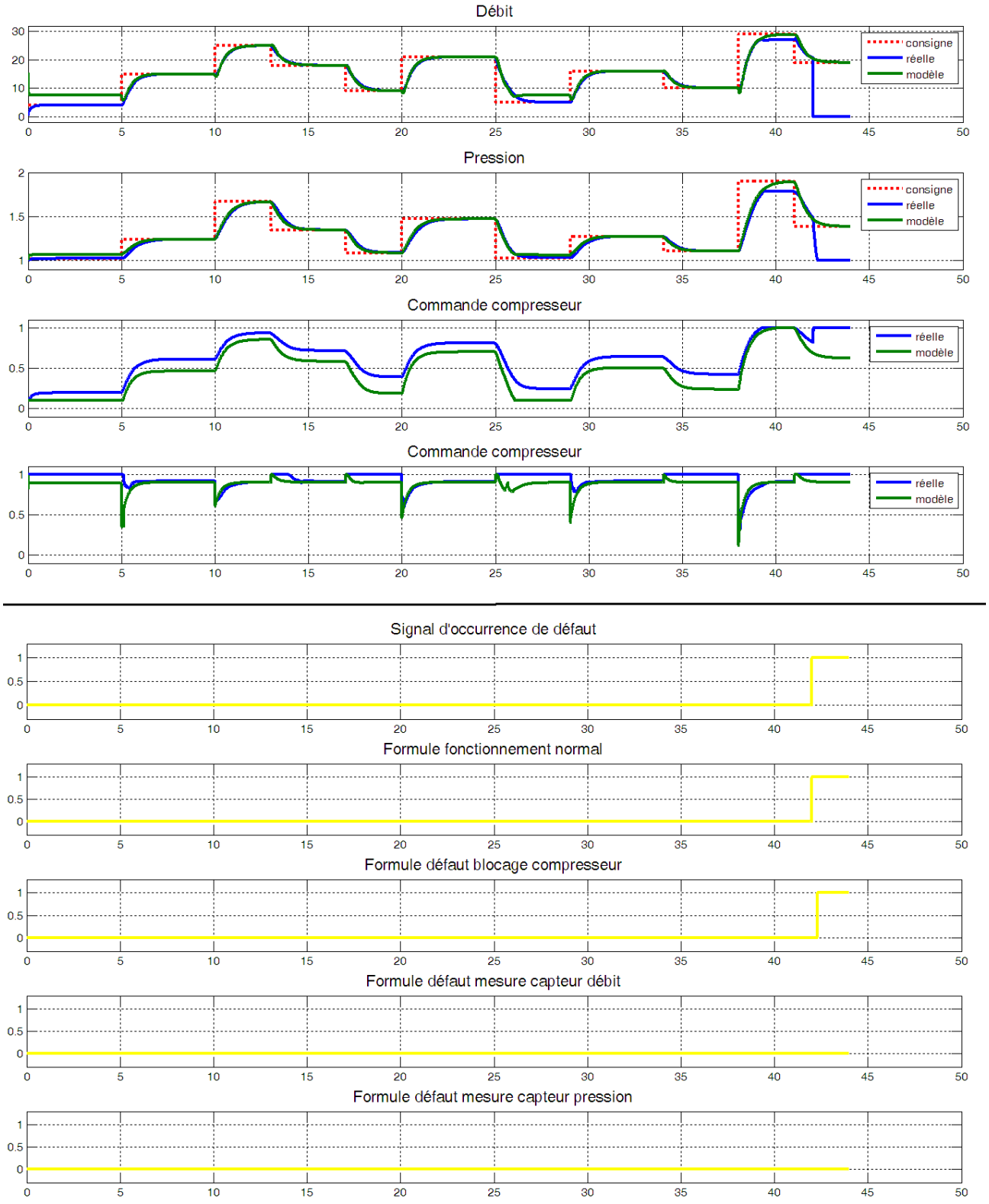

Figure 5.2 – Simulation test du diagnostiqueur avec un blocage du compresseur.

caractérisation de défauts, s'intègre directement dans le système de pilotage afin de surveiller en temps réel le comportement observé du système (i.e. : le flux de données constitué des valeurs des variables

observables à chaque instant du temps).

Nous avons décrit l'algorithme de fonctionnement d'un diagnostiqueur suivant une caractérisation de défauts ayant servi à cette étude préalable de diagnosticabilité. En fonctionnement et à chaque instant du temps, le diagnostiqueur vérifie que le comportement observé du système valide la propriété normale. Si oui, cela signifie donc que le système fonctionne normalement. Dans le cas contraire où ce comportement observé ne valide pas la propriété normale, il y a donc détection d'un mauvais fonctionnement du système et le diagnostiqueur doit alors rechercher quelle propriété d'un défaut diagnosticable est valide.

Nous avons dû, par ailleurs, réaliser une étude de complexité en temps de calcul et en espace mémoire de l'algorithme de fonctionnement d'un diagnostiqueur issu d'une caractérisation de défauts. En effet, nous avions jusqu'alors supposé être dans un cas idéal où le diagnostiqueur a les capacités de stockage et de puissance de calcul nécessaires pour effectuer toutes les vérifications. Or, en présentant la caractérisation parfaite au chapitre 4 d'étude de la diagnosticabilité, nous avions déjà remarqué que cette caractérisation risquait d'être inexploitable du fait du nombre important de données à sauvegarder et à comparer. Ainsi et avant toute implémentation d'un diagnostiqueur suivant une caractérisation de défauts, il convient donc de s'assurer que son implantation permette son fonctionnement : c'est-à-dire que les ressources disponibles en stockage et puissance de calcul soient compatibles avec la caractérisation considérée.

Enfin, nous avons appliqué cette génération d'un diagnostiqueur au cas d'étude. Nous avons considéré un diagnostiqueur suivant la caractérisation par formules temporelles en utilisant pour cela l'outil ARTiMon©. Nous avons alors intégré les formules des défauts conclus diagnosticables au chapitre précédent, puis avons effectué divers tests faisant apparaître autant ces défauts diagnosticables que d'autres défauts non-diagnosticables ; ceci afin de tester la robustesse du diagnostiqueur. Les résultats sont probants avec un fonctionnement en accord pour les contraintes temps réel et embarqué.

Avec ce chapitre, nous venons de finaliser la réalisation d'une chaîne de conception d'un diagnostiqueur appliquée aux systèmes technologiques pilotés. Le chapitre suivant va faire une conclusion générale de l'ensemble des travaux réalisés et ouvrira ensuite sur les perspectives potentielles à suivre pour de futurs travaux.

Conclusion générale et perspectives

Les travaux présentés dans ce mémoire furent le résultat d'un projet collaboratif réunissant l'entreprise Sherpa Engineering, le Laboratoire de Recherche en Informatique (LRI) unité mixte de recherche (UMR8623) de l'Université Paris-Sud et du CNRS et enfin le Laboratoire d'Ingénierie des Systèmes Embarqués (LISE) du CEA LIST. Ces travaux eurent pour objectif de définir une chaîne de conception outillée d'un diagnostiqueur, suivant un cadre théorique commun, appliquée aux systèmes technologiques pilotés. En conclusion de ce mémoire, nous allons d'abord rappeler l'ensemble des travaux réalisés en mettant en avant les différents apports, puis nous discuterons de quelques points pour lesquels d'intéressantes perspectives peuvent être mises en œuvre pour de futurs travaux complémentaires.

Apports de la thèse

Comme nous l'avions indiqué en introduction de ce mémoire, il est difficile de trouver une solution méthodologique générique permettant la conception et la réalisation d'un diagnostiqueur d'un système. Nous avions en effet remarqué que peu de théories générales existent bien que le diagnostic soit une discipline relativement ancienne avec de nombreuses méthodologies développées par différentes communautés, ce que nous avons présenté au chapitre 1. Nous avions alors remarqué que beaucoup d'approches sont plus *ad-hoc* sans démarche systématique ni générique.

Nous avions ainsi estimé qu'il serait utile de définir une chaîne de conception d'un diagnostiqueur appliquée aux systèmes technologiques pilotés. Nous avions par ailleurs indiqué que pour être exploitable, cette conception du diagnostiqueur doit être basée sur le résultat d'une étude préalable de diagnosticabilité (i.e. : s'assurer dès l'étape de conception du diagnostiqueur qu'il sera toujours capable de diagnostiquer sans ambiguïté les défauts préalablement répertoriés), elle-même préalablement basée sur une identification des défauts potentiels du système. Cette chaîne de conception se devant donc d'être non seulement cohérente, ce qui signifie qu'elle doit être définie suivant un cadre théorique commun, mais aussi complète en intégrant les étapes de représentation du système et des défauts permettant l'étude de la diagnosticabilité ainsi que la génération du diagnostiqueur associé à cette étude de diagnosticabilité.

Nous avions aussi jugé judicieux d'exploiter les méthodes et outils habituellement utilisés dans le diagnostic ainsi que dans la conception des systèmes. Nous avons pour cela considéré d'une part les méthodologies de diagnostic à base de modèles pour lesquelles nous avons voulu rajouter un aspect temporel dans la comparaison entre le comportement réellement observé du système, obtenu des variables observables, et le comportement prédit par un modèle embarqué et simulé par le diagnostiqueur. Nous avons par ailleurs considéré les méthodes de conception des systèmes basées sur l'utilisation d'un ensemble cohérent de modèles génériques représentant le système, ce qui a fourni notre cadre théorique commun.

De la représentation du système à l'intégration des défauts

Un système technologique piloté, présenté au chapitre 2, est un système mécatronique pour lequel nous distinguons particulièrement la partie pilotage, que nous avons nommée *système de pilotage*, de

la partie opérante, que nous avons nommée *système opérant*. Nous avons justifié cette représentation par le fait que, bien que la majorité des travaux de diagnostic considère uniquement la partie opérante, ces systèmes sont généralement bouclés avec leur partie pilotage afin d'accroître leurs performances et de les maintenir en dépit d'entrées inconnues pouvant les affecter. Le diagnostic des défauts est dans ce contexte plus délicat du fait des objectifs contradictoires entre la commande, qui cherche à minimiser, voire annuler, les effets des perturbations et des défauts, et le diagnostic qui, quant à lui, cherche justement à mettre en évidence ces défauts.

Comme nous avons par ailleurs souhaité reprendre l'idée de fonctionnement d'un diagnostiqueur basé sur les méthodologies de diagnostic à base de modèles : c'est-à-dire la comparaison du comportement réellement observé du système, obtenu des variables observables, à un comportement prédit par un modèle embarqué et simulé par le diagnostiqueur. Nous avons donc défini une structure dite *système complet* et composée de deux boucles alimentées par la consigne venant de l'opérateur : la boucle réelle composée du système opérant et de son système de pilotage, la boucle modèle composée d'un modèle embarqué du système et de sa propre partie pilotage (i.e. : une duplication des lois de commande de la partie réelle).

Nous avons utilisé les outils classiques de modélisation que sont la représentation par espace d'état avec une évolution temporelle en temps discret. Cette représentation fut privilégiée pour sa simplicité à nous permettre d'intégrer les défauts potentiels du système : c'est-à-dire d'intégrer directement les équations des défauts dans les équations du modèle de bon fonctionnement du système. Par ailleurs, c'est aussi grâce à cette représentation que nous avons été à même de définir simplement une notion de diagnosticabilité basée sur une analyse de traces d'observation du système, ce que nous indiquerons par la suite. Nous avons de ce fait considéré un modèle dit « parfait » représentant le système opérant de la boucle réelle et un modèle embarqué de la boucle modèle ; les deux systèmes de pilotage des deux boucles étant eux aussi intégrés dans cette modélisation.

Concernant l'ensemble des défauts, nous nous sommes rendu compte que la majorité des travaux de la littérature n'indique pas clairement ce qui définit les défauts ainsi que les moyens de les représenter suivant leur diversité.

Toujours dans le but de définir une chaîne de conception complète, il a d'abord fallu déterminer ce qui définit les défauts. Une étude de sûreté de fonctionnement permet de réaliser cette étape de définition. Tous les défauts potentiels d'un système considéré sont inventoriés et classifiés selon un indice de criticité, établi suivant le risque potentiel du défaut sur le fonctionnement du système, son intégrité ou celle de son environnement. Suivant cet indice de criticité, l'ensemble des défauts devant être pris en compte par un outil de diagnostic est alors défini.

Par la suite et en analysant l'ensemble des caractéristiques définissant les défauts dans les travaux classiques de la littérature, nous avons pu constater qu'ils peuvent être décrits suivant deux traits caractéristiques génériques : le comportement d'un défaut ainsi que son effet sur le système. Nous avons alors défini une typologie de défauts, basée sur ces deux caractéristiques génériques, qui nous a permis de les intégrer dans le modèle de bon fonctionnement du système par modification des équations du fonctionnement normal. Suivant l'architecture de diagnostic considérée (i.e. : le système complet composé d'une boucle réelle et d'une boucle modèle) nous avons bien sûr intégré ces défauts uniquement dans le modèle parfait du système et non dans le modèle embarqué. Enfin nous avons construit une bibliothèque de défauts, implémentée dans l'environnement MATLAB/Simulink©, basée sur cette typologie et permettant en pratique d'intégrer les défauts potentiels directement dans un modèle de simulation d'un système.

Cette première partie de travaux nous a ainsi permis de définir tous les modèles du système : le modèle de bon fonctionnement, aussi nommé le *cas normal*, et les modèles des défauts répertoriés durant l'étude de sûreté de fonctionnement du système. Ces modèles nous ont ensuite été utiles dans

l'étude de la diagnosticabilité.

De l'étude de la diagnosticabilité à la génération du diagnostiqueur

L'étude de la diagnosticabilité consiste à s'assurer, lors de l'étape de conception, qu'un diagnostiqueur sera toujours capable de diagnostiquer sans ambiguïté les défauts préalablement répertoriés. Or comme nous l'avons remarqué, bien que l'ensemble des travaux réalisés concernant l'étude de la diagnosticabilité reprennent les méthodologies du diagnostic, le lien entre cette étude et le passage au diagnostiqueur n'est pas toujours clairement indiqué.

Nous avons ainsi établi la correspondance entre cette étude de la diagnosticabilité et le passage au diagnostiqueur : ce sont à partir des règles d'analyse du diagnostiqueur que doit se mener l'étude de la diagnosticabilité du système. Ces règles d'analyse, que nous avons nommées une *caractérisation de défauts*, est une famille de propriétés (une propriété pour chacun des défauts répertoriés en incluant aussi le cas normal) liant les variables observables du système.

L'étude de la diagnosticabilité, basée sur une caractérisation de défauts, consiste alors à vérifier la validité des différentes propriétés de cette caractérisation par les différents comportements observables du système sous la présence des défauts, cas normal inclus. Nous avons défini la diagnosticabilité de chacun des défauts répertoriés, y compris la diagnosticabilité du cas normal. Le cas normal est diagnosticable si tout comportement observable normal valide toujours la propriété normale à tout instant du temps. Un défaut quelconque est diagnosticable si, dès qu'il apparaît et dans une fenêtre temporelle bornée par une certaine borne de diagnostic, la propriété normale devient invalide (ce qui implique qu'elle était préalablement valide) puis suite à un certain délai d'isolation et durant un certain délai de confiance, seule la propriété du défaut concerné reste toujours valide (ce qui implique que les autres propriétés sont invalides durant cette fenêtre temporelle). Cela signifie que tout comportement observable sous la présence de ce défaut, à une certaine occurrence (i.e. : à un certain instant du temps où il apparaît), invalide la propriété normale puis, suite au délai d'isolation, ne valide que la propriété du défaut considéré durant un certain délai de confiance.

Cette étude se réalise donc par analyse de la validité des différentes propriétés, d'une caractérisation de défauts considérée, par les comportements observables du système. Or un comportement observable du système représente l'évolution des variables observables suivant une certaine instruction de l'opérateur (i.e. : une suite de valeurs de la variable de la consigne). Il est obtenu par restriction, uniquement sur les variables observables, d'un comportement du système qui, lui, est obtenu par l'évolution des valeurs de toutes les variables du modèle, lors du fonctionnement suivant cette instruction et sous la présence ou non d'un défaut à une certaine occurrence. Nous avons donc supposé avoir un ensemble d'instructions permettant de rendre compte de tous les fonctionnements possibles du système, ainsi que d'ensembles d'occurrences de chacun des défauts permettant, eux aussi, de rendre compte des effets des défauts sur le fonctionnement du système.

Une caractérisation de défauts rend compte du fonctionnement du diagnostiqueur, c'est-à-dire ses règles d'analyse, pour détecter et isoler les défauts potentiels du système. Ce sont des propriétés, une pour chaque défaut répertorié incluant aussi le fonctionnement normal, établies suivant le même formalisme et liant les variables observables du système. L'invalidité de la propriété normale signifie la détection d'un défaut et la validité d'une unique propriété d'un défaut signifie son isolation. Nous avons défini deux caractérisations de défauts :

- La *caractérisation parfaite* est établie suivant un formalisme ensembliste et considère les comportements observables bornés sur de petites fenêtres temporelles de même longueur pour former les ensembles de comportements observables bornés sous la présence ou non d'un défaut. La propriété d'un défaut quelconque consiste à s'assurer qu'un comportement observable borné appartienne à l'ensemble des comportements observables bornés sous la présence du défaut considéré par la propriété. Cette caractérisation permet de s'assurer de manière intrinsèque de la diagnos-

ticabilité d'un défaut : c'est-à-dire qu'un défaut non diagnosticable suivant cette caractérisation signifie soit que les paramètres d'étude (les différents délais et bornes) sont trop répressifs (trop courts ou trop longs), soit que les modèles en eux-mêmes ne sont pas assez « observables » (i.e. : il n'y a pas assez de capteurs pour rendre compte de l'effet d'un défaut sur le fonctionnement observable du système).

– La *caractérisation par formules temporelles*, établie suivant un formalisme de logique temporelle, considère des formules temporelles décrivant l'évolution temporelle des comparaisons des variables observables de la boucle réelle et de la boucle modèle, suivant le fonctionnement sous la présence ou non d'un défaut répertorié. Ces formules sont évaluées par n'importe quels comportements observables du système, et la propriété d'un défaut quelconque consiste à s'assurer qu'un comportement observable valide la formule temporelle du défaut considéré par la propriété. Contrairement à la caractérisation parfaite, la non-diagnosticabilité d'un défaut, suivant cette caractérisation par formules temporelles, peut néanmoins être due aux propriétés en elles mêmes qui ne sont pas assez expressives.

Rappelons que nous souhaitions reprendre les méthodologies de diagnostic à base de modèles pour lesquelles nous avons voulu rajouter un aspect temporel dans la comparaison entre le comportement réellement observé du système, obtenu des variables observables, et le comportement prédit par un modèle embarqué et simulé par le diagnostiqueur. Ainsi les deux caractérisations présentées remplissent cette exigence. Néanmoins, nous avons remarqué que pour un passage au diagnostiqueur, la caractérisation parfaite n'est d'une part pas optimisée par l'architecture considérée (les deux boucles réelle et modèle), mais qu'elle risque en plus d'être impraticable pour des systèmes complexes. Elle est par contre très utile pour la conception du fait de l'assurance que la non-diagnosticabilité d'un défaut ne peut venir que des paramètres d'étude ou du « niveau d'observation » des modèles en eux-mêmes. La caractérisation par formules temporelles, quant à elle, présente l'avantage d'être facilement implémentable en solution de diagnostic par l'utilisation de l'outil ARTiMon© du CEA/LIST, comme nous l'avons montré dans le chapitre 5 de génération du diagnostiqueur.

La génération du diagnostiqueur issue d'une étude de la diagnosticabilité, et donc suivant une caractérisation de défauts considérée, est alors simple à implémenter, sous réserve que la complexité de fonctionnement de cette caractérisation le rende praticable pour le système concerné. Il suffit en effet d'intégrer les différentes propriétés de cette caractérisation, qui représentent les règles d'analyse, dans l'algorithme générique de fonctionnement du diagnostiqueur. Cette simplicité est établie par la mise en correspondance, dès le début, de l'étude de la diagnosticabilité avec le fonctionnement du diagnostiqueur : c'est-à-dire suivant une caractérisation de défauts qui représente les règles d'analyse du diagnostiqueur.

Résultat : une chaîne de conception cohérente et complète

En pratique, l'obtention de tous ces comportements observables ainsi que la définition des propriétés des caractérisations (i.e. : leur « apprentissage »), mais aussi toute l'étude de la diagnosticabilité ainsi que la génération du diagnostiqueur associé, peuvent se réaliser en utilisant des outils de simulation ; ce qui nécessite donc d'implémenter au préalable les différents modèles dans l'outil.

Pour ce faire, nous avons utilisé l'outil MATLAB/Simulink© qui permet une implémentation aisée de modèles de systèmes et pour lequel nous avons développé une bibliothèque spécifique de défauts. Ces différents modèles, normal et sous la présence des défauts répertoriés, sont ainsi simulés afin de construire tous les comportements observables avec ou sans défaut. L'étude de la diagnosticabilité est elle aussi réalisée directement dans l'outil, par comparaison de comportements observables bornés pour la caractérisation parfaite, ou vérification de la satisfaction de formules temporelles, grâce à l'outil ARTiMon© directement interfacé à MATLAB/Simulink©, par les comportements observables pour la caractérisation par formules temporelles. Enfin la génération du diagnostiqueur, associé à l'étude de

diagnosticabilité suivant la caractérisation par formules temporelles (car la caractérisation parfaite est potentiellement impraticable), est elle aussi réalisée dans MATLAB/Simulink© par intégration de l'outil ARTiMon© et des différentes formules temporelles directement dans le système de pilotage.

Ces travaux réalisés forment donc bien une chaîne de conception cohérente et complète d'un diagnostiqueur. La cohérence apparaît bien dans l'élaboration d'un cadre théorique commun permettant de modéliser un système, d'y intégrer des défauts préalablement répertoriés, puis d'en étudier la diagnosticabilité afin d'en générer le diagnostiqueur associé. Cette cohérence est de plus bien garantie, en pratique, par l'utilisation d'un même outil durant toute la chaîne de conception. La complétude apparaît dans la méthodologie qui permet de réaliser le diagnostiqueur suivant différentes étapes successives, il s'agit donc d'une chaîne dite « tout-en-un ».

Discussion et perspectives

Tous ces travaux réalisés s'ouvrent à quelques discussions et perspectives pouvant être mises en œuvre pour de futurs travaux complémentaires.

La perfection du modèle parfait

L'approche d'étude de la diagnosticabilité a consisté à considérer une représentation « fidèle » du système opérant de la boucle réelle. Nous avons en effet considéré un modèle de bon fonctionnement de celui-ci que nous avons qualifié de « parfait ». Nous avons justifié de cette « perfection » par le fait qu'il représente le plus fidèlement possible le système opérant réel, c'est-à-dire qu'il intègre tous les multiples phénomènes physiques évoluant dans le système réel, y compris les non-linéarités potentielles.

Or il pourrait être utile de pouvoir déterminer au préalable à quel point ce modèle doit être « parfait ». C'est-à-dire de définir un « degré de fidélité » d'un modèle qui, au-dessus d'un certain niveau, nous assure de son utilisation pour la démarche de diagnostic que nous avons élaborée. Cela permettrait alors de réduire les temps de conception de ce modèle et ainsi de réduire la durée de conception du diagnostiqueur.

Les ensembles d'instructions et d'occurrences des défauts

L'étude de la diagnosticabilité s'est réalisée par analyse de propriétés d'une caractérisation de défauts par des comportements observables. Ces comportements observables ont été obtenus en considérant différentes instructions ainsi que différentes occurrences des défauts répertoriés. Sans plus de précision, nous avons supposé que ces ensembles d'instructions et d'occurrences permettaient d'obtenir l'ensemble de tous les fonctionnements possibles du système complet sous la présence ou non d'un défaut.

Il pourrait néanmoins être utile de définir une méthode permettant, selon le type de système et des défauts considérés (par exemple que des défauts brusques, permanents, etc.), de générer directement ces ensembles. Cela se justifie dans le but de ne pas prendre en compte toutes les évolutions potentielles des variables de la consigne ainsi que tous les instants possibles d'occurrences sur une fenêtre temporelle, ce que nous avons fait sur le cas d'étude.

Méthode générique de construction des formules temporelles des défauts

Nous avons présenté deux caractérisations de défauts dont la seconde, la caractérisation par formules temporelles, permet d'exprimer l'évolution des variables observables du système complet par des formules temporelles.

Nous avons défini une procédure permettant d'obtenir la formule temporelle du fonctionnement normal. Mais pour les différents défauts répertoriés, nous n'avons pu définir de procédures permettant

de les obtenir. Cela se justifie par le fait que nous avions d'abord élaboré cette idée d'utiliser un formalisme de logique temporelle pour ajouter un aspect temporel aux méthodologies de diagnostic à base de modèles. Nous avions ainsi déterminé une formule temporelle normale caractéristique de l'évolution temporelle des comparaisons des variables observables de la boucle réelle et de la boucle modèle ; la méthode générique découlait ainsi de cette idée. Pour les formules des défauts, nous avons défini des formules temporelles, liant uniquement les variables observables de la boucle réelle, en faisant une analyse au cas par cas des différents comportements observables.

Il pourrait être utile de définir de telles procédures en essayant de se baser sur le comportement du défaut pour une partie générique de la formule, puis de l'augmenter par l'étude de son effet sur le système complet : c'est-à-dire en considérant autant les variables observables de la boucle réelle que celles de la boucle modèle.

Le cas des défauts simultanés

Durant tous nos travaux, nous avons fait une hypothèse importante qui est la non-occurrence de plusieurs défauts en simultané. Nous avons en effet supposé que le délai entre les occurrences de deux défauts distincts est toujours supérieur à la borne de diagnostic. Ceci nous a donc permis de ne pas considérer les défauts simultanés.

Nous avons néanmoins pris soin d'élaborer le cadre théorique afin qu'il puisse convenir pour la prise en compte et l'étude des défauts simultanés. C'est-à-dire qu'en considérant des sous-ensembles de défauts, constitués d'un ou plusieurs défauts, et en considérant des ensembles d'occurrences adéquates (i.e. : des n-uplets d'occurrences pour un sous-ensemble de défauts), il est alors possible de reprendre l'étude de la diagnosticabilité selon la théorie proposée.

Il risque néanmoins de se poser un problème d'explosion combinatoire dans le sens où pour être complète, l'étude doit ainsi considérer l'ensemble des parties de l'ensemble de tous les défauts potentiels. Cela risque, encore une fois pour des systèmes complexes, d'alourdir l'étude de la diagnosticabilité.

Étude de la diagnosticabilité des défauts progressifs d'usure

Comme nous l'avons indiqué au chapitre 4 d'étude de la diagnosticabilité, pour le cas des défauts faiblement progressifs représentant des usures normales du système, il risque de ne pas y avoir de détection : c'est-à-dire un passage de valide à invalide de la propriété normale. En effet, pour représenter un défaut faiblement progressif, nous avions supposé que le comportement du système commence avec la présence du défaut à un certain niveau et qu'il atteigne le niveau désiré à une occurrence choisie. Or cette représentation risque de rendre ces défauts non-diagnosticables car la propriété normale risque d'être invalide dès le début du comportement observable.

Nous avions alors apporté une solution possible qui consistait à supposer que la propriété normale soit toujours valide avant l'occurrence du défaut. Or nous avions soulevé un problème « d'atteignabilité » de tels défauts diagnosticables. C'est-à-dire qu'en fonctionnement le diagnostiqueur risque de détecter un comportement anormal avant que le défaut atteigne le seuil considéré lors de l'étude de la diagnosticabilité. Au délai d'isolation, le diagnostiqueur risque alors de ne pas conclure sur ce défaut car n'étant pas au bon seuil. Pour parer ce problème, il faudrait donc considérer le niveau pour lequel la propriété normale passe de valide à invalide, ce qui est en pratique très difficile car cela ne se fait généralement pas à un instant précis du temps mais durant une certaine période temporelle où ce niveau oscille entre deux valeurs. Notons que cette solution fut présentée afin de rester dans le même cadre théorique avec les mêmes paramètres de temps et de diagnostic.

Nous pouvons considérer une autre solution qui reste dans ce même cadre théorique mais ne pouvant néanmoins pas se définir suivant les mêmes constantes de temps et paramètres de diagnostic. Ce serait ainsi un autre diagnostiqueur qui ferait non plus une vérification en temps réel de la validité des propriétés des défauts d'usure, mais en ferait une vérification périodique. Cette solution consisterait

à considérer de petits morceaux de comportements observables cumulés bout à bout pour différents niveaux de présence du défaut, ceci afin de refléter la progression du défaut. Par exemple k secondes en fonctionnement sans présence du défaut, puis k secondes en fonctionnement avec 1% de présence du défaut, puis k secondes en fonctionnement avec 2% de présence du défaut, et ainsi de suite avec k représentant le temps de vérification à chaque période. Il faudrait ainsi définir des formules temporelles adéquates suivant ces comportements observables, en étudier la diagnosticabilité, puis générer le diagnostiqueur associé qui devrait ainsi réaliser une surveillance périodique pour ces défauts qui seraient diagnosticables ainsi qu'une surveillance en temps réel pour les autres défauts diagnosticables. Notons que cette solution satisfait la remarque de [BJL$^+$90] concernant les défauts évolutifs qui doivent être surveillés de façon périodique et à un rythme adapté au processus de vieillissement du système pour permettre soit une réadaptation des fonctions de commande, soit une action de maintenance préventive appropriée.

Références bibliographiques

[AD94] R. Alur et D.L. Dill : A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[AFH96] R. Alur, T. Feder et T.A. Henzinger : The benefits of relaxing punctuality. *Journal of the ACM*, 43:116–146, 1996.

[AFI09] AFIS : Découvrir et comprendre l'ingénierie système. Rapport technique, AFIS - Association Française d'Ingénierie Système, 2009.

[AFN01] AFNOR : Maintenance terminology. Rapport technique, European standard, NF EN 13306, 2001.

[AGPV04] M. Arcak, H. Gorgun, L.-M. Pedersen et S. Varigonda : A nonlinear observer design for fuel cell hydrogen estimation. *IEEE Transaction on Control System Technology*, 12:101–110, 2004.

[AGRPC07] L. Antoni, X. Glipa, F. Roy et J.-P. Poirot-Crouvezier : *Pile à combustible GENEPAC*. Techniques de l'Ingénieur, 2007.

[ASC01] J.L. Alcock, L.C. Shirvill et R.F. Cracknell : Compilation of existing safety data on hydrogen and comparative fuels. Rapport technique, Shell Global Solutions, May 2001.

[Bas99] M. Basseville : On fault detectability and isolability. *Rapport de recherche IRISA no 1240*, 1999.

[BBF+01] B. Berard, M. Bidoit, A. Finkel, F. Laroussinie, A. Petit, L. Petrucci et P. Schnoebelen : *Systems and Software Verification - Model-Checking Techniques and Tools*. Springer-Verlag Berlin, 2001.

[BDF+10] M. Batteux, M. Donain, P. Fiani, S. Garnit, E. Noirtat et F. Roy : Energy management of a hybrid electric vehicle. *The International Electric Vehicle Symposium and Exposition, EVS 25,*, 2010.

[BDNR09] G. Bourgne, P. Dague, F. Nouioua et N. Rapin : Diagnosability of input output symbolic transition systems. *1st International Conference on Advances in System Testing and Validation Lifecycle, VALID'09, Porto, Portugal*, Septembre 2009.

[BDRF11a] M. Batteux, P. Dague, N. Rapin et P. Fiani : Diagnosability study of technological systems. *Modern Approaches in Applied Intelligence - 24th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2011, Syracuse, NY, USA, June 28 - July 1*, 2011.

[BDRF11b] M. Batteux, P. Dague, N. Rapin et P. Fiani : Faulty models of a fuel cell system for model-based diagnosis study. *the Fundamentals and Developments of Fuel Cells conference, FDFC, Grenoble, France, January 19-21*, 2011.

[BDTM02] E. Benazera, P. Dague et L. Travé-Massuyès : State tracking of uncertain hybrid concurrent systems. *International Workshop on Principles of Diagnosis, DX'02*, 2002.

[BFDR10] M. Batteux, P. Fiani, P. Dague et N. Rapin : Fuel cell system improvement for model-based diagnosis analysis. *IEEE Vehicle Power and Propulsion Conference, VPPC, Lille, France, September 1-3*, 2010.

[BFRD11] M. Batteux, P. Fiani, N. Rapin et P. Dague : Caractérisation du comportement observable d'un système pour l'étude de la diagnosticabilité de défauts. *Congrès international pluridisciplinaire en qualité et sûreté de fonctionnement, QUALITA, Angers, France, Mars 22-25*, 2011.

[BFY05] J. Brunet, L. Flambard et A. Yazman : A hardware in the loop (hil) model development and implementation methodology and support tools for testing and validating car engine electronic control unit (ecu). *International Conferences on CAE and Computational Technologies for Industry, TCN CAE*, 2005.

[BH96] H.-P. Bloch et J.-J. Hoefner : *Reciprocating compressors : operation and maintenance*. Gulf Professional Publishing, 1996.

[BJL+90] J. Brunet, D. Jaume, M. Labarrère, A. Rault et M. Vergé : *Détection et diagnostic de pannes*. Hermès, 1990.

[BKLS03] M. Blanke, M. Kinnaert, J. Lunze et M. Staroswiecki : *Diagnosis and fault-tolerant control*. Springer-Verlag, 2003.

[Blo66] H.-P. Bloch : *Compressors and modern process applications*. Wiley-Interscience, 20066.

[BN93] M. Basseville et I.V. Nikiforov : *Detection of Abrupt Changes : Theory and Application*. Prentice-Hall, 1993.

[BOTM08] M. Bayoudh, X. Olive et L. Travé-Massuyès : Coupling continuous and discrete event system techniques for hybrid system diagnosability analysis. *Proceedings of the 18th European Conference on Artificial Intelligence*, 2008.

[BR03] F. Bonnans et P. Rouchon : *Analyse et commande de systèmes dynamiques - manuel de cours*. Département de mathématiques appliquées, École Polytechnique, 2003.

[Bra98] M.S. Branicky : Analyzing and synthesizing hybrid control systems. *In* G. Rozenberg et F. Vaandrager, éditeurs : *Lectures on Embedded Systems, Lecture Notes in Computer Science*, volume 1494, pages 74–113. Springer Berlin, 1998.

[BRKG02] C. Byington, M. Roemer, G. Kacprzynski et T. Galie : Prognostic enhancements to diagnostic systems for improved condition-based maintenance. *Aerospace Conference Proceedings, IEEE*, 2002.

[BSMB07] P. Bhowal, D. Sarkar, S. Mukhopadhyay et A. Basu : Fault diagnosis in discrete time hybrid systems - a case study. *Information Sciences*, 177(5):1290–1308, 2007.

[BSTMD98] K. Bousson, J. Steyer, L. Travé-Massuyès et B. Dahhou : From a rule-base to a predictive qualitative model-based approach using automated model generation. application to the monitoring and diagnosis of biological process. *Engineering Applications of Artificial Intelligence*, 11:477–493, 1998.

[BTM03] E. Benazera et L. Travé-Massuyès : The consistency approach to the on-line prediction of hybrid system configurations. *IFAC Conference on Analysis and Design of Hybrid Systems, ADHS'03*, 2003.

[Bub05] Z. Bubnicki, éditeur. *Modern Control Theory*. Springer-Verlag Berlin Heidelberg, 2005.

[Car99] T. Carpentier : *Placement de capteurs pour la surveillance des processus complexes*. Thèse de doctorat, Université des Sciences et Technologies de Lille, 1999.

[CFW75] R.N. Clark, D.C. Fosth et W.M. Walton : Detecting instrument malfunctions in control systems. *IEEE Transactions on Aerospace and Electronic Systems.*, 11(4):465–473, 1975.

[CGP00] E.M. Clarke, O. Grumberg et D.A. Peled : *Model Checking*. MIT Press, 2000.

[CL03] R. Cori et D. Lascar : *Logique mathématique - Tome 1 et 2*. Dunod, 2003.

[Coc04] V. Cocquempot : *Contribution à la surveillance des processus industriels complexes - mémoire d'habilitation à diriger des recherches*, 2004.

[CP99] J. Chen et R.J. Patton : *Robust Model-based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, 1999.

[CPC03] A. Cimatti, C. Pecheur et R. Cavada : Formal verification of diagnosability via symbolic model checking. *International Joint Conferences on Artificial Intelligence, IJCAI'03*, 2003.

[CPR00] L. Console, C. Picardi et M. Ribando : Diagnosability analysis using process algebra. *International Workshop on Principles of Diagnosis, DX'00*, 2000.

[CS97] J.-P. Cassar et M. Staroswiecki : A structural approach for the design of failure detection and identification systems. *Proc IFAC, IFIP, IMACS Conference on Control of Industrial Systems*, 1997.

[CTMP06] M.-O. Cordier, L. Travé-Massuyès et X. Pucel : Comparing diagnosability in continuous and discrete-event systems. *International Workshop on Principles of Diagnosis, DX'06*, June 2006.

[DBDGD07a] M.D. Di Benedetto, S. Di Gennaro et A. D'Innocenzo : Diagnosability verification for hybrid automata and durational graphs. *IEEE Conference on Decision and Control*, 2007.

[DBDGD07b] M.D. Di Benedetto, S. Di Gennaro et A. D'Innocenzo : Verification of hybrid automata diagnosability. *IEEE Transactions on Automatic Control*, 2007.

[Der09] H. Derbel : *Diagnostic à base de modèles des systèmes temporisés et d'une sous-classe de systèmes dynamiques hybrides*. Thèse de doctorat, Université Joseph Fourier, Grenoble 1 et École nationale des sciences de l'informatique, Tunisie, Décembre 2009.

[DF94] X. Ding et P.M. Frank : Comparison of observer-based fault detection approaches. *Proceedings of IFAC Safeprocess'94, Helsinki, Finlande*, pages 556–561, 1994.

[DKMR92] J. De Kleer, A. Mackworth et R. Reiter : Characterizing diagnosis and systems. *Artificial Intelligence*, 56:197–222, 1992.

[dR75] J. de Rosnay : *Le macroscope : vers une vision globale*. Seuil, 1975.

[DS03] O. Dressler et P. Struss : A toolbox integrating model-based diagnosability analysis and automated generation of diagnostics. *International Workshop on Principles of Diagnosis, DX'03*, 2003.

[Dub90] B. Dubuisson : *Diagnostic et reconnaissance de formes*. Hermès, Paris, 1990.

[Eti11] M. Etique : *Régulation automatique - manuel de cours*. Institut d'Automatisation Industrielle, Haute école d'ingénierie et de gestion du canton de Vaud (`http://iai.heig-vd.ch`), Février 2011.

[Far89] H. Farreny : *Les systèmes experts - Principes et exemples*. Cépadues, 1989.

[FG03] P. Famouri et R.S. Gemmen : Electrochemical circuit model of a pem fuel cell. *IEEE Power Engineering Society General Meeting*, 2003.

[Fou10] N. Fouquet : Real time model-based monitoring of a pem fuel cell flooding and drying out. *IEEE Vehicle Power and Propulsion Conference, VPPC*, September 2010.

[GCF+95] J.J. Gertler, M. Costin, X. Fang, Z. Kowalczuk, M. Kunwer et R. Monajemy : Model based diagnosis for automative engines - algorithm development and testing on a production vehicle. *IEEE Transactions on Control Systems Technology*, 3:61–69, 1995.

[Ger98] J.J. Gertler : *Fault Detection and Diagnosis in Engineering Systems.* Marcel Dekker Inc., 1998.

[GG96] M. Gomaa et S. Gentil : Hybrid industrial dynamical system supervision via hybrid continuous causal petri nets. *IEEESMC IMACS Symposium on Discrete Events and Manufacturing Systems*, pages 380–384, 1996.

[Gir04] A. Girard : *Analyse Algorithmique des Systèmes Hybrides.* Thèse de doctorat, Institut National Polytechnique de Grenoble, Septembre 2004.

[GLR00] G.L. Gissinger, M. Loung et H.-F. Reynaund : Failure detection and isolation - optimal design of instrumentation system. *IFAC Workshop SAFEPROCESS*, 2000.

[GM95] J.J. Gertler et R. Monajemy : Generating directional residuals with dynamic parity relations. *Automatica*, 31:627–635, 1995.

[GPH+03] J. Garnier, M.C. Pera, D. Hissel, F. Harel, D. Candusso, N. Glandut, J.P. Diard, A.D. Bernardinis, J.M. Kauffmann et G. Coquery : Dynamic pem fuel cell modeling for automotive applications. *IEEE 58th 2003 Vehicular Technology Conference*, 2003.

[GS90] J.J. Gertler et D. Singer : A new structural framework for parity equation-based failure detection and isolation. *Automatica*, 26:381–388, 1990.

[GSW92] R. Greiner, B. Smith et R. Wilkerson : A correction to the algorithme in reiter's theory of diagnosis. *Readings in Model-based Diagnosis*, 1992.

[IAOM79] M. Iri, K. Aoki, E. O'Shima et H. Matsuyama : An algorithm for diagnosis of system failures in chemical processes. *Computers and Chemical Engineering*, 3(4):489–493, 1979.

[IB97] R. Isermann et P. Ballé : Trends in the application of model-based fault detection and diagnosis in technical processes. *Control Engineering Practice - CEP*, 5(5):638–652, May 1997.

[IFI83] IFIP : *Proc. of the IFIP 9th World Computer Congress*, 1983.

[Ise84] R. Isermann : Process fault detection based on modelling and estimation. *Automatica*, 20(4):387–404, 1984.

[Ise97] R. Isermann : Supervision, fault-detection and fault-diagnosis methods - an introduction. *Control Engineering Practice*, 5(5):639–652, May 1997.

[Ise05] R. Isermann : *Mechatronic systems - Fundamentals.* Springer-Verlag, 2005.

[Ise06] R. Isermann : *Fault-Diagnosis Systems.* Springer-Verlag, 2006.

[ISM08] A. Ingimundarson, A. Stefanopoulou et D.-A. McKay : Model-based detection of hydrogen leaks in a fuel cell stack. *IEEE Transactions on Control Systems Technology*, 16(5):1004–1012, September 2008.

[JHCK01] S. Jiang, Z. Huang, V. Chandra et R. Kumar : A polynomial algorithm for testing diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 46:1318–1321, 2001.

[JK04] S. Jiang et R. Kumar : Failure diagnosis of discrete event systems with linear-time temporal logic fault specifications. *IEEE Transactions on Automatic Control*, 49(8):934–945, 2004.

[JM05] M.A. Johnson et M.H. Moradi, éditeurs. *PID Control - New Identification and Design Methods.* Springer-Verlag London, 2005.

[Jon73] H.L. Jones : *Failure Detection in Linear System.* Thèse de doctorat, MIT, Cambridge, 1973.

[KHV06] R. Kothamasu, S.H. Huang et W. VerDuin : System health monitoring and prognostics : a review of current paradigms and practices. *Journal of Advanced Manufacturing Technology*, 28(9-10):1012–1024, 2006.

[Kin03] M. Kinnaert : Fault diagnosis based on analytical models for linear and nonlinear systems - a tutorial. *Proceedings of IFAC Safeprocess'03, Washington DC, USA*, pages 37–50, 2003.

[KN02] M. Krysander et M. Nyberg : Structural analysis utilizing mss sets with application to a paper plant. *International Workshop on Principles of Diagnosis, DX'02*, 2002.

[Kui86] B.J. Kuipers : Qualitative simulation. *Artificial Intelligence*, 1986.

[LAB+96] J.-C. Lapprie, J. Arlat, J.-P. Blanquart, A. Costes, Y. Crouzet, Y. Deswarte, J.-C. Fabre, H. Guillermain, M. Kaaniche, C. Mazet, D. Powell, C. Rabéjac et P. Thévenod : *Guide de la Sûreté de Fonctionnement.* Cépaduès, 2ème édition, 1996.

[LdR96] R. Lassaigne et M. de Rougemont : *Logique et Complexité.* Hermes, 1996.

[Lef09] A. Lefebvre : *Contribution à l'amélioration de la testabilité et du diagnostic de systèmes complexes : Application aux système avioniques.* Thèse de doctorat, Université Joseph Fourier, Grenoble 1, Mai 2009.

[LLA98] J.H. Lee, T.R. Lalk et A.J. Appleby : Modeling electrochemical performance in large scale proton exchange membrane fuel cell stacks. *Journal of Power Sources*, 70:258–268, 1998.

[LMR97] M. Luong, D. Maquin et J. Ragot : Sensor network design for failure detection and isolation. *3rd IFAC Symposium SICICA*, 1997.

[LP77] S.A. Lapp et G.A. Powers : Computer-aided synthesis of fault-trees. *IEEE Transactions on Reliability*, 1977.

[LR08] J. Lévine et P. Rouchon : *Systèmes dynamiques et commande.* Techniques de l'Ingénieur, 2008.

[MAH+00] R.F. Mann, J.C. Amphlett, M.A.I. Hooper, H.M. Jensen, B.A. Peppley et P.R. Roberge : Development and application of a generalized steady-state electrochemical model for a pem fuel cell. *Journal of Power Sources*, 86:173–180, 2000.

[MGT+04] P. Moro, J. Girard, G. Tabet, J.-C. Laprie, C. Duquesne et Codet F. : Sûreté de fonctionnement. *Revue de l'électricité et de l'électronique*, 11, Décembre 2004.

[Mos01] J. Mosterman : Diagnosis of physical systems with hybrid models using parameterised causality. *Hybrid Systems : Computation and Control, 4th International Workshop*, pages 447–458, 2001.

[MP71] R. Mehra et J. Peschon : An innovation approach to fault detection and diagnosis in dynamic system. *Automatica*, 7:637–640, 1971.

[Omd88] T. Omdahl : Reliability, availability and maintainability (ram) dictionnary. *ASQC Quality Press, Milwaukee*, 1988.

[Pap93] C. Papadimitriou : *Computational Complexity.* Addison-Wesley, 1993.

[PCR06] J.-P. Poirot-Crouvezier et F. Roy : Genepac project : Realization of a fuel cell stack prototype dedicated to the automotive application. *World Hydrogen Energy Conference*, 2006.

[Pey09] F. Peysson : *Contribution au pronostic des systèmes complexes.* Thèse de doctorat, Université Paul Cézanne d'Aix-Marseille, 2009.

[PFC89] R.J. Patton, P.M. Frank et R.N. Clark : *Fault diagnosis in Dynamic systems : theory and application*. Prentice Hall, 1989.

[PM01] Y. Papadopoulos et J. McDermid : Automated safety monitoring : a review and classification of methods. *International Journal of Condition Monitoring and Diagnostic Engineering Management*, 2001.

[PPS02] J.T. Pukrushpan, H. Peng et A.G. Stefanopoulou : Simulation and analysis of transient fuel cell system performance based on a dynamic reactant flow model. *ASME International Mechanical Engineering Congress and Exposition*, 2002.

[PR08] N. Petit et P. Rouchon : *Automatique dynamique et contrôle des systèmes - manuel de cours*. Centre Automatique et Systèmes, École des Mines de Paris, Février 2008.

[PSP02] J.T. Pukrushpan, A.G. Stefanopoulou et H. Peng : Modeling and control for pem fuel cell stack system. *American Control Conference*, 2002.

[PSP05] J.T. Pukrushpan, A.G. Stefanopoulou et H. Peng : Control of natural gas catalytic partial oxidation for hydrogen generation in fuel cell aplications. *IEEE Transactions on Control Systems Technology*, 13:3–14, 2005.

[QK06] W. Qiu et R. Kumar : Decentralized failure diagnosis of discrete-event systems. *IEEE Transactions on Systems, Man and Cybernetics - Part A*, 36(2):384–395, 2006.

[Rap08] N. Rapin : Procédé et système permettant de générer un dispositif de contrôle à partir de comportements redoutés spécifiés. Demande de brevet Français n°0804812, déposé le 2 septembre 2008.

[Rei87] R. Reiter : A theory of diagnosis from first principals. *Artificial Intelligence*, 32:57–96, 1987.

[RGPC09] F. Roy, S. Garnit et J.-P. Poirot-Crouvezier : Fisypac project : The first vehicle integration of genepac fuel cell stack. *The International Electric Vehicle Symposium and Exposition, EVS 24*, 2009.

[Rib09] P. Ribot : *Vers l'intégration diagnostic/pronostic pour la maintenance des systèmes complexes*. Thèse de doctorat, Université Toulouse 3 Paul Sabatier, Laboratoire d'Analyse et d'Architecture des Systèmes, Décembre 2009.

[Ric97] J. Richalet : *Commande prédictive*. Techniques de l'Ingénieur, 1997.

[RO09] J. Richalet et D. O'Donovan : *Predictive Functional Control - Principles and Industrial Applications*. Springer-Verlag London, 2009.

[SNCH+00] P. Stevens, F. Novel-Cattin, A. Hammou, C. Lamy et M. Cassir : *Piles à combustible*. Techniques de l'Ingénieur, 2000.

[SP07] A. Schumann et Y. Pencolé : Scalable diagnosability checking of event driven systems. *International Joint Conference on Artificial Intelligence, IJCAI07*, pages 575–580, 2007.

[SRB+02] P. Struss, B. Rehfus, R. Brignolo, F. Cascio, L. Console, P. Dague, P. Dubois, O. Dressler et D. Millet : Model-based tools for the integration of design and diagnosis into a common process - a project report. *International Workshop on Principles of Diagnosis, DX'02*, 2002.

[SSL+95] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen et D. Teneketzis : Diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 1995.

[Sta09] R.F. Stapelberg : *Handbook of Reliability, Availability, Maintainability and Safety in Engineering Design*. Springer-Verlag, 2009.

[TCDTS95] M. Tagina, J.P. Cassar, G. Dauphin-Tangy et M. Staroswiecki : Monitoring of systems modelled by bond graph. *International Conference on Bond Graph Modeling and Simulation*, pages 275–280, 1995.

[TMDG97] L. Travé-Massuyès, P. Dague et F. Guerrin : *Le raisonnement qualitatif pour les sciences de l'ingénieur*. Hermès, 1997.

[TMEO06] L. Travé-Massuyès, T. Escobet et X. Olive : Diagnosability analysis based on component supported analytical redundancy relations. *IEEE Transactions on Systems*, 2006.

[VCL92] A. Villemeur, A. Cartier et M.C. Lartisien : *Reliability, Availability, Maintainability and Safety Assessment, Volume 1, Methods and Techniques*. Wiley, 1992.

[Vil97] A. Villemeur : *Sûreté de fonctionnement des systèmes industriels*. Eyrolles, EDF, 1997.

[VRYK03] V. Venkatasubramanian, R. Rengaswamy, K. Yin et S.N. Kavuri : A review of process fault detection and diagnosis. 'part I to III'. *Computers and Chemical Engineering*, 2003.

[VS87] N. Viswanadham et R. Srichandert : Fault detection using unknown input observers. *Control-Theory and Advanced Technology*, 3:91–101, 1987.

[Wil76] A.S. Willsky : A survey of design methods for failure detection in dynamic systems. *Automatica*, 12:601–611, 1976.

[WN96] B.C. Williams et P.P. Nayak : A model-based approach to reactive self-configuring systems. *Conference on Artificial Intelligence, AAAI'96*, 1996.

[WNS05] C. Wang, M.H. Nehrir et S.R. Shaw : Dynamic models and model validation for pem fuel cells using electrical circuits. *IEEE Transactions on Energy Conversion*, 20(2):442–451, June 2005.

[YL02] T. Yoo et S. Lafortune : Polynomial-time verification of diagnosability of partially-observed discrete-event systems. *IEEE Transactions on Automatic Control*, 47(9):1491–1495, 2002.

[Zei09] X. Zeitoun : Correction/complétude et complexité d'un algorithme de monitoring de formules temporelles. Rapport technique, Rapport de master - CEA : DRT/LIST/DTSI/SOL/09-0192/XZ, 2009.

[Zwi99] G. Zwingelstein : *Sûreté de fonctionnement des systèmes industriels complexes*. Techniques de l'Ingénieur, 1999.

[Zwi09] G. Zwingelstein : *Sûreté de fonctionnement des systèmes industriels complexes - Principaux concepts*. Techniques de l'Ingénieur, 2009.