République Algérienne Démocratique & populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

THÈSE

Présentée pour obtenir le grade de

DOCTEUR

de

L'université 8 Mai 1945 Guelma

Filière: Automatique et Informatique Industrielle

Spécialité : Automatique, Informatique Industrielle et Traitement de Signal

Présentée par

BILAL TOLBI

Surveillance et diagnostic des processus dynamiques hybrides par réseaux de Pétri hybrides élémentaires et automates hybrides linéaires

---

Directeur de thèse : Pr. TEBBIKH Hicham
Co-directeur de thèse : Pr. ALLA Hassane

Devant le jury d'examen composé de :

NEMAMCHA Mohamed	Professeur, Université de Guelma	Président
TEBBIKH Hicham	Professeur, Université de Guelma	Rapporteur
ALLA Hassane	Professeur, UJF de Grenoble France	Co-rapporteur
MOSTEFAI Mohammed	Professeur, Université de Sétif	Examinateur
KECHIDA Sihem	MCA, Université de Guelma	Examinateur
GHOMRI Latéfa	MCA, Université de Tlemcen	Examinateur

2016

Thèse présentée pour obtenir le grade de :

DOCTEUR

de

L'université 8 Mai 1945 Guelma

Filière : Automatique et Informatique Industrielle
Spécialité : Automatique, Informatique Industrielle et Traitement de Signal

Dirigée par :

Pr. TEBBIKH Hicham **Université 8 Mai 1945 de Guelma**
Pr. ALLA Hassane **Université Joseph-Fourier Grenoble, France**

Préparée au sein du :

- LAIG dans l'école doctorale AIITS

- GIPSA-Lab dans le cadre du projet de **coolaboration Franco-Algérien** : **CMEP-TASSILI**, entre l'université 8 Mai 1945 de Guelma et l'université de Joseph-Fourier Grenoble, France

Soutenue publiquement le : **13 Novembre 2016.**

" La théorie, c'est quand on sait tout et que rien ne fonctionne.
La pratique, c'est quand tout fonctionne et que personne
ne sait pourquoi "

- Albert Einstein -

" La Vie est comme une bicyclette. Pour garder l'équilibre, il
faut avancer "

- Albert Einstein -

" N'essaie pas d'être un homme de succès,
essaie plutôt de devenir un homme de valeurs "

- Albert Einstein -

Dédicace

Je tiens à dédier cette thèse:

A ma très chère ***Mère*** *et à mon cher* ***Père****, en témoignage et en gratitude de leurs dévouement, de leurs soutien permanent durant toutes mes années d'études, leurs sacrifices illimités, leurs réconfort moral, eux qui ont consenti tant d'effort pour mon éducation, mon instruction et pour me voir atteindre ce but, pour tout cela et pour ce qui ne peut être dit, mes affectations sans limite.*

A ceux qui sont la source de mon inspiration et mon courage, à qui je dois de l'amour et de la reconnaissance :

Mes frères, ***Mohamed Amine*** *et* ***Azdine*** *;*

Ma fidèle ***femme*** *;*

Toute ma famille ;

Mes chers amis(es).

Remerciement

Cette thèse marque une étape importante dans ma vie. Pour cela, je tiens tout d'abord à remercier **ALLAH** *de m'avoir permis d'accomplir ce travail avec succès qui a été réalisé au sein des deux laboratoires : Laboratoire d'Automatique et Informatique de Guelma (***LAIG***) de l'université de Guelma et Grenoble Images Parole Signal Automatique (***GIPSA-Lab***), de L'Institut National Polytechnique INP de Grenoble dans le cadre du projet de recherche de collaboration* **Franco-Algérien (CMEP-TASSILI)**.

C'est avec un immense plaisir que je rédige cette rubrique de remerciement. Dans cette partie où sont évoqués les personnes qui m'ont aidé à faire et à achever ce projet que je considère comme le plus important de ma vie professionnelle.

Je tiens à exprimer ma reconnaissance profonde à mes directeurs de recherche le Professeur **Hicham TEBBIKH** *et le professeur* **Hassane ALLA** *qui m'ont accordé leur confiance et m'ont soutenu sur de nombreux plans durant cette thèse. Je les remercie vraiment du fond du coeur, que ce soit pour leurs qualités scientifiques ou humaines. Je les remercie pour l'aide scientifique qu'ils m'ont toujours apporté, mais surtout pour leur disponibilité, leur soutien et leurs encouragements.*

Je présente aussi mes remerciements à **Pr. NEMAMCHA Mohamed** *pour avoir accepté de présider mon jury de soutenance de thèse. Je tiens également à remercier les examinateurs de thèse* **Pr. MOSTEFAI Mohammed**, **Dr. KECHIDA Sihem** *et* **Dr. GHOMRI Latéfa**, *d'avoir apporté leur caution scientifique en acceptant d'examiner ce travail et d'avoir participer au Jury. Ils ont également contribué par leurs nombreuses remarques et suggestions à améliorer la qualité de la thèse ; je leur en suis très reconnaissant.*

Mes sincères remerciements à **Dr. Yamen EL Touati** *pour son soutien et sa coopération professionnelle.*

Je suis également reconnaissant au soutien financier apporté par le projet de coopération ***Franco-Algérien CMEP-TASSILI.*** *Ce soutien, attribué par le Ministère des affaires étrangères et européennes française et géré par le* ***CAMPUS****, m'a été d'un grand apport pour la réalisation de ce travail. Je tiens à remercier les initiateurs de ce projet.*

Mes remerciements vont à tout le personnel du laboratoire ***GIPSA-lab*** *et de l'équipe de recherche* ***SysCo*** *qui m'ont accueilli durant mes séjours. Je tiens aussi à remercier particulièrement mes deux amis du laboratoire* ***GIPSA-lab Nassim*** *et* ***Rachid*** *qui m'ont aidé à m'intégrer au sein du laboratoire et surtout de leurs aides lors de mes séjours à Grenoble.*

*Je ne pourrais pas terminer sans exprimer un remerciement venant du plus profond du coeur à tous mes amis(es) et collègues, en particulier ceux et celles qui m'ont apporté un soutien moral et une amitié inoubliable et précieuse (****Fares, Abdelhafid, Abd ALLAH, Amine, Fayçal, Djalal, Zinou****....).*

Ce travail représente un résultat parmi les résultats du projet de recherche coopératif Franco-Algérien (CMEP-TASSILI), dans lequel nous nous sommes intéressés à la surveillance et diagnostic d'une classe particulière des SDHs, à savoir la classe des systèmes à flux continus. Cette classe a pour particularité de comporter une dynamique continue linéaire et positive commandée par une dynamique discrète et elle est suffisamment riche pour permettre une modélisation réaliste de nombreux problèmes. D'autre part, sa simplicité relative permet une conception facile d'outils et de modèles pour sa description et son analyse.

La course à améliorer la performance des SDHs a conduit à l'élaboration de systèmes de plus en plus complexes multipliant les risques de dysfonctionnement pouvant mettre en péril le système lui-même et son environnement. Par conséquent, pour un grand nombre d'applications, il est nécessaire d'implanter un système de surveillance et/ou de diagnostic afin de détecter, localiser et identifier les défauts.

La modélisation des SDHs cherche à formaliser des modèles précis qui peuvent décrire le comportement riche et complexe. Dans notre manière de modélisation, nous avons utilsé l'approche mixte pour modéliser l'évolution du système en utilisant deux outils les plus forts et les plus utilisés dans la littérature : Le modèle Réseaux de Pétri Hybride (RdPH) et le modèle d'Automate Hybride (AH). Pour avoir une représentation forte, nous avons combiné les avantages des deux modèles. Cette modélisation a été réalisée en trois parties: la première produit un modèle en fonctionnement normal du système, la deuxième produit les modèles des défauts et la troisième est une combinaison structurelle de tous les modèles. Nous avons introduit une variété des RdPH, dite RdPH élémentaires combinant un RdP T-temporel et RdP continu à vitesses constantes (RdPCC). Dans ce modèle, la partie discrète contrôle la partie continue et vice–versa. Le RdPH élémentaire a été translaté systématiquement, avec une manière structurelle, en automates hybrides linéaires. Le modèle résultant représente le modèle de base pour effectuer la surveillance et le diagnostic. Une technique de bisimilarité temporelle a été présentée et prouvée mathématiquement pour démontrer la similarité entre les deux modèles en termes d'un système de transitions temporisées. Le principe de cette méthode a été illustré à travers un exemple illustratif du type considéré et en présence des défauts.

Pour diagnostiquer le type des systèmes considérés, une solution a été introduite pour synthétiser un diagnostiqueur hors-ligne à base de l'automate dynamique qui a été construit à base de l'automate hybride linéaire résultant de la translation en vue d'élaborer une nouvelle méthode de diagnostic complémentaire pour l'approche de modélisation. Ensuite, le cadre de ces idées a été présenté sur le même exemple illustratif en plusieurs scénarios de défaillance. A la fin de ce manuscrit, nous avons présenté les perspectives de continuation de nos travaux de recherche.

Mots clés : Système à flux continu ; modélisation des défauts ; RdPH élémentaire ; Translation ; Bisimilarité temporelle ; Automate hybride linéaire ; surveillance ; Diagnostic.

This work represents one of the results of the Franco-Algerian cooperative research project (CMEP-TASSILI), in which we are interested on monitoring and diagnosis of a particular class of Hybrid Dynamic Systems (HDS), namely continuous flow systems. This class has the distinction of including a positive linear continuous dynamic controlled by a discrete dynamic. It is sufficiently rich to allow a realistic modelling of numerous problems. On the other hand, its relative simplicity enables an easy design of tools and models for its description and analysis.

The race to improve the performance of HDS has led to the development of systems increasingly complex multiplying the risks of malfunction that could jeopardize the system itself and its environment. Therefore, for many applications, it is necessary to implement a monitoring and/or diagnostic system to detect, locate and identify faults.

HDS modelling seeks to formalize accurate models that can describe the rich and complex behaviour. In our way of modelling, we have used a mixed approach to model the system evolutions using both strongest and the most used tools in the literature: Hybrid Pétri Nets (HPN) and the Hybrid Automata (HA). To have a strong representation, the advantages of both models have been combined. This modelling has been performed in three parts: the first produces a normal operation model of the system, the second produces faults models and the third is a structural combination of all models. A variety of HPN has been introduced, called elementary HPN combining T-time PN and Continuous Constant speeds PN (CCPN). In this model, the discrete part control the continuous part and vice-versa. Elementary HPN has been translated systematically, with a structural manner, into linear hybrid automata. The resulting model is the appropriate model for conducting monitoring and diagnosis. A timed bisimilarity technique has been presented mathematically and proven to demonstrate the similarity between the two models in terms of a timed transitions system. The principle of this method has been shown through an illustrative example in the presence of faults.

To diagnose the type of the considered systems, a solution has been introduced to synthesize a diagnoser off-line based on the dynamic automaton that has been built based on the linear hybrid automaton resulting from the translation to develop a new complementary diagnostic method for the modelling approach. Then, the framework of these ideas has been presented on the same illustrative example in several failure scenarios. At the end of this manuscript, the continuation prospects of our research have been presented.

Keywords: continuous flow system; Faults modelling; elementary HPN; Translation; Timed bisimilarity; Linear hybrid automaton; Monitoring; Diagnosis.

تمثل هذه الأطروحة ثمرة من ثمار مشروع البحث التعاون الفرنسي-الجزائري (CMEP-TASSILI)، حيث إهتممنا بمراقبة وتشخيص فئة إستثنائية من الأنظمة الديناميكية الهجينة، وهي فئة أنظمة التدفق المستمر. هذه الفئة تحتوي على ديناميكية مستمرة خطية و موجبة تحكمها ديناميكية متقطعة. هذه الفئة غنية بما فيه الكفاية لتسمح بنمذجة واقعية لمشاكل عديدة. كما تمكن بساطتها النسبية من تصميم أدوات ونماذج سهلة من أجل وصفها وتحليلها.

أدى التسابق لتحسين أداء هذه الأنظمة الديناميكية الهجينة إلى تطوير أنظمة معقدة بشكل متزايد مما أدى إلى تضاعف مخاطر حدوث أي خلل وظيفي الذي يمكن أن يعرض النظام وبيئته للخطر. لذالك، فمن الضروري زرع نظام مراقبة و/أو تشخيص لكشفه و تحديد موقع الأعطاب.

تسعى نمذجة هذه الأنظمة الديناميكية الهجينة لإضفاء الطابع الرسمي على النماذج الدقيقة التي يمكن أن تصف السلوك الغني و المعقد. فيما يخص طريقتنا في النمذجة، إستخدمنا نهج مختلط لنمذجة تطورات النظام باستخدام أداتين هما الأقوى و الأكثر إستعمالا في هذا المجال: نموذج شبكات بتري الهجينة (RDPH) ونموذج الأوتومات الهجينة (AH). من أجل الحصول على تمثيل قوي، قمنا بتركيبه محاسن كلا النموذجين. كما قمنا بإنجاز هذه النمذجة على ثلاثة أجزاء: الجزء الأول ينتج نموذج نظام أثناء التشغيل العادي، و الجزء الثاني ينتج نماذج الأعطاب، و الجزء الثالث هو تركيبه هيكلي لجميع النماذج. كما قدمنا مجموعة متنوعة من RDPH، تدعى RDPH العنصرية التي تجمع بين شبكات بتري ذات الإنتقالات الزمنية (RdP T-temporel) و شبكات بتري المستمرة ذات سرعة ثابتة (RdPCC). في هذا النموذج، الجزء المتقطع يسيطر على الجزء المستمر والعكس صحيح. الRDPH العنصري تمت ترجمته منهجيا، و بطريقة هيكلية، إلى أوتومات هجين خطي، النموذج الناتج هو النموذج الأساسي لإجراء المراقبة والتشخيص. كما قدمنا البيسيميلارتي الزمنية وبرهانها الرياضي لتوضيح أوجه التشابه بين النموذجين على شكل نظام الإنتقالات الزمنية. تم توضيح مبدأ هذه الطريقة من خلال مثال توضيحي من النوع المطروح مع وجود أعطاب.

لتشخيص الأنظمة المعنية في هذه الأطروحة، تم تقديم حلا لبناء مشخص غير متصل يعتمد على الأتومات الديناميكي، الذي تم بناؤه على أساس الأوتومات الهجين الخطي الناتج عن عملية الترجمة من أجل تطوير طريقة تشخيص جديدة مكملة لمنهجية النمذجة المقترحة. ثم عرضنا إطار هذه الأفكار على نفس المثال التوضيحي في عدة سيناريوهات الإخفاق. في نهاية هذه المخطوطة، قدمنا آفاق استمرارية أبحاثنا.

كلمات البحث: نظام التدفق المستمر؛ نمذجة الأعطاب؛ RDPH العنصرية؛ الترجمة؛ البيسيميلارتي الزمنية؛ الأوتومات الهجين الخطي؛ المراقبة؛ التشخيص.

Table des matières

Chapitre 2 : SURVEILLANCE DES SYSTÈMES DYNAMIQUES

Chapitre 3 : MODÉLISATION DES SYSTÈMES DYNAMIQUES HYBRIDES À FLUX CONTINU TOLÉRANTS AUX DÉFAUTS

Table des Figures

Chapitre 1

Chapitre 2

Chapitre 3

Chapitre 4

Chapitre 5

INTRODUCTION GÉNÉRALE

L'automatique traite différemment les problèmes de type continu et ceux de type discret. Chacun de ces domaines a créé un ensemble de théories et de méthodes et développé des solutions performantes pour régler les problèmes homogènes qui se posent, mais sans toujours intégrer les solutions et les apports de l'autre domaine.

En effet, la modélisation des systèmes dynamiques est, depuis longtemps, étudiée pour l'analyse, la commande, la surveillance et le diagnostic. Ainsi, il est possible d'étudier le comportement dynamique d'une variété considérable de procédés et systèmes physiques pour observer leur comportement, le modifier, détecter et identifier toute anomalie dans celui-ci. Chronologiquement, les systèmes continus ont été les premiers à être étudiés. Leur modélisation s'effectue généralement au moyen d'équations différentielles. Ces systèmes traitent des grandeurs continues comme la température, la pression, le flux, etc.

On trouve dans la littérature un classement selon la nature et l'évolution des états des systèmes. Les systèmes à événements discrets (SED) est la classe des systèmes dynamiques dont l'espace d'état est discret, c'est-à-dire que l'évolution se fait par l'occurrence instantanée d'événements faisant ainsi changer l'état du système. L'information temporelle dans ces systèmes n'est pas retenue d'une manière explicite ; seulement l'ordre logique de l'occurrence des événements est pris en considération. Cette dernière classe - les SED - ne couvre pas les systèmes où les contraintes temporelles représentent une information cruciale pour décrire leurs comportements. Par extension, les systèmes définis, en sus, par les contraintes temporelles sont les systèmes à événements discrets temporisés (SEDT). Les SEDT sont toujours pilotés par les événements discrets. Cependant, ils comportent une partie continue réduite à un simple écoulement du temps. Certains des systèmes à temps réels, des systèmes "Man Made" et des systèmes embarqués ont une partie continue où des variables physiques, représentant le système, évoluent de façon continue conformément à une loi donnée. Ces systèmes sont composés de sous processus continus qui sont commandés, supervisés, reconfigurés et arrêtés par une commande logique discrète. Les systèmes dans lesquels les dynamiques discrètes et continues interagissent et où leurs interactions déterminent le comportement quantitatif et qualitatif, sont appelés systèmes dynamiques hybrides (SDHs [Zaytoon, 01]. Les deux aspects sont étroitement liés, et le degré d'intégration étant élevé, la modélisation du système hybride par une seule de ses composantes n'est pas toujours possible.

Dans ce travail, nous nous sommes intéressés à une classe particulière des SDHs, à savoir la classe des systèmes à flux continus. Cette classe a pour particularité de comporter une dynamique continue linéaire et positive commandée par une dynamique discrète. Un système hybride est dit positif si ses variables d'état prennent des valeurs positives dans le temps. Il est dit linéaire par morceaux si les lois décrivant son évolution continue sont formulées au moyen d'équations différentielles linéaires au sens des automates hybrides, les variables continues ont une évolution affine en fonction du temps. Un effort particulier a été apporté à l'étude de cette classe pour deux raisons principales. D'abord, elle est suffisamment riche pour permettre une modélisation réaliste de nombreux problèmes. Ensuite, sa simplicité relative permet une conception facile d'outils et de modèles pour sa description et son analyse. Cette classe englobe

plusieurs problèmes réalistes, comme les procédés batch, les systèmes manufacturiers traitant une quantité importante de produits, les systèmes de transport, de communication et autres.

La modélisation des SDHs cherche à formaliser des modèles précis qui peuvent décrire le comportement riche et complexe. Plusieurs formalismes ont été proposés afin d'établir un modèle homogène permettant la conciliation entre les parties discrètes et continues. Pour intégrer les aspects continu et événementiel au sein d'un même formalisme, nous devons tenir compte du modèle dominant, à partir duquel s'effectue l'extension, et trois approches sont, alors, possibles:

1. L'approche continue a pour principe d'intégrer l'aspect discret dans un formalisme pour les systèmes continus, et ceci par l'introduction de variables booléennes ou entières dans un système d'équations.

2. L'approche discrète a pour principe d'intégrer l'aspect continu dans un formalisme pour les systèmes à événements discrets comme les réseaux de Pétri (RdP) ou les automates à états finis. Le champ d'application de ces derniers a été étendu pour prendre en compte les systèmes dynamiques hybrides. Lorsqu'un RdP discret contient un grand nombre de jetons, le nombre d'états atteignables explose. Cette limitation des RdP a conduit à l'apparition des RdP continus (RdPC) en 1987 par David et Alla [David et Alla, 87], puis les RdP hybrides (RdPH). Les marquages de places dans un RdPC sont des nombres réels et le franchissement des transitions est un processus continu. Cependant, si le nombre de pièces dans un stock peut être approximé à un processus continu, l'état d'un élément du système tel que vanne ouverte ou fermée ne peut être modélisé par un nombre réel. La modélisation d'un système hybride conduit naturellement aux RdPH contenant une partie discrète et une partie continue.

3. L'approche mixte a pour principe d'intégrer l'aspect continu et discret dans une même structure. Cette approche repose sur l'utilisation d'un modèle à événements discrets comme moniteur de système d'équations.

Le développement de l'automatisation des systèmes vise à améliorer leurs performances. Cette course à la performance a conduit à l'élaboration de systèmes de plus en plus complexes multipliant les risques de dysfonctionnement pouvant mettre en péril le système lui-même et son environnement. Par conséquent, pour un grand nombre d'applications, il est nécessaire d'implanter un système de surveillance et/ou de diagnostic afin de détecter, localiser et identifier les défauts. Cette dernière tâche nécessite la connaissance d'un modèle de défauts.

Dans ce contexte, de nombreuses approches sont développées, en vue de la détection de défauts et du diagnostic, par les différentes communautés de recherche en Automatique. Les méthodes se différencient par rapport au type de connaissance à priori sur le processus qu'elles nécessitent. Ainsi, elles peuvent être classées, de façon générale, comme suite : Les méthodes à base de modèles considèrent un modèle structurel du comportement du processus basé sur des principes physiques fondamentaux. Ces modèles peuvent être de type quantitatif, exprimés sous forme d'équations mathématiques ou bien de type qualitatif, exprimés par exemple sous forme de relations logiques ; Les méthodes sans modèles exploitent les compétences, le raisonnement et

les connaissances des experts sur le processus pour les transformer en règles, de manière à résoudre des problèmes spécifiques.

Dans ce travail, nous nous sommes intéressés à la modélisation d'une classe particulière des systèmes dynamiques hybrides, à savoir la classe des systèmes à flux continus tolérents aux défauts. Pour cela, la modélisation proposée dans ce travail est éllaborée en trois parties : la première produit un modèle en fonctionnement normal du système, la deuxième produit les modèles des défauts, la troixième est une combinaison structurelle de tous les modèles. Nous avons introduit une variété des RdPH, dite RdPH élémentaires combinant un RdP T-temporel et RdP continu à vitesses constantes (RdPCC). Dans ce modèle, la partie discrète contrôle la partie continue et vice–versa.

C'est un des formalismes adaptés pour la modélisation, cependant pour l'analyse, la surveillance et le diagnostic de ce type des systèmes, on utilise traditionnellement une sous classe d'automates hybrides à cause de leur facilité d'analyse formelle. Pour cela, une procédure systématique est élaborée dans ce manuscrit permettant la translation structurelle des RdPH élémentaires en automates hybrides linéaires. Le modèle résultant sera le modèle de base pour effectuer la surveillance et le diagnostic. Ensuite, une analyse faite sous le logiciel PHAVer permettant de calculer l'espace atteignable de l'automate résultant. Le rapport donc est organisé en cinq chapitres, comme suit :

Dans **le premier chapitre**, nous introduirons les notions fondamentales relatives aux SDHs. Leur définition ainsi leurs différentes classes et phénomènes hybrides trouvés en littérature seront présentés. Nous passerons en revue les principaux formalismes de modélisation, de simulation et d'analyse des SDHs. Dans **le deuxième chapitre**, nous présenterons le concept général de la surveillance des systèmes dynamiques. Ce chapitre introductif permet de situer les premiers pas de notre contribution proposée dans cette thèse. **Le troisième chapitre** est dédié aux modèles de base de la technique proposée pour la modélisation des SDHs à flux continu tolérant aux défauts. La notion de la tolérance aux défauts et la modélisation des défauts possibles seront établies pour ces systèmes dans leurs différents types. Ensuite, les deux modèles utilisés pour la représentation du système en comportement normal et en présence des défauts seront présentés. Ce chapitre constitue la première contribution de ce travail de thèse. Le **quatrième chapitre** est dédié à la présentation d'une procédure de translation en automates hybrides linéaires d'une classe particulière des RdPH appelée RdPH élémentaires. Pour cela, nous présenterons les étapes permettant la translation structurelle à travers des exemples intuitifs, en proposant un algorithme permettant de garantir cette translation d'une manière systématique. Une technique de bisimilarité temporelle sera présentée et prouvée pour démontrer la similarité entre les deux modèles en termes d'un système de transitions temporisées. Enfin, nous illustrerons le principe de cette méthode sur un exemple (illustratif) du type considéré et en présence des défauts. Nous utiliserons, l'application PHAVer pour l'analyse de l'atteignabilité de l'automate hybride résultant. Ce chapitre constitue la deuxième contribution de ce travail de thèse. Sur le **dernier chapitre**, nous présenterons quelques pistes de recherche dans le cadre du diagnostic des SDHs. Nous en déduirons des perspectives de continuation de nos travaux de recherche.

CHAPITRE 1

LES SYSTÈMES DYNAMIQUES HYBRIDES : PRÉSENTATION, MODÉLISATION ET ANALYSE

Résumé : Dans le premier chapitre, nous présenterons la notion des systèmes dynamiques hybrides allant de la caractérisation de ces derniers à la description et la classification des aspects hybrides. Par la suite, nous entamerons les approches et les outils de modélisations et nous discuterons les outils de simulation et analyse des systèmes dynamiques hybrides. Notons que ce chapitre ambitionne, d'une part, à présenter les outils facilitant la lecture des chapitres qui le suivent et d'autre part à justifier le choix des Automates Hybrides Linéaires et les Réseaux de Petri Hybrides élémentaires comme outils de modélisation.

1.1. Introduction

Un système est un ensemble d'objets interagissant entre eux pour réaliser une fonction. Il est connecté au monde extérieur à travers ses entrées (commande et perturbations) et sorties (répenses du système). Les systèmes peuvent être classifiés à : systèmes statiques; systèmes dynamiques; systèmes monovariables; systèmes multivariables; systèmes continus; systèmes à événements discrets; systèmes linéaires ou non linéaires; systèmes causaux; systèmes invariants ou stationnaires…etc. Les progrès technologiques liés à l'automatisation et à l'informatisation ont largement contribué à la croissance de la complexité des systèmes et des processus industriels. En effet, l'absence de plus en plus, de l'intervention humaine et l'introduction de l'informatique industrielle dans l'industrie ont imposé de nouvelles méthodologies dans le développement et la conception des systèmes complexes en particulier sur le plan du contrôle, de la surveillance et du diagnostic.

La plupart de ces processus présentent un comportement à la fois continu et discret. Avec de telles structures, la dynamique de ces derniers nécessite de plus en plus des outils qui tiennent compte de l'ensemble des dynamiques, ce que les approches classiques ne peuvent pas appréhender. En effet, la séparation des systèmes à évènement discret (SED) des systèmes continus (SC) et le traitement de chaque type à part peuvent engendrer d'une part des simplifications significatives mais aussi de nouvelles problématiques de modélisation, d'identification et d'analyse.

Néanmoins, les systèmes réels ou industriels sont souvent des systèmes complexes dont la dynamique est modélisée, d'un point de vue macroscopique, par des phénomènes discrets et continus. De ce fait, le couplage de l'aspect continu avec l'aspect discret nous oriente vers de nouveaux concepts et de nouvelles approches par l'intégration des méthodologies des deux théories dans une nouvelle classe de système nommée *''système hybride''(SH)*. Depuis les années quatre-vingt-dix, une attention particulière de la part de la communauté scientifique s'est portée sur l'étude de ces systèmes hybrides [Alur *et al,* 95] [Branicky, 95] [Engell, 97] [Guéguen et Lefebvre, 01] et de nombreuses approches de modélisation traitant à la fois les aspects continus et discrets ont été proposées.

Aussi, dans ce chapitre introductif, nous nous consacrerons, dans un premier temps, à présenter les systèmes dynamiques hybrides et à revenir sur la définition de ces derniers. Par la suite, nous abordons par un tour d'horizon les outils de modélisation présentés dans la littérature, dans le but d'argumenter le choix du modèle considéré dans notre travail.

1.2. Caractérisation des systèmes dynamiques hybrides

Les systèmes dynamiques hybrides (SDH) sont des systèmes pour lesquels les dynamiques discrètes et continues interagissent. Cette interaction détermine le comportement du système. On peut trouver plusieurs types de systèmes hybrides : systèmes intrinsèquement hybrides, systèmes continus avec commandes discrètes, systèmes à événements discrets évoluant d'une manière continue ou systèmes continus évoluant avec des commutations discrètes. Les deux composantes

continue et discrète d'un SDH sont interconnectées avec une loi qui orchestre cette interconnexion (voir Figure 1.1) [Antsaklis, 00].

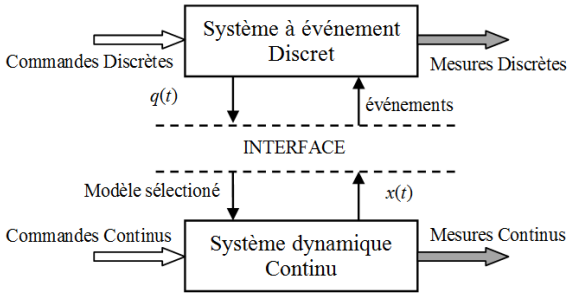


Figure 1.1. Structure d'un système dynamique hybride.

Le SDH est caractérisé par la nature de ces variables d'état. En fonction des caractéristiques de ses paramètres, ces dernières sont généralement présentées soit par des variables d'état continues (et/ou échantillonnées), soit par des variables d'état discrètes (évènementielles). À partir de là, nous pouvons situer le concept hybride des systèmes. La Figure 1.2 illustre les notions de base suivantes :

♦ Dans un système continu, les variables d'état sont considérées comme des fonctions continues et dérivables en fonction du temps. Une telle variable d'état présente une trajectoire continue en fonction du temps. Ainsi, le modèle correspondant exprime le taux de progression de l'état du système en fonction du temps. Nous considérerons les systèmes dits échantillonnés dans cette même classe de systèmes.

♦ Dans un système à évènement discret, les variables d'états prennent leurs valeurs sur un ensemble fini (sous ensemble des réels). Il s'agit en quelque sorte d'une abstraction de type logique. L'état du système change dès l'apparition d'un événement instantané et demeure constant durant l'intervalle de temps séparant deux événements consécutifs. Ainsi, le modèle correspondant est caractérisé par des transitions d'état instantanées et le temps évolue en fonction de la date des prochaines transitions. Cette évolution correspond, dans la plupart des cas, à l'état ouvert ou fermé (par exemple marche ou arrêt).

♦ Le système hybride regroupe les deux précédents types. Il présente des phases et des séquences décrivant les modes ou bien les différentes tâches et activités du système. Chaque phase est décrite par une évolution continue. Ainsi, un modèle hybride est caractérisé à la fois par une évolution continue et une évolution évènementielle.

Les interactions entre les deux modèles se font par l'intermédiaire des évènements. Au niveau de la partie discrète, un événement correspond à un franchissement de transition. Alors qu'au niveau des systèmes continus, il s'agit par exemple d'un dépassement de seuil d'une variable continue. Une transition d'un mode vers un autre mode a lieu lorsque certaines conditions logiques sont vérifiées.

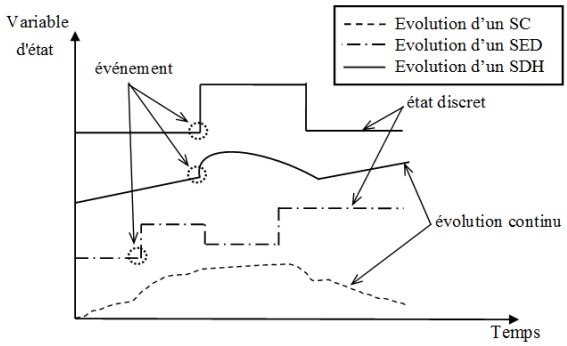


Figure 1.2. Evolution d'un système dynamique selon sa nature.

1.2.1. Exemples de systèmes dynamiques hybrides

Il existe plusieurs systèmes physiques, réels ou technologiques qui ont un comportement hybride du fait qu'ils possèdent à la fois une dynamique discrète et une dynamique continue. Nous allons présenter plusieurs types de ces systèmes rencontrés dans la littérature.

A. *Systèmes continus supervisés par un contrôleur à événements discrets*

Un procédé continu commandé ou supervisé par un système à événements discrets est appelé système hybride par la commande. Cette classe de systèmes hybrides est largement étudiée dans la littérature [Branicky, 94] [Antsaklis *et al*, 93] [Stiver *et al*, 96]. Citons à titre d'exemple un thermostat utilisé pour maintenir la température dans une pièce [EL Mezyani, 05].

B. *Systèmes continus comportant des discontinuités*

Les phénomènes de discontinuités se produisent lorsque l'état passe instantanément de sa valeur courante à une autre valeur. Ce phénomène de commutations est illustré à travers l'exemple classique d'une balle en rebondissement sur un sol de façon élastique ou la collision entre deux corps [Branicky, 95]. Dans les deux cas, la vitesse change brutalement et subit donc un saut.

C. *Systèmes comportant des éléments discrets et continus*

Certains systèmes sont constitués intrinsèquement d'éléments de type "continu" (les variables contraintes ou produites ont une évolution continue) et d'éléments de type "discret" (les variables contraintes ou produites sont à valeur discrète). Les circuits électroniques contenant des éléments à caractéristiques continues (résistance, condensateur, self, etc.) et des éléments à caractéristiques discrètes (interrupteur, diode, thyristor, etc.) sont des exemples de tels systèmes.

D. *Systèmes continus pour lesquels des dynamiques discrètes sont introduites par abstraction*

Dans certains cas où les phénomènes physiques sont complexes, la modélisation nécessite l'utilisation de fonctions non-linéaires difficiles à manipuler. Certains travaux proposent

d'introduire des phénomènes discrets au sein de l'évolution continue afin de simplifier la modélisation [EL Mezyani, 05].

Un système non-linéaire, un multi-modèle ou un système continu par parties correspondent tous à des structures résultant de l'agrégation de modèles continus locaux. Cette succession de modèles continus peut être représentée comme un SDH. La dynamique discrète sera introduite par abstraction des dynamiques rapides qui peuvent avoir lieu au moment du changement de modèle (commutations spontanées). Les dynamiques complexes mais très rapides par rapport à la dynamique globale peuvent être négligées. Ces approximations doivent être utilisées avec beaucoup de précautions et dépendent de l'utilisation qui est faite du système et des objectifs visés.

E. ***Systèmes discrets pour lesquels des dynamiques continues sont introduites par abstraction***

Ces systèmes sont généralement des systèmes ayant des dynamiques discrètes riches, c'est-à-dire dont l'évolution de l'état discret est rapide par rapport à la dynamique globale du système [Kurovszky, 02]. Citons à titre d'exemple un système de production, un système de trafic urbain …etc.

F. ***Systèmes complexes composés de sous-systèmes continus et discrets***

Dans les industries dites de « process », élaborant les matières premières qui seront travaillées par les industries manufacturières, la production peut se faire en continu ou par des traitements successifs : on parle de procédés de traitement par lots. Ces procédés, très présents dans le domaine de l'industrie chimique, pharmaceutique ou agro-alimentaire, comportent des séquences de transfert et de conditionnement relevant des systèmes à événements discrets (SED) et des opérations continues pendant un certain temps : évaporation, cristallisation, mélange, etc. Citons à titre d'exemple le procédé Batch.

1.2.2. Les classes des SDHs

La nature hybride d'un système peut être inhérente aux phénomènes physiques qui le régissent. Nous pouvons distinguer plusieurs formes de comportement hybride. Leur définition repose sur la notion de changement brusque ou instantané de l'état ou du modèle. Ces changements peuvent être autonomes ou contrôlés. Le changement autonome résulte d'une évolution interne du système, alors que le changement contrôlé est dû à une action extérieure. Les comportements hybrides ont été regroupés selon les sauts ("*jump*") et les commutations ("*switching*") en quatre catégories principales [Branicky, 96] traduisant leur influence sur les modèles mathématiques, la classification est donnée comme suit

A. Commutations autonomes

Ce type de comportement est la conséquence de l'évolution de la variable d'état continue. Ainsi, une commutation autonome est caractérisée par un changement discontinu du champ de

vecteur lorsque l'état atteint certains seuils. La Figure.1.3 illustre que le système change de dynamique dès que la variable d'état atteint une valeur donnée dans l'espace d'état.

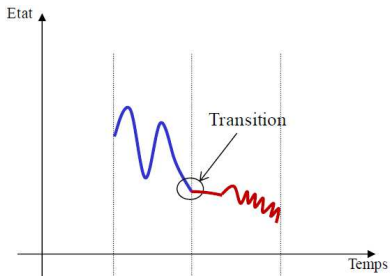


Figure.1.3. Commutation autonome.

B. Commutations contrôlées

Il s'agit de transitions provoquées par une commande ou dues à une modification de l'environnement, donc à une action extérieure au système considéré. C'est le cas d'une transmission manuelle ou du système à réservoir lors du passage de l'état ''vidange'' à l'état ''remplissage'' après action sur les vannes. Cette dernière affecte le vecteur champ du système d'une façon instantanée.

C. Sauts autonomes

La transition subit un saut de type autonome quand la variable d'état atteint une certaine région de l'espace d'état. Il effectue un saut ; i.e. il passe de façon discontinue de sa valeur courante à une autre (Figure 1.4). C'est le cas par exemple lors de la collision entre deux corps où la vitesse change de signe brutalement et subit un saut (balle qui rebondit). Notons que ce type de phénomènes est généralement engendré par une approximation lors de la modélisation qui suppose que certains phénomènes sont infiniment rapides. Il se traduit par une discontinuité de la valeur courante à une autre valeur.

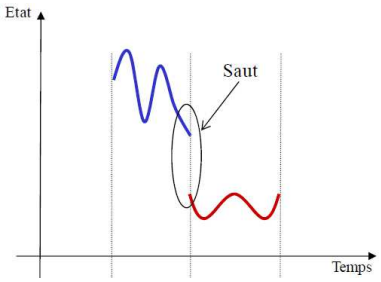


Figure.1.4. Saut autonome.

D. Sauts contrôlés

Ce comportement est exprimé par un changement de l'état sous l'effet d'une commande. La réponse du système se fait d'une façon discontinue. C'est le cas d'un modèle de stock où on dépose les quantités a_1, a_2, … de matière aux instants $t_1<t_2<\dots$. Plusieurs problèmes de

complexité ont été rencontrés par certains chercheurs, notamment qui sont rencontré par Van der Schaft et Schumacher [Schaft et Schumacher, 00]:

♦ Le système atteint un état où il n'existe plus de trajectoires continues définies et où il n'y a aucune transition menant d'un état vers un autre. Le système est dit alors « *bloqué* ».
♦ Le système commute indéfiniment entre deux états, cette situation est appelée « *livelock* ».
♦ Les durées des trajectoires continues (i.e., le temps pendant lequel le système évolue entre deux sauts/commutations) deviennent de plus en plus petites. Le système est dit « *Zénon* ».
♦ L'ensemble des trajectoires est vaste et nous n'avons donc pas une solution unique.
♦ L'évolution du système rencontre plusieurs événements simultanés. Nous avons plusieurs changements d'état au même instant.
♦ Un état du système commute sur lui-même indéfiniment.
♦ La partie continue tend vers l'infini dans un temps fini.

Ces difficultés liées à l'intégration des deux aspects (continue et discret) ont conduit les scientifiques à mener des recherches dans plusieurs domaines de manière à répondre aux besoins et aux exigences des systèmes réels, complexes et industriels.

1.2.3. Les systèmes dynamiques hybrides *à flux continu*

Dans ce travail nous nous intéressons a une sous classe des SDHs, appelée, *systèmes à flux continu* (SaFC). Ces systèmes ont les trois caractéristiques suivantes :

♦ Positifs, un système dynamique est dit positif si toutes ses variables d'état prennent toujours des valeurs positives dans le temps, la positivité est souvent une conséquence de la nature du système.
♦ Linéaire, nous considérons des systèmes dont les variables continues sont linéaires au sens de l'automate hybride qui sera défini plus tard, *i.e.* elles évoluent suivant des équations différentielles du type $\dot{x} = k$, où k est une constante rationnelle.
♦ La partie discrète contrôle la partie continue, dans le sens où la configuration de la partie discrète est complètement autonome et influe sur la partie continue.

Il existe dans la technologie et l'industrie de nombreux exemples de SaFC, nous pouvons citer par exemple : une machine qui contrôle un flux de production, un feu tricolore qui contrôle un flux de véhicules, une décision de routage qui contrôle le flux d'écoulement de messages, …etc. Nous présenterons dans la suite un exemple simple de SaFC, il s'agit d'une section de route contrôlée par des feux de signalisation.

Exemple 1.1. Considérons une section de la route qui peut tolérer un nombre maximum de 150 véhicules (en supposant que la distance moyenne entre deux véhicules est L= 4m). Nous supposons que la section a une rampe d'entrée, les deux entrées de la section sont contrôlées par des feux de signalisation. Les véhicules peuvent accéder à la section avec une vitesse moyenne de 30km/h à partir de deux entrées et sortent avec une vitesse moyenne de 48km/h. L'exemple est présenté dans la Figure suivante :

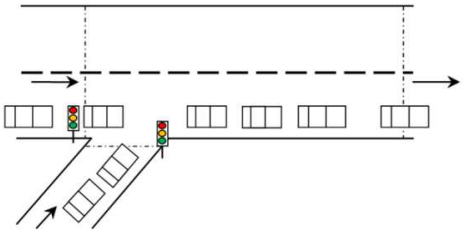


Figure 1.5. Une section de la route.

❑

Après avoir fixé la notion des SDHs, ses différentes classes par quelques exemples et la sous classe considérée durant notre travail de thèse, nous examinons les approches, les outils de modélisation, de simulation et d'analyse des SDHs. Dans la littérature, on trouve plusieurs approches théoriques, permettant de modéliser les systèmes hybrides [Allure *et al*, 93], [Henzinger, 96] [Tavernini, 87] [Stiver, 92] [Buisson et Lu, 94] [Tittus, 95] [Branicky,96] [Mosterman et Biswas, 00] [Lygeros *et al.*, 03] [De Schutter *et al.*, 03], d'analyser leurs évolution [Allure *et al*, 95], [Favela, 99], [Asarin *et al*, 00] [Asarin, 04] [Asarin *et al.*, 07] [Aswani et Tomlin, 07]. Une brève présentation de ces approches, les outils de modélisation, de simulation et d'analyse est donnée ci-dessous.

1.3. Modélisation des SDHs

La modélisation des systèmes dynamiques hybrides doit donc décrire deux comportements : d'une part, la dynamique continue généralement représentée par un système d'équations différentielles et algébriques et d'autre part, la dynamique discrète représentée par un ensemble d'états et de transitions. Les études menées pour concilier les composantes continue et discrète ont conduit à de nombreux formalismes. Il est évidemment impossible de passer en revue la totalité des approches proposées. Par conséquent, nous invitons le lecteur à lire les articles référencés pour plus d'informations. Selon l'approche de modélisation adoptée, une classification a été mise en place [Chombart *et al.*, 96] [Flaus, 98] [Zaytoon, 01].

1.3.1. Approches de modélisation des SDHs

La difficulté du choix adéquat de la méthode de modélisation a entrainé l'existence de plusieurs approches de modélisation des SDHs. Le principal critère de sélection est lié à la problématique considérée. Les modèles proposés sont souvent une extension des modèles existants [Zaytoon, 01] qu'ils soient continus [Mosterman, 97] [Mosterman, 02], discrets [Alur, 99] [Chouikha et Schnieder, 98] [David et Alla, 01], ou mixte [Champagnat, 97] [Villani *et al.*, 05] [Villani *et al.*, 07]. Nous pouvons organiser les approches de modélisation des SDHs en trois classes : 1) les approches basées sur une extension des modèles continus; 2) celles basées sur une extension des modèles discrets et 3) celles qui sont mixtes.

1.3.1.1. Approches basées sur une extension des modèles continus

Le principe de cette approche consiste soit à éliminer la composante discrète du système, soit à la transformer en des équations différentielles [Kurovszky, 02]. De ce fait, le système hybride

est présenté comme un système ne comportant que des équations algébriques différentielles linéaires ou non linéaires. L'avantage de cette approche est que l'on revient à des méthodes classiques d'analyses des systèmes continus linéaires ou non linéaires. L'inconvénient majeur de cette modélisation réside dans le fait de ne pas représenter explicitement l'évolution discrète pour l'utilisateur, autre inconvénient non négligeable est la complexité des équations obtenues.

1.3.1.2. Approches basées sur une extension des modèles discrets

Contrairement à l'approche continue, cette approche est purement discrète, elle consiste à remplacer la dynamique continue du système hybride par une évolution discrète, ou à faire une approximation de l'évolution continue de façon à ce que le système hybride soit représenté uniquement par les évènements qui le caractérisent. Les travaux de Puri présentent une méthode directe afin d'obtenir un modèle événementiel du système hybride qui consiste à découper l'espace d'état continu en plusieurs régions, associées chacune à un état discret [Puri *et al*, 96]. Toutefois, ce concept de modélisation reste confronté au compromis entre la précision et le nombre d'états discrets rapidement explosif.

1.3.1.3. Approches mixtes

L'approche mixte consiste à regrouper les aspects continus et discrets dans le même formalisme de modélisation. Cette approche a engendré l'apparition de nouveaux modèles hybrides à partir des modèles continus et discrets. Nous citons comme exemple les réseaux de Petri hybrides et les automates hybrides obtenues à partir des modèles discrets et les modèles MLD (*Mixed Logical Dynamical*) obtenue par des modèles continus [Bemporad et Moriari, 99]. L'idée de base de cette dernière approche est d'introduire des variables auxiliaires qui permettent de modéliser les relations existantes entre les parties continues et discrètes. Ainsi le passage à la partie discrète nécessite l'ajout de variables logiques. Pour la partie correspondant au passage discret/continu, des variables auxiliaires sont ajoutées.

L'aspect événementiel influe sur le modèle continu par la validation de certaines des équations continues en fonction de l'état discret actif et l'aspect continu agit sur le modèle événementiel en validant ou en forçant le franchissement de certaines transitions.

Parmi les outils de modélisation résultant de cette approche mixte, on trouve :

- les automates hybrides ([Alur *et al*, 93], [Nicollin *et al*., 93]) qui représentent le modèle formel fondamental de cette approche.
- Les automates hybrides rectangulaires [Henzinger *et al.,* 98],
- les automates hybrides linéaires [Müller et Stauner, 00],
- les réseaux de Petri hybrides [Alla et David, 98] [David et Alla, 04] [David et Alla, 10],
- les statecharts hybrides. Ils apportent des solutions aux problèmes posés par la spécification des modèles [Harel, 87] [Harel *et al*, 87] [Mendler et Lüttgen, 01], en particulier pour la structuration hiérarchisée [Kesten et Pnueli, 92],
- les bonds graphs hybrides [Mosterman, 97],… etc.

1.3.2. Outils de modélisation des systèmes dynamiques hybrides.

Dans ce qui suit, nous allons insister sur les outils dans lesquels nous avons trouvé des réponses à notre problématique et pour atteindre les objectifs que nous nous sommes fixés dans notre travail de thèse.

1.3.2.1. Les réseaux de Petri

Le terme « réseaux de Petri » (RdP) désigne une famille de graphes orientés, munis d'un formalisme mathématique qui fait intervenir la manipulation des nombres entiers ou réels positifs ainsi que l'algèbre linéaire. Les réseaux de Petri ont connu depuis leur invention en 1962 par Carl Adams Petri [Petri, 62] un réel succès en raison de leur simplicité mathématique, des avantages de leur représentation graphique et de leur compacité.

Ils représentent un formalisme puissant pour la modélisation et l'analyse des systèmes à événements discrets (SED), comme les systèmes de télécommunication, les réseaux de transports, les systèmes automatisés de production…etc. Plusieurs travaux existent sur les réseaux de Petri, leurs fondements théoriques et leurs applications pratiques. Nous citons cette liste non exhaustive des articles et ouvrages les plus importants ; G.W. Brams [Brams, 82], [Brams, 83], T. Murata [Murata, 89], R. David et H. Alla [David et Alla, 92], J.L. Peterson [Peterson, 81]. Leur représentation graphique permet de visualiser d'une manière naturelle le parallélisme, la synchronisation, le partage de ressources et le non-déterminisme. Leur représentation mathématique permet d'établir les équations d'état à partir desquelles il est possible d'apprécier les propriétés du modèle et les comparer au comportement du système modélisé [Bonhomme, 01].

Plusieurs classes de réseaux de Petri ont été développées et étudiées. Parmi celles-ci, on distinguera les réseaux de Petri autonomes, les réseaux de Petri temporels, les réseaux de Petri continus, les réseaux de Petri hybrides. Nous utilisons ici le modèle RdP hybride élémentaire, qui est une combinaison d'un RdP T-temporel et d'un RdP continu à vitesse constante (RdPCC). Ces derniers seront abordés dans ce document.

A. Les réseaux de Petri autonomes

Informellement, un RdP autonome est un graphe biparti, c'est-à-dire avec deux types de nœuds, les places (représentées par des cercles) et les transitions (représentées par des barres), des arcs permettent de relier une place à une transition ou une transition à une place. Un poids (nombre entier strictement positif) est affecté à chaque arc, ce poids vaut 1 quand ce n'est pas précisé. Le RdP autonome est dit ordinaire si les valeurs de tous ses poids valent 1 et il est dit généralisé dans le cas contraire. L'ensemble des places ainsi que l'ensemble des transitions sont finis et non vides. Chaque place contient un nombre entier (qui peut être nul) de jetons ou marques, c'est le mouvement de ces jetons entre les places qui décrit la dynamique du système. Le marquage d'un RdP autonome est un vecteur dont la dimension est égale au nombre de places

et dont les composantes sont des entiers positifs ou nuls. D'une manière plus formelle les RdP autonomes sont définis de la manière suivante :

Définition.1.1. Un RdP autonome marqué est composé de deux parties distinctes :
1- Une partie *statique* (structurelle) représentée par un quadruplet $R = (P, T, Pré, Post, M_0)$ tel que :

- ♦ P est un ensemble fini et non vide de places ;
- ♦ T est un ensemble fini et non vide de transitions. Les ensembles P et T sont disjoints $P \cap T = \emptyset$;
- ♦ *Pré* est l'application d'incidence avant, telle que :
  $Pré : (P \times T) \rightarrow N$
  $(P_i , T_j) \rightarrow Pré (P_i , T_j)$ = Poids de l'arc reliant la place P_i à la transition T_j ;
- ♦ *Post* est l'application d'incidence arrière, telle que :
  $Post : (P \times T) \rightarrow N$
  $(P_i , T_j) \rightarrow Post (P_i , T_j)$ = Poids de l'arc reliant la transition T_j à la place P_i ;
- ♦ $M_0 : P \rightarrow N$
  $P_i \rightarrow M_0(P_i)$ est le marquage initial de la place P_i.

2- Une partie *dynamique* (comportementale) consiste à faire évoluer le marquage. Nous adopterons par la suite les notations suivantes :

- ♦ $^\circ T_j$ (T_j°) représentera l'ensemble des places d'entrée (sortie) de la transition T_j.
- ♦ $^\circ P_i$ (P_i°) représentera l'ensemble des transitions d'entrée (sortie) de la place P_i.

❑

Les transitions dans un RdP autonome modélisent les événements dont l'occurrence change l'état du système et chaque état est modélisé par un marquage particulier du RdP autonome. Ce marquage change chaque fois qu'une transition est franchie (Figure 1.6). Le franchissement d'une transition est conditionné par sa validation. Les définitions de la validation, du franchissement d'une transition et l'ensemble de marquages accessibles sont formalisées ci-dessous :

Définition 1.2. Une transition T_j est dite validée par le marquage M, si le marquage M satisfait :

$$\forall P_i \in {}^\circ T_j , \ \frac{m_i}{Pré(P_i, T_j)} \geq 1$$

T_j est dite q-validée par le marquage M, ce qui signifie qu'elle à la possibilité d'être franchie β fois d'un seul coup, ($\beta \leq q$), si :

$$\forall P_i \in {}^\circ T_j , \ \min\left(\frac{m_i}{Pré(P_i, T_j)} \right) = q$$

q est un entier positif, il est appelé *degré de validation* de la transition T_j.

❑

Définition 1.3. Soit M un marquage d'un RdP et T_j une transition validée par le marquage M. *Franchir la transition* T_j consiste à :

- ♦ Retirer $Pré (P_i, T_j)$ jetons de toute place $P_i \in {}^\circ T_j$.

♦ Ajouter *Post* (P_i, T_j) jetons à toute place $P_i \in T_j°$.

Le franchissement de la transition T_j provoque *le passage du marquage M* au marquage *M'*, ce qui est noté : $M\,[T_j\rangle\, M'$. Le marquage *M'* est donné par :

$$\forall P_i \in P : M'(P_i) = M(P_i) - Pré(P_i, T_j) + Post(P_i, T_j)$$

❑

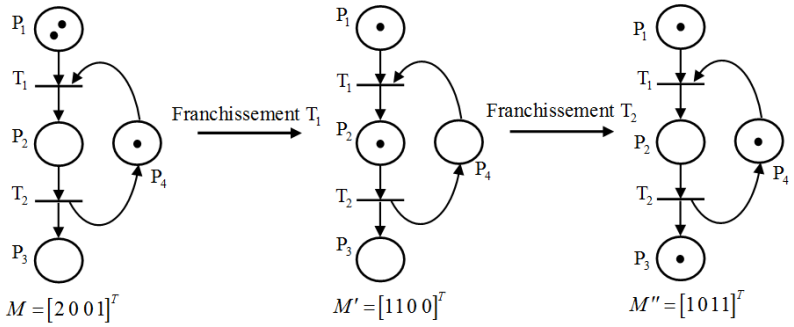
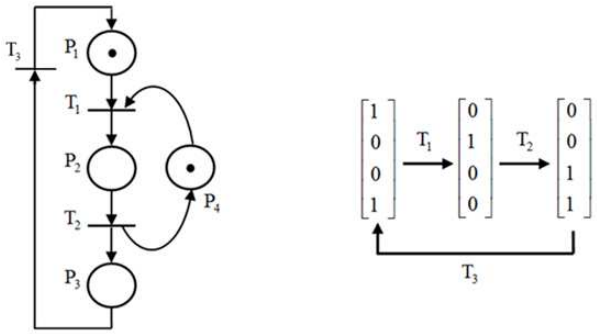


Figure 1.6. Evolution du marquage d'un réseau de Pétri.

Définition 1.4. Soit un RdP autonome marqué *R* et M_0 son marquage initial. *L'ensemble des marquages accessibles* par *R* à partir de M_0 est l'ensemble des marquages tel qu'il existe une séquence de franchissement *y* menant depuis M_0.

❑

L'ensemble des marquages accessibles est généralement représenté par un graphe, appelé graphe des marquages accessibles du RdP autonome (Figure 1.7.b). Cet outil permet de générer la totalité des états atteignables à partir de l'état initial. Ce graphe peut très rapidement exploser en fonction du nombre de jetons mis en jeu, ce qui réduit beaucoup le nombre de propriétés pouvant être pratiquement étudiée, et cela constitue la limitation majeure dans l'utilisation des RdP autonome.



(a) *(b)*

Figure 1.7. (a) RdP autonome. (b) Graphe des marquages accessibles.

La modification du marquage est exprimée par la matrice d'incidence du RdP autonome qui est définie comme suit :

Définition 1.5. La matrice d'incidence *W* d'un RdP autonome est une application de $P \times T$ dans *Z* définie par :

$$\forall P_i \in P\ , \forall T_j \in T\ , W(P_i, T_j) = Post(P_i, T_j) - Pré(P_i, T_j)$$

❑

B. Les réseaux de Petri T-temporels

Plusieurs extensions temporisées des RdP ont été proposées. En effet le temps peut être associé à chacun des éléments d'un RdP: 1) les places [Sifakis, 79], [Coholahan et Roussopoulos, 83], [Khansa, 97], 2) les transitions [Merlin, 74], [Ramchandani, 74], [Starke, 78], [Ramamoorhy et Ho, 80], [Holliday et Vernon, 87] et 3) les arcs [Walter, 83], ou sur plusieurs de ces éléments en même temps, [Cerone et Maggiolo-Schettini, 99]. Dans [Boyer, 01], on trouve des présentations complètes de différentes extensions temporisées des RdP ainsi que des comparaisons entre ces modèles. La totalité de ces modèles peuvent être scindés en deux familles, les RdP temporises d'une part, qui trouvent leur origine dans le travail de Ramchandani [Ramchandani, 74], et les RdP temporels d'autre part dont l'origine est la thèse de Merlin [Merlin, 74]. De façon informelle, les RdP temporisés utilisent la notion de *durée* fixe à opposer à la notion de *délai* de franchissement pour les RdP temporels.

Les RdP T-temporels (Time Petri nets) ont été introduits par Merlin dans sa thèse [Merlin, 74]. L'idée fondatrice des RdP T-temporels est d'associer un intervalle de temps $[\alpha_j, \beta_j]$ à chaque transition T_j. Si cette dernière est validée de façon continue pendant au moins α_j unités de temps, elle peut être franchie. De plus si elle est validée pendant α_j-β_j unités de temps de manière continue, elle *doit être* franchie. On trouve ici les notions de délais minimal et maximal dans un état au lieu de durée d'un état dans les modèles temporisés. Intuitivement, les RdP temporels généralisent les RdP temporisés et autonomes. C'est cet outil que nous avons retenu dans notre travail.

Définition 1.6. Un RdP T-temporel R_T est un couple $R_T = (R, \text{T-Temp})$ tel que :

- R est un RdP autonome marqué ;
- T-Temp : $T \rightarrow Q^+ \times (Q+ \cup \{\infty\})$

$$T_j \rightarrow \left[\alpha_{js}\,,\,\beta_{js}\right]$$

$[\alpha_{js}\,,\,\beta_{js}]$ est l'intervalle statique de franchissement de la transition T_j.

❑

T-Temp est la fonction d'intervalle statique (Static Interval Mapping), elle associe à chaque transition un intervalle statique de franchissement. Cet intervalle est dit *statique* car pendant l'évolution d'un tel RdP, des occurrences d'intervalles de franchissement *dynamiques* (qui évoluent avec le temps) apparaissent, ces derniers sont notés $[\alpha_j\,,\,\beta_j]$. Considérons le réseau de la Figure 1.8, avec pour origine des temps l'instant où le jeton vient d'arriver dans la place P_1. Pour la représentation fournie, seule la transition T_1 est sensibilisée, son intervalle dynamique correspondra à son intervalle statique [1, 5].

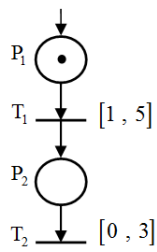


Figure 1.8. RdP T-temporel.

Une transition T_j d'intervalle temporel $[\alpha_j , \beta_j]$ peut être franchie à un instant t si et seulement si $t \in [\theta+\alpha_j, \theta+\beta_j]$, θ représenté l'instant où le marquage a validé la transition T_j. C'est donc l'instant du dernier franchissement d'une transition. Un des problèmes posés par l'analyse des RdP T-temporel est la gestion de la multi-validation (Définition 1.2).

C. Les réseaux de Petri continus et continus à vitesse constante

Parmi les difficultés que présente l'exploitation des RdP est l'augmentation rapide de la complexité du modèle, résultant du fait d'avoir un nombre important de jetons dans les places. Cela a conduit à introduire la notion de réseau de Petri continus (RdPC) où le marquage devient un nombre réel positif, ils ont été définis par David et Alla [David et Alla, 87]. Une place d'un RdPC est dite place continue ou C-place. De même, une transition dans un RdPC est dite transition continue, ou C-transition, elle est validée si toutes ses places d'entrée sont marquées.

Exemple 1.2. Considérons le système de la Figure 1.9 composé de 3 machines MA_1, MA_2 et MA_3 et de deux stocks S_1 et S_2 de capacité C_1 et C_2 respectivement [Alla et David, 98]. Nous admettons qu'il y a toujours des pièces brutes qui arrivent à l'entrée de MA_1 et de la place disponible pour les pièces en sortie de MA_3. Le nombre d'états atteignables pour ce système est $N = 2^3(C_1 + 1)(C_2 + 1)$. Ce nombre peut très rapidement exploser en fonction des valeurs numériques de C_1 et C_2. Soit, pour $C_1 = C_2 = 12$, $N = 1352$. Si le nombre de machines passe à 10 et le nombre de stock à 9, alors pour les mêmes valeurs de C_1 et C_2, N est supérieur à 10^{13} états. ❑

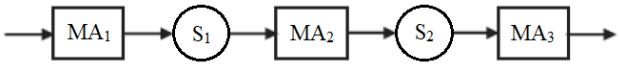


Figure 1.9. Ligne de production.

Les réseaux de Petri continus étendent le marquage dans l'espace d'état des réels en fonction du taux de franchissement des transitions. Ainsi, le processus du franchissement des transitions obéit aux conditions suivantes [Alla et David, 98] :

- ♦ Une transition est considérée sensibilisée dès que le marquage de la place en amont est strictement positif.
- ♦ Dans le cas temporisé, le franchissement d'une transition n'est plus instantané et a nécessité d'associer des vitesses de franchissement aux transitions.

Avec ces principes, la structure est formée d'une transition continue avec une place continue d'entrée et une place continue de sortie. L'ensemble peut être illustré par un sablier, où les marques s'écoulent continument de la place d'entrée vers la place de sortie.

Définition 1.7. Un RdP *continu autonome marqué* (RdPC) est un quintuple $R_C = (P, T, Pré, Post, M_0)$ où on retrouve les mêmes composants d'un RdP autonome marqué (Définition 1.1) avec les différences suivantes :

- ♦ Les applications d'incidences avant et arrière *Pré* et *Post* prennent leurs valeurs dans Q^+ (ensemble des rationnels positifs).

♦ Les composants du vecteur M_0 prennent leurs valeurs dans R^+ (ensemble des réels.

❑

Définition 1.8. Comme pour un RdP autonome marqué, la transition continue (*C-transition*) T_j *est dite validée* par le marquage *M* si ce dernier satisfait :

$$\forall P_i \in {}^\circ T_j\ ,\ \frac{m_i}{Pré(P_i, T_j)} > 0$$

T_j est dite q-validée (avec q un réel positif) si et seulement si :

$$\forall P_i \in {}^\circ T_j\ ,\ \min_i \left(\frac{m_i}{Pré(P_i, T_j)} \right) = q$$

Une C-transition T_j *q*-validée peut être franchie β fois simultanément, où β est un réel inférieur ou égal à *q*. β est dite quantité de franchissement de la C-transition T_j. La notation $[T_j]^\beta$ dénote le franchissement simultané de la transition T_j β fois.

❑

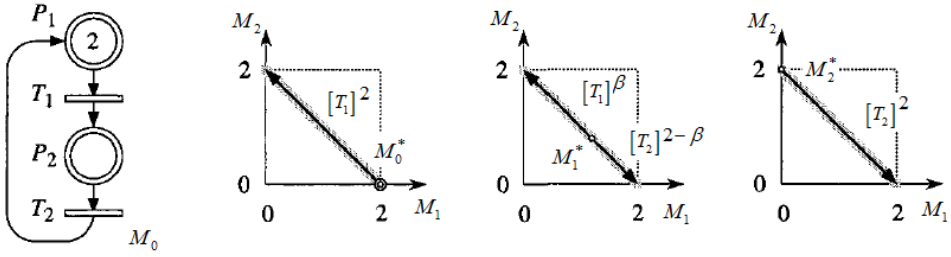
Le marquage atteignable d'un RdPC est non dénombrable, d'où l'impossibilité de construction d'un graphe des marquages accessibles. Ce dernier est remplacé par un graphe d'atteignabilité, où chaque nœud modélise un ou plusieurs marquages. La construction de ce graphe est basée sur la notion du *macro-marquage* [David et Alla, 05].

Définition 1.9. Soit M_k un marquage d'un RdPC, l'ensemble des places *P* peut être divisé en deux sous-ensembles :

♦ $P^+(M_k)$ l'ensemble des places P_i tel que $m_i > 0$
♦ $P^0(M_k)$ l'ensemble des places P_i tel que $m_i = 0$

Un *macro-marquage*, note *M** est l'union de tous les marquages M_k ayant le même ensemble $P^+(M_k)$ de places marquées. Un macro-marquage est caractérisé par son ensemble de places marquées $P^+(M_k)$. Un RdPC ayant n places peut atteindre au maximum 2^n macro-marquages. Nous illustrons ci-dessous le macro-marquage d'un exemple simple de RdPC.

❑



(a) *(b)* *(c)* *(d)*

Figure 1.10. (a) RdPC. de (b) à (d) Illustration de son macro-marquage.

Le RdPC dans la figure 1.10(a) dispose de trois macro-marquages (illustrée dans b à d), à savoir M_0^*, M_1^* et M_2^*, tel que $P^+(M_0^*) = \{P_1\}$, $P^+(M_1^*) = \{P_1, P_2\}$ et $P^+(M_2^*) = \{P_2\}$. Le quatrième macro-marquage est M_3^*, tel que $P^+(M_3^*) = \varnothing$. Dans cet exemple, M_0^* correspond au seul marquage $M_0 = (2,0)$; M_1^* correspond à un nombre infini de marquages $M_\alpha = (\alpha, 2-\alpha)$, $0 < \alpha < 2$; $M_2^* = (0,2)$. Le macro-marquage $M_3^* = (0,0)$ n'est pas accessible.

Dans la littérature, on dénombre, trois configurations de réseaux de Petri continus qui ont été formellement définis : 1) RdPC à vitesse constante (RdPCC) [Alla et David, 97], 2) RdPC à vitesse variable [Alla et David, 97], et 3) RdPC à vitesse asymptotique [Le bail *et al.*, 92]. Ils se différencient par la façon de calculer les vitesses de franchissement des transitions. Nous allons présenter par la suite le modèle que nous avons retenu pour la sous classe de SDH étudiée, à savoir le RdPCC.

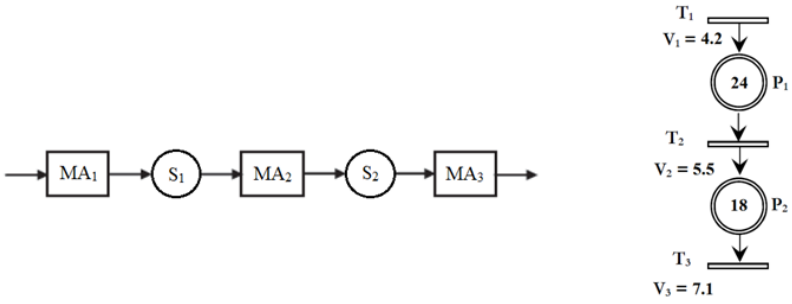
Définition 1.10. Un RdPCC est un couple $R_{CC} = (R_C, V)$ tel que :

- R_C est RdP continu autonome marqué;
- $V : T \rightarrow R^+$

  $T_j \rightarrow V_j$, la vitesse de franchissement maximale de la transition T_j ;

❑

Exemple 1.3. A titre d'illustration, considérons le système de la Figure 1.11(a) composé de 3 machines MA_1, MA_2 et MA_3 et de deux stocks S_1 et S_2 de capacité C_1 et C_2 respectivement [Alla et David, 98]. Nous admettons qu'il y a toujours des pièces brutes qui arrivent à l'entrée de MA_1 et de la place disponible pour les pièces en sortie de MA_3, dont les vitesses de traitement des pièces sont respectivement V_1, V_2 et V_3 (pièces/seconde). Le système est modélisé par le RdPCC de la Figure 1.11(b) pour les valeurs numériques suivantes : $V_1 = 4.2$, $V_2 = 5.5$ et $V_3 = 7.1$. Les marquages des places P_1 et P_2 représentent le nombre des pièces dans les stocks. Les vitesses associées aux transitions du RdPC modélisent les vitesses de production des machines.



(a) *(b)*

Figure 1.11. (a) Ligne de production. (b) Son RdPCC.

A l'instant initial, les trois transitions sont fortement validées, une transition est dite fortement validée si toutes ses places d'entrée sont marquées et elles sont franchies à leurs vitesses maximales. Le marquage des places P_1 et P_2 évolue suivant les équations suivantes :

$$m_1(t + dt) = m_1(t) + (V_1 - V_2).dt$$
$$m_2(t + dt) = m_2(t) + (V_2 - V_3).dt$$

Puisque $M_0 = [24\ 18]^T$

$$m_1(t) = 24 - 1.3\ t$$
$$m_2(t) = 18 - 1.6\ t$$

Ces équations restent vraies aussi longtemps que $m_1 > 0$ et $m_2 > 0$. A $t = 11.25$s m_2 s'annule, ce qui empêche T_3 d'être franchie à sa vitesse maximale. Mais puisque la place P_2 est toujours alimentée à une vitesse de 5.5 pièces/s par la transition T_2, T_3 est aussi franchie avec cette même vitesse, qui n'est plus sa vitesse maximale, T_3 est dite faiblement validée (Sa seule place d'entrée

n'est pas marquée mais elle est alimentée). A t = 15.38s m_1 s'annule, ce qui empêche T_2 d'être franchie à sa vitesse maximale, elle sera franchie à la vitesse de T_1.

Le vecteur des vitesses instantanés du RdPCC en Figure 1.11(b) est donne par :

$$\begin{pmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{pmatrix} : \begin{cases} [V_1\ V_2\ V_3]^T & \text{pour } 0 \le t \le 11.25 \\ [V_1\ V_2\ V_2]^T & \text{pour } 11.25 \le t \le 15.38 \\ [V_1\ V_1\ V_1]^T & \text{pour } t \ge 15.38 \end{cases}$$

L'évolution du marquage en fonction du temps est illustre en Figure 1.12.

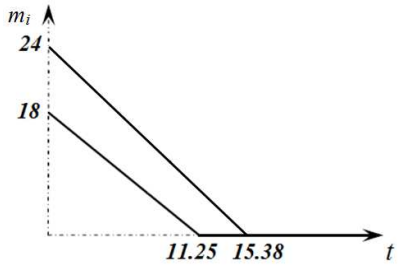


Figure 1.12. Illustration du comportement du RdPCC en figure 1.11(b).

❑

L'état d'un RdPCC est caractérisé soit par le vecteur des franchissements instantanés ou, par dualité, par les dérivés des marquages. Le seul événement susceptible de changer l'état d'un RdPCC est que le marquage d'une place devienne nul. Le comportement de ce modèle est généralement représenté par un graphe d'évolution qui est construit comme un RdP où chaque place représente un IB-état (ou IB-state) et où à chaque transition est associé l'événement qui provoque le changement de l'état des vitesses (un changement de marquage). Il est évident que chaque sommet correspond à un macro-marquage. Le graphe d'évolution du RdPCC a toujours un nombre fini de nœuds même si le RdPCC est non borné. À titre d'illustration le graphe d'évolution du RdPCC de la Figure 1.11(b) est représenté dans la Figure 1.13. Le graphe d'évolution d'un RdPCC est facilement translatable en un automate hybride particulier, ceci sera détaillé dans la suite.

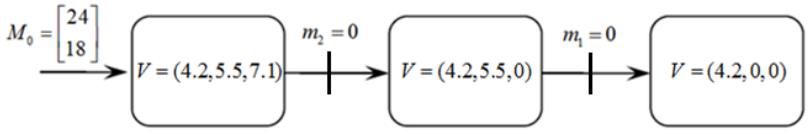


Figure 1.13. Graphe d'évolution du RdPCC en figure 1.11(b).

Définition 1.11. Un IB-état (IB-state ou Invariant Behavior state) d'un RdPCC est une phase d'évolution où le vecteur de vitesses instantanées reste constant. Autrement dit, tant qu'on est dans un même IB-état, les marquages évoluent de façon linéaire.

❑

D. Les réseaux de Petri Hybrides

Ni le formalisme *RdP autonome,* ni le formalisme *RdPC* ne permettait de représenter les SDHs. Cela a ainsi conduit à la définition d'une nouvelle extension : les *réseaux de Petri hybrides (RdPH)* [Le Bail *et al.,* 91]. Un RdPH est un modèle combinant un RdP discret et un RdP continu dans lesquels coexistent des places et des transitions continues (C-places et C-

transitions) et discrètes (D-places et D-transitions). Plusieurs applications utilisant ce formalisme ont été réalisées. Citons par exemple la modélisation d'un système de production de la société Motorola [Alla *et al.*, 92], la modélisation d'un réseau d'eau potable constitué de réservoirs interconnectés par des vannes [Alla, 94] ou la modélisation d'un système de production d'énergie hydraulique [David, 91].

Le réseau de Petri hybride de la Figure 1.14 modélise un système de fabrication qui produit des pièces par lots de 5. À la fin de la production de 2 lots de 5 pièces, un nouveau cycle de production est entamé.

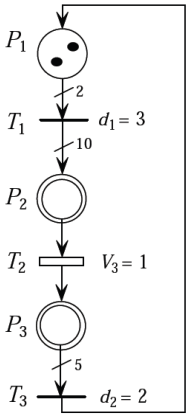


Figure 1.14. Modèle de RdPH d'un système de fabrication par lots.

Le marquage de la place P_1 (D-place) est associé au nombre de lots à l'entrée du système de fabrication. Les durées d_1 et d_2 sont les temps de chargement et déchargement des pièces (D-transitions). La transition T_3 modélise une machine dont la vitesse de production est V_3 (C-transition). Cette machine possède un stock d'entrée et un stock de sortie modélisés par les places P_2 et P_3 (C-places). Le franchissement continu de la transition T_3 correspond à une production continue à la vitesse V_3 quand la place P_2 n'est pas vide. Lorsque P_2 est marquée, le franchissement d'une quantité V_3dt de T_2 correspond à retirer V_3dt marques à P_2 et à ajouter la même quantité à P_3.

Définition 1.12. Un *RdP hybride* est une structure $R_H = (P, T, h, \Sigma, I, Pré, Post, Tempo, V, M_0)$, tel que :

-$P = \{P_1, P_2, \ldots, P_n\}$ est un ensemble de n places, $P = P^C \cup P^D$ avec :

$P^C = \{P_1, P_2, \ldots, P_{nc}\}$ est l'ensemble fini de places continues (ou C-places) ;

$P^D = \{P_{nc+1}, \ldots, P_n\}$ est l'ensemble fini de places discrètes (ou D-places) ;

- $T = \{T_1, T_2, \ldots, T_m\}$ est un ensemble de m transitions, $T = T^C \cup T^D$ avec :

$T^C = \{T_1, T_2, \ldots, T_{mc}\}$ est l'ensemble fini de transitions continues (ou C-transitions) ;

$T^D = \{T_{mc+1}, \ldots, T_m\}$ est l'ensemble fini de transitions discrètes (ou D-transitions) ;

-$h : P \cup T \rightarrow \{D, C\}$ est une application qui désigne les nœuds discrets, $h(x)=D$, et les nœuds continus, $h(x)=C$;

- Σ est un ensemble fini d'événements ;

-I: $T^D \rightarrow \Sigma$ est une fonction qui associe à chaque transition discrète un événement de Σ;

-*Pré* et *Post* désignent respectivement les applications d'incidence avant et arrière ; ces applications doivent satisfaire la condition suivante :

$$\forall (P_i, T_j) \in P^D \times T^C : Pré(P_i, T_j) = Post(P_i, T_j)$$

-*Tempo*:$T^D \rightarrow Q^+$ est une application qui associe à chaque D-transition la durée de sa temporisation.

-*V*: $T^C \rightarrow R^+$ est une application qui associe à chaque C-transition sa vitesse maximale de franchissement :

-M_0 est le marquage initial, les D-places contiennent un marquage entier positif et les C-places contiennent un marquage réel positif.

❑

La technique de modélisation via le modèle RdPH décrit des phénomènes discrets et continus particuliers mais ne représente pas l'influence vice-versa des deux parties du SDH en même temps, soit l'influence de la partie discrète sur la partie continue ou celle la continue sur la discrète. Dans ce contexte, l'utilisation des RdPHs s'avèrent mal adaptés à la sous classe des SDHs considérée dans notre travail. L'utilisation d'une classe des RdPHs, à savoir les RdPH élémentaires, apparaît comme une solution adéquate pour nos besoins de modélisation. Ce modèle représente le premier modèle de base de notre travail de recherche, le deuxième modèle sera présenté par la suite.

E. Les réseaux de Petri Hybrides élémentaires

Les RdP élémentaires constituent une classe particulière de RdPH où il n'y a pas de transformation de marquage, du discret vers le continu ou du continu vers le discret. Le RdPH élémentaire considéré dans notre travail est un modèle combinant un RdP T-temporel et un RdPCC. Le RdP T-temporel contrôle le comportement du RdPCC via des boucles connectant certaines D-places à certaines C-transitions, ce qui signifie que ces dernières ne sont pas validées et par conséquent ne peuvent être franchies que si les D-places sont marquées. Le RdPCC à son tour peut influencer le comportement du RdP T-temporel. Une D-transition T_j peut avoir comme condition de franchissement le marquage d'une C-place P_i qui atteint un seuil *S*.

Graphiquement, ceci est représenté par deux manières soit par une boucle (un arc de P_i vers T_j et un arc de T_j vers P_i) dont le poids est *S* si ce seuil est un seuil supérieur, c'est-à-dire si le marquage de P_i ne peut être supérieur à *S* (Figure 1.15.a). Dans le cas contraire, si le marquage de P_i ne doit pas être inferieur à *S*, un arc inhibiteur est utilisé pour relier T_j à P_i (Figure 1.15.b). Et dans les deux cas le franchissement de T_j ne modifie pas le marquage de P_i.

(a) *(b)*

Figure 1.15. (a) Validation de T_j si $m_i \geq S$. (b) validation de T_j si $m_i < S$.

Définition 1.13. Un *RdPH élémentaire* est un couple (R_H, *IN*) tel que :
-R_H est un RdPH dont les applications *Pré* et *Post* satisfont la condition suivante :

$$\forall(P_i, T_j) \in (P^D \times T^C) \cup (P^D \times T^C) : Pré(P_i, T_j) = Post(P_i, T_j)$$

-*IN* : (P_i, T_j)→R, est une application d'inhibition, si un arc inhibiteur de poids *S* relie la place P_i à la transition T_j, le franchissement de T_j n'est possible que si le marquage de P_i est inferieur à *S*.

❑

Exemple 1.4. Reprenons l'exemple 1.3, et supposons que l'on désire limiter le nombre des pièces dans le stock 2 entre NB_{min} et NB_{max}. Ce système est modélisé par le RdPH élémentaire de la Figure 1.16. Dans ce modèle le franchissement de la C-transition T_3 n'est possible que si la D-place P_4 est marquée. De même franchir la D-transition T_4 (T_5) n'est possible que si le marquage de la C-place P_2 est supérieur à NB_{max} (inferieur à NB_{min}).

❑

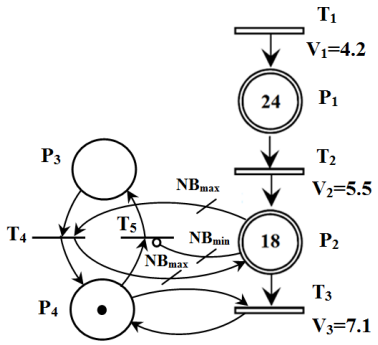


Figure 1.16. RdPH élémentaire du système en figure 1.9.

Le modèle RdPH présenté dans la définition 1.12 est un modèle déterministe dans le sens ou, connaissant son état à l'instant initial, on peut connaitre exactement son état à n'importe quel instant *t* et cet état est unique (à condition de choisir une politique de résolution des conflits). L'état de ce modèle est donné par l'état des parties discrètes et continues. Les événements qui peuvent changer l'état d'un RdPH sont de deux types, ce sont les événements qui peuvent changer l'état d'un RdP T-temporel et ceux qui change l'état d'un RdPCC, à savoir :

- ♦ Une D-transition est franchie ;
- ♦ Le marquage d'une C-place s'annule ;

Comme nous l'avons vu précédemment, un RdPCC peut être défini par son marquage et les vecteurs vitesses instantanées de franchissement. Ceci définit, dans les RdPHs, un état comportemental invariant, c'est l'IB-état d'un RdPH.

Définition 1.14. Un IB-état d'un RdPH représente des phases où l'évolution des marquages est constante et continue. Un IB-état correspond à un intervalle de temps dans lequel:

- ♦ Le marquage des D-places est constant ;
- ♦ Le vecteur de vitesses instantanées des C-transitions est constant;

❑

1.3.2.2. Les automates hybrides

Les Automates Hybrides (AHs) [Alur *et al.*, 93] sont une extension des automates à états finis. Ils représentent des systèmes qui intègrent deux composantes : celle ayant un

comportement discret, modélisée naturellement par un automate à états finis et celle dont le comportement varie de manière continue dans le temps, modélisée par un système algébro-différentiel. Un automate hybride (AH) évolue par une alternance de pas continus, où les variables d'état et le temps évoluent de façon continue, et de pas discrets où plusieurs transitions discrètes et instantanées peuvent être franchies.

D'un point de vue informel et général, un AH apparaît ainsi comme un automate à états fini pilotant un ensemble d'équations différentielles modélisant la dynamique continue du système. L'état de l'automate change pour deux raisons [Lafortune et Cassandra, 98] :

- Le franchissement d'une transition discrète, qui change brusquement la situation et souvent alors l'évolution de l'état continu, voire directement la valeur de cet état (saut). Ce franchissement se produit sur occurrence d'un événement approprié et/ou si une condition devient vraie ;

- L'évolution temporelle qui affecte la valeur du vecteur d'état suivant l'équation différentielle associée à la situation courante. Cette situation reste inchangée.

À chaque sommet q d'un automate hybride on associe une fonction d'évolution F_q, sous la forme $\dot{x} = F_q(x)$ et un prédicat sur la valeur des variables appelé invariant du sommet. À chaque transition T on associe une condition de franchissement, appelée garde, et une affectation qui réinitialise les valeurs des variables continues. Un automate hybride est formellement défini comme suit :

Définition 1.15. Un *automate hybride* est un sextuple $A_H = (Q, X, E, \delta, F, Inv)$ tel que :
- $Q = \{q_1, q_2, \ldots\}$ est un ensemble fini de sommets ;
- $X \in \Re^n$ est un vecteur d'état comportant n variables réelles ;
- E est un ensemble fini d'événements ;
- δ est un ensemble fini de transitions, chaque transition est un quintuple $T = (q, a, g, init, q')$ tel que:
  - ♦ $q \in Q$ est le sommet source ;
  - ♦ $a \in E$ est un événement associe à la transition T ;
  - ♦ g est la garde de la transition T, c'est un prédicat sur X ; la transition T ne peut être franchie que si sa garde g est vérifiée;
  - ♦ *init* est la fonction de réinitialisation qui affecte une expression aux variables continues quand la transition T est franchie;
  - ♦ $q' \in Q$ est le sommet but;

- F est une fonction qui associe à chaque sommet q une fonction continue f_q qui représente l'évolution dynamique du vecteur d'état dans le sommet ;
- *Inv* est une fonction qui associe à chaque sommet q, un prédicat $Inv(q)$, qui doit être vérifié par les valeurs des variables continues lors du séjour de l'automate dans le sommet q.

❑

L'avantage de cette représentation avec l'AH est sa simplicité. Décrivant sans ambiguïté les évolutions possibles d'un SDH, elle sera à la base de l'analyse en vue d'établir des propriétés formelles. À chaque instant, un seul état discret est actif, donc il n'y a qu'un seul jeu d'équations (un seul modèle continu). Le caractère hybride se marque par le fait qu'un événement discret peut entraîner le changement d'état, donc la commutation du jeu d'équations. L'atteinte d'une valeur seuil sur une variable continue peut aussi entraîner un changement d'état discret.

❑

Exemple 1.5. Considérons l'automate représenté dans la Figure 1.17 modélisant un système hybride. Dans ce modèle, l'évolution continue est représentée par des équations différentielles associées aux sommets du graphe. L'évolution événementielle est modélisée par les arcs étiquetés du graphe.

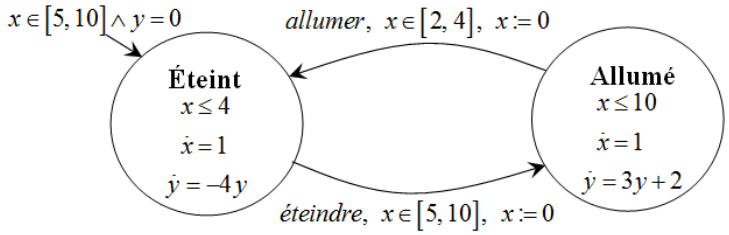


Figure 1.17. Automate hybride.

Les sommets **Éteint** et **Allumé** représentent les états discrets du système où l'évolution continue a lieu. Les prédicats $x \in [2\ ,\ 4]$ et $x \in [5\ ,\ 10]$ sur les arcs traduisent les conditions, pour l'occurrence des événements *allumer* et *éteindre*, respectivement. Les prédicats $x \le 4$ et $x \le 10$ dans les sommets représentent les invariants de l'automate, c'est-à-dire, des conditions imposées aux variables continues du système pour rester dans un état discret (ici les états **Éteint** ou **Allumé**). L'état initial du système est représenté par un arc d'entrée dans le sommet d'origine. L'étiquette de cet arc $x \in [5\ ,\ 10] \wedge\ y = 0$ représente la région de l'espace continue à partir de laquelle la dynamique du système hybride démarre. Les variables x et y évoluent dans le sommet **Éteint**, respectivement le sommet **Allumé**, conformément aux équations différentielles, $\dot{y} = -4y$ et $\dot{x} = 1$, respectivement $\dot{y} = 3y + 2$ et $\dot{x} = 1$, appelées conditions de flux ou les dynamiques[1].

❑

A. Les sous classes d'automates hybrides

L'analyse d'atteignabilité consiste à déterminer l'espace d'état atteignable par l'évolution du SDH étudié, les déférents outils d'analyse des SDHs seront présentés par la suite. Ce problème n'est pas décidable[2] pour un automate hybride sans hypothèses particulières [Henzinger *et al.*, 98]. Il faut alors apporter des restrictions pour avoir des sous classes pour lesquelles certaines propriétés sont décidables. Dans la suite, nous allons présenter quelques sous classes d'AH qui ont été explorées dans la littérature, pour lesquelles l'analyse d'atteignabilité est décidable. Ces sous classes ont été étudiées dans le but d'alléger la structure du modèle initial, afin de simplifier son analyse et sa vérification. Parmi les modèles proposés, nous citons :

[1] Nous désignons par $\dot{x}$ la dérivée du premier ordre par rapport au temps de la variable x.
[2] Pour comprendre la décidabilité, nous invitons le lecteur à consulter les références:[Mayr, 81], [Kosaraju, 82] et [Reutenauer, 89].

♦ les automates hybrides linéaires (AHLs) [Alur *et al.*, 93]: un automate hybride est dit linéaire si les conditions de flux, des invariants, des gardes, sont définies par des expressions linéaires sur l'ensemble des variables.

L'importance de cette sous classe est due au fait que plusieurs problèmes intéressants, tels que l'analyse d'atteignabilité [Henzinger *et al.*, 98], la synthèse de contrôleurs [Spathopoulos, 00], la synthèse du diagnostiqueur [Derbel, 09], le model-checking [Henzinger et Majumdar, 00] sont décidables pour les automates hybrides linéaires. Ce modèle représente le deuxième modèle de base de notre travail de recherche. Une description plus détaillée et formelle de la sous classe, les automates hybrides linéaires, sera donnée en Chapitre 3.

La Figure 1.18 représente la syntaxe d'un automate hybride linéaire à entrées/sorties.

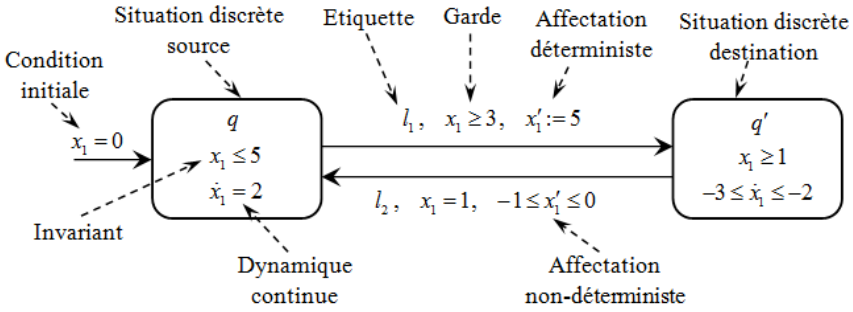


Figure 1.18. Syntaxe d'un automate hybride linéaire.

♦ les automates hybrides rectangulaires (AHRs) [Kopke, 96][Henzinger *et al.*, 98]: c'est une sous classe des automates hybrides linéaires. La condition de flux dans ce modèle est définie sous la forme de prédicats rectangulaires de la forme $\dot{x} \in [a \ \ b]$, pour chaque variable x du modèle. De même, les invariants, les gardes, la condition initiale sont décrits par des prédicats rectangulaires. Elle a été largement étudiée dans la littérature [Spathopoulos, 00] [Henzinger et Majumdar, 00].
♦ les automates hybrides rectangulaires initialisés (AHRIs) [Henzinger *et al.*, 98]: c'est une sous classe des automates hybrides rectangulaires. Dans ce modèle, chaque variable qui change de condition de flux, suite au franchissement d'une transition entre deux sommets, doit être réinitialisée.

B. Outils d'analyse des Automates hybrides

L'analyse et la vérification des systèmes en utilisant les automates hybrides représentent un thème central dans le cadre de l'étude des SDHs. Il s'agit de vérifier des propriétés qualitatives et quantitatives sur un système en utilisant des méthodes formelles. La vérification repose, d'abord, sur une modélisation du comportement du SDH à vérifier par un AH. Ensuite, l'application d'une analyse des propriétés qualitatives à travers les techniques de model-checking [Henzinger *et al.*, 97], ou des propriétés quantitatives à travers l'analyse d'atteignabilité [Alur *et al.*, 93] [Alur *et al.*, 95]. Cette analyse est généralement basée sur le calcul d'un ensemble d'états atteignables à partir d'un ensemble d'états donné. Il existe deux types d'évolutions à partir d'un état initial, à savoir : une évolution continue, en restant dans le même sommet et en laissant le

vecteur d'état évoluer suivant la fonction d'évolution du sommet, ou de manière discrète en franchissant une transition discrète.

Définition 1.16. Soit un ensemble d'état $(q, \aleph)$ ou $q \in Q$ et $\aleph$ un espace d'état continu. On définit l'ensemble des *successeurs continus* de $(q, \aleph)$, qu'on note $Succ_{Cont}(q, \aleph)$, comme suit :

$$Succ_{cont}(q, \aleph) = \left\{ (q, x') / \exists x \in \aleph, \exists t > 0, \ (q, x) \xrightarrow{t, f_q} (q, x') \right\}$$

❑

Définition 1.17. Soit une transition *T*, dont la garde est *g* et la fonction de réinitialisation *init*, reliant *q* à *q'* et $\aleph$ un espace d'état continu, où $q, q' \in Q$. On définit l'ensemble des *successeurs discret* de $(q, \aleph)$, par rapport à la transition *T*, qu'on note $Succ_{dis}(q, \aleph)$, comme suit :

$$Succ_{dis}(q, \aleph) = \left\{ (q', x') / \exists x \in \aleph \cap g \wedge x' \in init \cap inv(x') \right\}$$

❑

Plusieurs outils d'analyse appelés model-checkers ont été développés pour la vérification automatique des propriétés des AHs. Ces model-checkers contiennent des procédures d'analyse de l'atteignabilité. Parmi ces model-checkers, nous citons :

CMC (*Compositional Model-Checking*) a été développé au laboratoire Spécification et Vérification de l'ENS de Cachan, vise à vérifier des réseaux d'automates temporisés en utilisant comme langage de spécification la logique Lv.

UPPAAL a été développé aux universités d'Uppsala en Suède et d'Aalborg au Danemark. Ce model-checker est assez facile d'utilisation puisqu'il intègre une interface graphique très conviviale. Il permet de modéliser des automates temporisés communiquant par synchronisation. La logique temporelle utilisée est un fragment de *TCTL*, ce qui ne permet que la vérification de propriétés d'accessibilité. Le point fort d'*UPPAAL* est le temps de calcul rapide.

KRONOS a été développé au laboratoire **VERIMAG** de l'université de Grenoble. Ce model-checker modélise des réseaux d'automates synchronisés et utilise la logique *TCTL*. Un automate temporisé est exprimé sous une forme textuelle. Le point fort de *Kronos* est l'expressivité de sa logique : *Kronos* implante un algorithme de model-checking pour la logique temporelle *TCTL*.

HYTECH (*HYbrid TECHnology*) a été développé à l'université de Berkeley. Cet outil vise à vérifier des réseaux d'automates hybrides très généraux. Il permet même d'analyser des systèmes paramétrés. Il vérifie principalement des propriétés d'accessibilité et de sûreté. La représentation symbolique qui est utilisée est celles des polyèdres convexes, c'est-à-dire des intersections d'hyperplans.

PHAVer (*Polyhedral Hybrid Automaton Verifyer*) est un outil de vérification des automates hybrides plus récent, développé par Goran Frehse [Frehse, 05], qui vise à combler les lacunes de *Hytech*. Le problème des dépassements de capacité est réglé par l'utilisation de la Parma Polyhedra Library qui permet un calcul exact sur les polyèdres. Le but de *PHAVer* est également de permettre l'analyse de systèmes plus complexes que ceux pris en charge par *HyTech*. Il donne

l'espace des états atteignables pour chaque sommet sous la forme d'inégalités entre les différentes variables continues. Cette formalisation analytique sera utilisée pour le calcul de l'automate atteignable dans le Chapitre 4.

Le fait de disposer d'un modèle permet de simuler le comportement du SDH pour le prévoir, l'influencer et chercher à l'optimiser. Il permet aussi, sous certaines conditions, de l'analyser pour détecter à priori d'éventuels dysfonctionnements. Nous allons présenter par la suite les outils de simulation et d'analyse des SDHs.

1.4. Outils de simulation des systèmes dynamiques hybrides

Les systèmes complexes se composent d'un grand nombre de composants agissants les uns sur les autres avec des comportements non-linéaires et hybrides. La construction précise des modèles efficaces de simulation pour ces systèmes est une tâche difficile. Des chercheurs ont adopté le composant orienté objet [Lee *et al.*, 03] pour modéliser de grands systèmes hybrides. Les modèles mathématiques indiquent les différents comportements des composants et les modèles formels du calcul définissent des interactions entre les composantes, qui fournissent la base pour développer des environnements efficaces pour simuler le comportement des systèmes hybrides.

De nombreux outils de simulation ont vu le jour, ils dépendent généralement du modèle utilisé pour représenter le SDH (bond-graph, automate hybride, etc.), de leur objectif (orienté vers une classe d'application ou au contraire à vocation générale) et du type de simulation utilisée (séquentielle ou globale). Nous pouvons citer quelques outils de simulation comme :

HYSDEL (*Hybrid System Description Language*) est un langage basé sur la modélisation MLD du système hybride. Cet outil peut être employé pour l'analyse et la synthèse de la commande du SDH [Potocnik *et al.*, 03].

YAHMST (*Yet another Hybrid Simulation Tool*) a été développé au laboratoire d'automatique de Grenoble et appliqué au cas d'un processus batch complexe. Il est implanté à l'aide du langage de programmation orienté objet Java et permet la structuration hiérarchique du modèle. Basé sur l'association à un mode discret d'un jeu d'équations continues, il intègre au solveur un détecteur d'événements, facilitant le calcul des instants de commutation [Flaus et Thévenon , 00].

HyBrSim (*Hybrid Bond Graph Simulator*) est un outil réalisé sur la base des bonds graphs hybrides. Cet environnement expérimental de modélisation permet d'établir un cadre formel au SDH considéré [Mosterman, 02].

HSML (*Hybrid Systems Modeling Language*) a pour but de définir formellement le système et fournir une base pour les langages «front ends» pour des environnements de simulation des systèmes hybrides. Le langage HSML permet une construction hiérarchique et modulaire des modèles ; définition du temps continu, du temps discret et composants à base logique ; établissement du programme prioritaire des composants en temps discret ; mécanismes pour la manipulation d'état-événement ; traitement des conflits ; vérification rigoureuse de type et de

gamme ; une base sémantique stricte qui permet la vérification et la validation du modèle [Taylor, 94].

HSCAP (*Hybrid Sequential Causal Assignment Procedure*) permet la mise à jour dynamique de l'information causale [Daigle *et al.*, 06]. Il est constitué de structures de diagramme de bloc reconfigurable.

1.5. Analyse des systèmes dynamiques hybrides

La simulation est une approche expérimentale permettant d'analyser les propriétés du système. Cependant, elle ne permet pas en général de considérer toutes les évolutions du fait de l'explosion combinatoire du nombre de situations possibles. Ainsi, de nombreux travaux ont été consacrés à la vérification formelle en parallèle avec les travaux sur les propriétés structurelles des modèles comme la stabilité, l'observabilité et la commandabilité.

1.5.1. Vérification des systèmes dynamiques hybrides et accessibilité

La vérification formelle des propriétés est un domaine très important dans l'analyse des SDHs. Elle permet de s'assurer que des problèmes tels que le blocage ou le non déterminisme dû aux transitions discrètes ne se posent pas lors de l'exécution du modèle. Notamment, si une séquence infinie d'événements se produit en un temps fini, comme dans le cas où deux phases bouclent sur elles-mêmes, alors l'exécution ne s'arrêtera jamais. Cela signifie que le modèle représentant le système n'est pas adapté ou que le système lui-même est mal conçu.

La vérification de propriétés de sûreté et l'analyse de l'accessibilité des SDH sont en général des problèmes non décidables [Alur *et al.*, 95], toutefois quelques techniques et algorithmes sont de plus en plus utilisés. La vérification a été principalement appliquée à des systèmes modélisés par des automates hybrides. Il s'agit d'une vérification du modèle au sens informatique en utilisant la logique temporelle ou le model-checking [Henzinger *et al.*, 97]. Parmi les outils informatiques dédiés à la vérification, nous pouvons citer *HyTech* (*HYbrid TECHnology*), conçu sur la base d'automates hybrides linéaires [Alur *et al.*, 95].

1.5.1.1. Vérification des systèmes dynamiques hybrides

Une grande part des travaux sur la vérification des SDH provient de ceux utilisés pour la vérification des systèmes à événement discrets (SED). A ce titre, nous pouvons citer deux principales méthodes de vérification [Guéguen *et al.*, 08] : la première est basée sur une abstraction du comportement continu par un SED et la seconde tend à adapter les méthodes vouées aux SED à la dynamique continue.

La vérification basée sur l'abstraction des événements discrets consiste à définir les régions de l'espace d'état hybride pour construire le modèle discret. Chacune de ces régions est alors associée à un état discret. Les transitions discrètes sont alors établies et les régions sont dédoublées d'une manière itérative selon les considérations de l'accessibilité. Le processus

itératif du modèle discret s'arrête quand il n'y a plus de changement de l'ensemble des secteurs hybrides dans deux itérations consécutives. Le modèle discret est alors une bi-simulation du système hybride qui peut être employé pour la vérification.

Une des difficultés de cette approche est le calcul de l'ensemble hybride de prédécesseurs d'un secteur. La difficulté principale est liée à l'algorithme itératif où il est impossible de garantir sa convergence. Afin de pallier cette difficulté, une décomposition itérative de la région est généralement arrêtée à une certaine étape. Le modèle discret résultant est alors une abstraction du système hybride [Alur *et al.*, 03] et [Kloetzer et Belta, 06].

1.5.1.2. Vérification basée sur l'accessibilité des systèmes hybrides

Une deuxième famille d'approches s'intéresse à la vérification des propriétés d'accessibilité des systèmes hybrides. Cette restriction aux propriétés d'accessibilité peut sembler être importante. Tant que l'espace d'états des SDH inclut implicitement le temps, plusieurs propriétés importantes comme celles de la sûreté peuvent être exprimées comme celles de l'accessibilité. La vérification de l'accessibilité est intégrée dans les outils comme *HyTech* [Henzinger *et al*., 97] et *PHAVer* [Frehse, 05].

1.5.2. Stabilité des systèmes dynamiques hybrides

Des travaux sur l'analyse des propriétés structurelles des SDHs concernent essentiellement l'étude de la stabilité des SDHs qui ne concerne essentiellement que la partie continue de celui-ci. La plupart de ces travaux tendent à étendre les approches classiques comme celle de Lyapunov pour l'étude de la stabilité [Pettersson et Lennartson, 02]. En effet, la stabilité des sous-systèmes d'un SDH ne garantit pas la stabilité du système SDH global. De même, l'instabilité des sous-systèmes du SDH sous l'effet d'une commutation particulière peut provoquer la stabilité globale du système [Branicky, 94].

D'énormes efforts et plusieurs travaux ont été effectués dans le but de la mise en point des concepts de base concernant la stabilité des SDHs [Liberzon, 03], [Peleties et Morse, 91], [Branicky, 93], [Branicky, 97], [Branicky, 98]. Le premier diagnostic des problèmes de la stabilité a été effectuée dans [Liberzon et *al*, 99] où les auteurs ont résumé ces derniers en trois points énoncés comme suit:

1: Trouver la condition qui garantit la stabilité asymptotique du système pour n'importe quel signal de commutation (Arbitrary Switching Sequence).
2: Identifier les classes de signaux de commutations pour lesquelles le système est asymptotiquement stable.
3: Construire un signal de commutation pour lequel le système est asymptotiquement stable.

Les résultats de ces recherches ont conduit à d'autres notions de stabilité. Celles-ci spécifient les conditions nécessaires et les propriétés de la stabilité des SDHs. Nous trouvons un état de l'art assez complet sur la stabilité des SDHs dans [Décarlo *et al.*, 00]. Dans ce dernier, l'auteur a

mentionné que dans la majorité des cas le contexte de l'analyse de la stabilité des SDHs se basent sur l'approche de Lyapunov, ou plus spécifiquement le principe s'appuie sur la fonction de Lyapunov quadratique.

1.6. Conclusion

Nous avons organisé ce chapitre autour de quatre grands axes dans le but de donner une vue sur la caractérisation, la modélisation, la simulation et l'analyse des SDHs. En effet, nous avons commencé par la description de ces systèmes en donnant les définitions de base, suivis d'une classification des comportements hybrides en donnant des exemples pour ces derniers. Nous avons consacré la suite aux différentes approches de modélisation utilisées dans le développement du modèle de ces systèmes. Dans ce volet, nous avons fait un tour d'horizon sur les outils de modélisation de ces systèmes et nous nous somme concentré sur les automates hybrides et les réseaux de Petri hybrides. Ces derniers sont les outils exploités dans l'élaboration du modèle de la sous classe des SDHs considérée dans notre travail.

Les SDHs englobent une large classe de systèmes dynamiques, mais nous avons limité le contenu de ce point aux classes les plus rencontrées dans la littérature et dans la réalité. Ainsi, nous avons ciblé la description de la sous-classe qui nous intéresse des systèmes à commutations contrôlées, à savoir les systèmes à flux continus.

Du fait que la fiabilité de l'estimation de l'état est en relation implicite avec l'analyse de la stabilité, nous avons évoqué ce problème dans le dernier volet de ce chapitre.

Ce chapitre présente une vue sur les SDHs, et bien entendu, cet état de l'art n'est pas complet. Nous avons retenu les AH linéaires et les RdPH élémentaires comme outils de modélisation pour notre travail. Ces derniers seront les modèles sur lesquels nous allons nous baser pour représenter le comportement du système en présence des fautes. Nous introduirons une nouvelle approche de translation du RdPH élémentaires vers l'AH linéaires. Cette représentation, ainsi que l'outil d'AH linéaires seront détaillés dans la suite de cette thèse. Le prochain chapitre présente un bref état de l'art sur la surveillance, les différentes approches ainsi que les différentes méthodes qui existent dans la littérature ; et d'une façon plus détaillée la méthode retenue dans notre étude et son exploitation dans notre contribution.

CHAPITRE 2

SURVEILLANCE DES SYSTÈMES DYNAMIQUES

Résumé : Dans le deuxième chapitre, nous présenterons le concept général de la surveillance des systèmes dynamiques allant de quelques définitions et terminologies de base à la fonction de surveillance ; son principe et ses sous fonctions. Par la suite, deux grandes classes des méthodes de surveillance seront présentées, à savoir les méthodes sans modèles et les méthodes à base de modèles. Nous discuterons les critères de classification de ces méthodes avant de donner quelques approches de surveillance des systèmes dynamiques hybrides. Ce chapitre introductif permet de situer la méthode de surveillance que nous allons suivre dans notre travail.

2.1. Introduction

La complexité croissante des systèmes dynamiques, rend ces derniers vulnérables aux défaillances, celles-ci étant généralement à l'origine de coûts importants en termes de sécurité (risque d'accidents, de pollutions,...), et en termes de disponibilités (diminution de la productivité). Cette vulnérabilité justifie l'introduction de modules de surveillance.

La surveillance a pour premier objectif d'accroître la sécurité de l'installation en émettant, à partir des informations générées par les capteurs, des alarmes [Valette *et al.*, 89]. L'objectif est d'attirer l'attention de l'opérateur de supervision sur l'apparition d'un ou plusieurs événements susceptibles d'affecter le bon fonctionnement de l'installation, comme le dépassement d'un seuil de sécurité au niveau du remplissage d'un réservoir. Cette fonction a principalement été développée pour des systèmes critiques tels que les installations chimiques, les centrales nucléaires, les plates-formes pétrolières ou aéronautique.

Les méthodes de surveillances sont devenues une aide significative, en particulier pour l'exploitation des systèmes dynamiques. Elles contribuent à l'amélioration de la disponibilité, de la qualité et de la sûreté de fonctionnement ainsi qu'à la réduction des coûts des installations pour les systèmes industrielles. La surveillance regroupe l'ensemble des algorithmes de détection et de diagnostic des défaillances.

La surveillance s'intègre dans le cadre plus général de la supervision et permet d'améliorer la qualité et de réduire les coûts, en intervenant au cours des phases du cycle de vie du produit qui peut être décomposé en 3 parties :

La conception : Une méthode d'analyse préventive peut être utilisée dès les premières étapes d'un projet pour déterminer au mieux les défaillances possibles ainsi que leurs effets : "diagnostic de conception",
La production : Les défauts peuvent être identifiés et localisés en cours de production. Le diagnostic permet de corriger ou d'arrêter la fabrication de produits puisqu'ils ne pourront satisfaire le cahier des charges,
L'utilisation : Une procédure d'arrêt et/ou de retrait peut être déclenchée si la sécurité est mise en péril par l'occurrence d'un défaut lors d'une phase d'utilisation. Une localisation précise des défaillances pourra permettre d'améliorer la maintenabilité et la disponibilité en indiquant les composants à remplacer. Ici, l'intérêt du diagnostic est de fournir les informations qui définissent une politique de maintenance appropriée.

Nous pouvons distinguer deux fonctions principales dans la surveillance ; **la détection** et **le diagnostic** qui seront présentées dans le paragraphe 2.3.

2.2. Définitions et terminologie

Il est intéressant, dans un premier temps, de rappeler les principales définitions et terminologie utilisées dans les notions de la surveillance des systèmes dynamiques. Elles

reposent notamment sur les travaux effectués dans les références suivantes : [Isermann et Balle, 97], [Milne, 87], [Isermann et Balle, 00], [Villemeur, 88], [Combacau, 91], [Toguyeni, 92], [Lefebvre, 00], [Zemouri, 03], [Philippot, 06], [Deschamps, 07].

Définition 2.1. Système physique : Un système physique est un ensemble d'éléments (composants, constituants) interconnectés ou en interaction organisés pour réaliser une fonction.

❑

Définition 2.2. Composant : Un composant est une partie du système choisie selon des critères liés à la modélisation. Il doit être simple à modéliser dans le sens où cela doit être naturel : il peut s'agir d'un composant (physique ou logique) complet du système ou d'une partie parfaitement délimitée de ce groupe de composants. Le comportement du composant élémentaire n'est pas décomposable ou sa décomposition n'est pas souhaitée ; il constitue une "brique" du comportement du système.

❑

Définition 2.3. Modèle : Un modèle d'un système physique est une description de sa structure et une représentation comportementale ou fonctionnelle de chacun de ses composants. Le modèle contient toute l'information relative à un système physique, il est utilisable ensuite par la procédure de la surveillance.

❑

Définition 2.4. Le système de surveillance : Dans le cadre de notre travail, nous considérons la surveillance comme un dispositif passif, dans le sens où ce dispositif n'influence pas le comportement du système à diagnostiquer. Un système de surveillance a pour vocation première d'émettre des alarmes dont l'objectif est d'attirer l'attention de l'opérateur de supervision sur l'apparition d'un ou plusieurs événements susceptibles d'affecter le bon fonctionnement du système. Son rôle donc, est de détecter les défaillances en observant l'évolution du système, puis de les diagnostiquer en localisant les éléments défaillants et enfin identifier les causes premières.

❑

Compte tenu de la complexité des systèmes, la génération d'alarmes est le moyen le plus employé pour avertir l'opérateur de l'occurrence d'un événement anormal. Les alarmes sont donc liées aux dysfonctionnements pouvant apparaître sur le système.

Définition 2.5. L'anomalie : Condition anormale diminuant ou supprimant l'aptitude d'une entité fonctionnelle à accomplir une fonction requise. Ce terme générique permet de décrire tout ce qui n'est pas conforme à une référence.

❑

Définition 2.6. Défaut : C'est une déviation du système par rapport à son comportement normal, qui ne l'empêche pas de remplir sa fonction. Un défaut est donc une anomalie qui concerne une ou plusieurs propriétés du système, ou :

♦ Tout écart entre la caractéristique observée sur le dispositif et la caractéristique de référence, lorsque celui-ci est en dehors des spécifications.
♦ N'importe quel état indésirable d'un composant ou d'un système.
♦ Déviation non permise d'au moins une propriété ou un paramètre caractéristique du système des conditions acceptables ou/et standards.

Un défaut peut aboutir à une défaillance et parfois même à une panne.

❑

Définition 2.7. Dégradation : Tout état qui se caractérise par une évolution irréversible des caractéristiques d'un système est une dégradation. La dégradation peut être liée à des facteurs directs, tels que l'usage, le temps..., ou à des facteurs indirects, tels que l'humidité, la température…. La dégradation peut aboutir à une défaillance, quand les performances du système sont en dessous d'un seuil d'arrêt défini par les spécifications fonctionnelles.

❑

Définition 2.8. Défaillance : Une défaillance est une anomalie altérant ou empêchant l'aptitude d'une unité fonctionnelle à accomplir la fonction souhaitée. Une défaillance correspond à un passage d'un état à un autre, par opposition à une panne qui est un état de dysfonctionnement.

Une défaillance implique l'existence d'un défaut, puisqu'elle aboutit à un écart entre la caractéristique mesurée et la caractéristique de référence. Inversement, un défaut ne conduit pas nécessairement à une défaillance. En effet, le système peut très bien conserver son aptitude à assurer une fonction requise, si les défauts qui l'affectent n'ont pas d'impacts significatifs sur la mission. Si une défaillance peut conduire à une cessation de l'exécution de la mission principale du système, ce dernier est déclaré en état de panne. Ainsi, la panne est toujours le résultat d'une défaillance.

❑

Définition 2.9. Panne : C'est un état de dysfonctionnement et conséquence d'une défaillance affectant le système, aboutissant à une interruption permanente de sa capacité à remplir une fonction requise et pouvant provoquer son arrêt complet. C'est la cause de l'apparition de symptômes. Deux types de pannes peuvent être distingués :

♦ Les pannes permanentes : une fois la panne est produite, elle nécessite une action de réparation.

♦ Les pannes intermittentes : le système peut retrouver son fonctionnement nominal après l'occurrence de la panne. Une panne intermittente est généralement le résultat d'une dégradation partielle et progressive d'un composant du système, pouvant aboutir à une panne permanente.

Par abus de langage, on pourra appeler une panne par mode de défaillance.

❑

Définition 2.10. Symptôme, Observation, Mesure : Un symptôme correspond à une ou plusieurs observations qui révèlent d'un dysfonctionnement. Il s'agit d'un effet qui est la conséquence d'un comportement anormal.

Un dysfonctionnement est un état où le système évolue avec des dynamiques pouvant l'amener soit à une défaillance soit à une violation du cahier des charges. Ce dysfonctionnement peut être à l'origine de plusieurs causes: Apparition d'un défaut, une mauvaise décision de la partie commande, une fausse information du capteur.

Une observation est une information obtenue à partir du comportement ou du fonctionnement réel du système.

Une mesure est une observation élémentaire du fait qu'elle reflète une et une seule grandeur physique. Elle est représentée par une variable dont le contenu est l'image d'une grandeur physique. Son obtention s'effectue par l'intermédiaire de capteurs.

❑

La Figure 2.1 représente les anomalies suivant leur criticité.

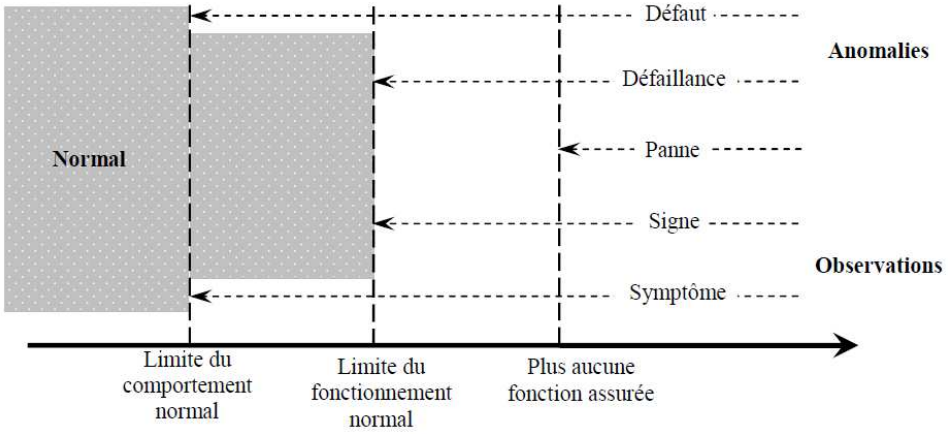


Figure 2.1. Anomalies et Observations classées par criticité croissante.

Définition 2.11. Le résidu : Un résidu est un signal indicateur de défauts. Il reflète la cohérence des données mesurées vis-à-vis du modèle comportemental du système. Autrement dit : Le résidu est l'écart produit par la comparaison entre le comportement réel et le comportement nominal du système.

❑

2. 3. Fonctions de la surveillance :

Le rôle de la surveillance est de recueillir en permanence tous les signaux en provenance du système et de la commande, de reconstituer l'état réel du système commandé et de faire toutes les inférences nécessaires pour produire les données utilisées ou utilisables, en vue de :

- ♦ Dresser des historiques de fonctionnement,
- ♦ Le cas échéant, mettre en œuvre un processus de traitement de défaillances.

Comme l'illustre la figure suivante, la surveillance regroupe les fonctions suivantes : La détection, le diagnostic qui regroupe à son tour deux sous-fonctions ; La localisation et l'identification :

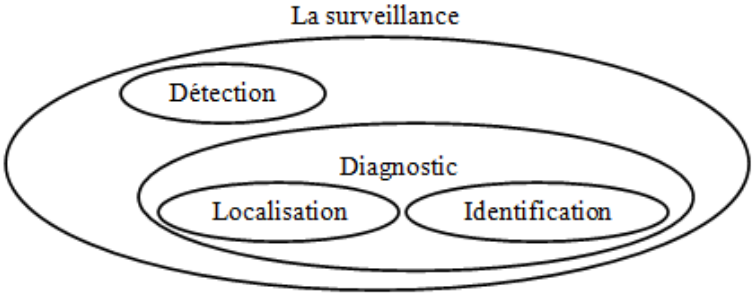


Figure 2.2. Fonctions de la surveillance.

2.3.1. La détection

La détection (Figure 2.3)**,** qui répond à la question "y-a-t-il une (nouvelle) anomalie dans le système ?", permet de déterminer la normalité ou l'anormalité du système en fonctionnement.

Autrement dit : La détection vise à déterminer l'apparition et l'instant d'occurrence d'un défaut.

On peut distinguer deux grandes classes d'anomalies :

♦ La première regroupe les situations pour lesquelles le comportement du système devient anormal par rapport à ses caractéristiques intrinsèques.

♦ La seconde regroupe les situations dans lesquelles le comportement est anormal par rapport à la loi de commande appliquée.

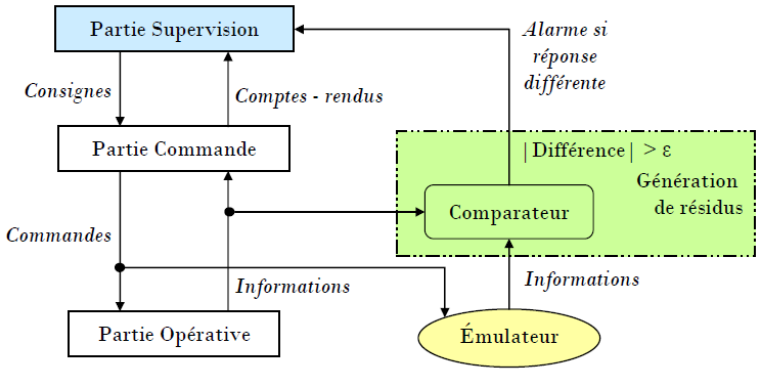


Figure 2.3. Un exemple de détection grâce à un émulateur

La détection consiste à comparer la signature courante à la signature de référence associée aux modes de fonctionnement identifies et ensuite à prendre une décision en fonction du résultat de la comparaison. D'autres techniques de détection s'appuient sur les dates limites d'occurrences des signaux attendus, le suivi de l'évolution de l'état du système, les systèmes experts, l'utilisation de capteur spécialisés et les techniques d'analyse de fréquence.

Dans les procédures de détections, les signatures utilisées sont des grandeurs scalaires, des courbes ou des images. Sachant que le signal d'écart possède un comportement aléatoire, la prise de décision nécessite la définition de seuils aux maxima et aux minima au-delà desquels on déclarera un dysfonctionnement, c.-à-d. de caractériser le fonctionnement du système de normal ou d'anormal.

La détection de symptômes d'anomalies liés aux éléments du procédé requiert généralement l'élaboration d'un modèle à surveiller. Ce modèle peut être de bon fonctionnement ou un modèle de dysfonctionnement. Par exemple, dans le cas des systèmes discrets, un modèle correspondrait à un RdP et dans le cas de système continu un modèle correspondrait à un ensemble d'équations différentielles. Sans modèle du système à surveiller, la stratégie adoptée consiste en l'exploitation des informations données par les capteurs et les détecteurs au niveau local du système.

Un test de détection (dit aussi test de cohérence ou test de consistance) a pour finalité de vérifier si un ensemble d'informations représentatives de l'état d'un système physique est cohérent avec la connaissance d'un comportement donné qui peut être normal ou anormal comme le montre la Figure 2.4. Le résultat de la comparaison produit un écart, appelé résidu. Cet écart sera comparé à des seuils fixés à priori. Si le seuil de la détection est trop petit, il peut y avoir des fausses alarmes. Si le seuil est trop grand, on aboutit à des manques à la détection. Les informations sont associées à des variables ; elles peuvent être des observations qualitatives ou des mesures.

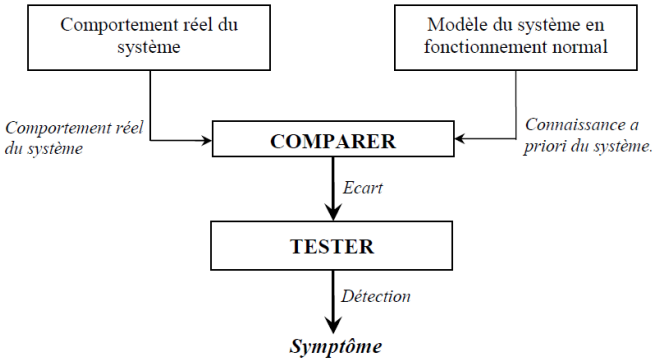


Figure 2.4. Test de cohérence (test de détection, test de consistance).

Une défaillance sera détectable si au moins un résidu permet de la détecter. Les résidus sont obtenus en comparant des modèles s'exprimant sous la forme d'état et un système réel. Lorsque le modèle permet de représenter exactement le système (aucune erreur de modélisation, connaissance de la nature des signaux inconnus agissant sur le système), alors les résidus générés seront strictement égaux à zéro en fonctionnement normal et différent de zéro en présence de défaillances. La procédure de détection se résumera alors à déclencher une alarme lorsqu'au moins un résidu différera de zéro. Ceci est bien sûr un point de vue théorique car un modèle est toujours une approximation, d'où l'importance du seuil de détection.

Une procédure de décision doit être implantée afin de décider si la valeur différente de zéro du résidu doit générer une alarme ou non. La qualité de la détection dépend bien entendu de la procédure de décision choisie mais aussi et surtout de la qualité des résidus utilisés. Afin de réduire les taux de fausse alarme et de non détection, les résidus doivent être robustes, c'est-à-dire rendus le plus possible sensible aux défaillances et le moins possible aux perturbations ou erreurs de modélisation.

Dans certains travaux [Combacau *et al.*, 00], [Boufaied, 03], cette fonction est considérée comme un élément distinct de la fonction de diagnostic et plutôt une entité de la surveillance. D'autres travaux [Chow et Wilsky, 84], [Isermann, 84] considèrent cette fonction comme une information primordiale et indissociable du diagnostic.

2.3.2. Le diagnostic

L'objectif du diagnostic consiste à déterminer à chaque instant le mode de fonctionnement du système par ses manifestations extérieures. Il s'appuie sur une connaissance a priori des modes

de fonctionnement et sur une connaissance instantanée matérialisée par une nouvelle observation de l'état du système. Son principe général consiste à confronter les données relevées au cours du fonctionnement réel du système avec la connaissance que l'on a de son fonctionnement normal ou défaillant qui a entraîné la dégradation du système [Combacau *et al.*, 00]. Si le mode de fonctionnement identifié est un mode défaillant, le système de diagnostic pourra localiser sa cause. La fonction du diagnostic est donc de chercher une causalité liant le symptôme, la défaillance et son origine.

On peut dire aussi, que le diagnostic consiste à localiser les éléments défaillants et à identifier les causes à l'origine du problème, ceci en établissant un lien causal entre les symptômes et les éléments fautifs à remplacer. La phase qui suit correspond à la décision. Elle a pour rôle de déterminer et d'engager les actions permettant de ramener au mieux le système dans un état normal. Ces actions peuvent être des ordres d'arrêts d'urgence ou des lancements de réparations ou d'opérations préventives. Dans le cas où on voudrait éviter une perte de production, cette décision peut être une reconfiguration du procédé. Le cadre de notre travail du diagnostic sera présenté en Chapitre 5.

La fonction de diagnostic fait apparaître les deux sous-fonctions : la **localisation** et **l'identification** [Isermann et Balle, 97].

2.3.2.1. La localisation

Cette sous fonction a pour but de répondre à la question " à quelles classes de défauts ou de défaillances appartiennent les anomalies du système ? ". La localisation consiste à déterminer l'endroit du procédé où s'est produite la défaillance et la nature de celle-ci. Lorsqu'une défaillance est détectée, une procédure de localisation est utilisée pour permettre de déterminer l'origine de celle-ci. A la différence de la détection où un seul résidu est nécessaire, la procédure de localisation nécessite un ensemble (ou vecteur) de résidus. Pour permettre la localisation, le vecteur de résidus doit avoir un certain nombre de propriétés permettant de caractériser de manière unique chaque défaut. En effet, il est possible de déterminer une défaillance, ou une panne, résultant d'un défaut. Par contre, le problème inverse est plus difficile à résoudre, puisque une panne peut résulter d'un ou plusieurs défauts, comme il est montré dans la Figure 2.5, inspirée de [Zemouri, 03].

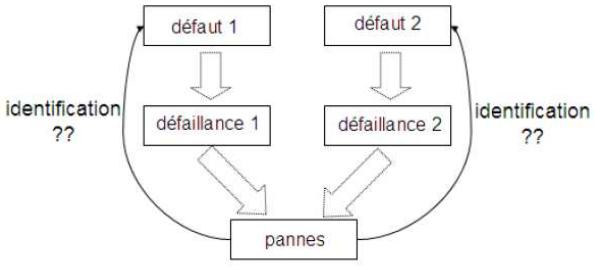


Figure 2.5. La difficulté de localiser des défauts

2.3.2.2. L'identification

L'identification, qui a pour rôle de déterminer quelles sont les caractéristiques de chacun des défauts ou des défaillances. Elle consiste à déterminer (identifier) les caractéristiques précises de la défaillance. L'identification (ou estimation) du défaut est une tâche plus délicate qui nécessite d'utiliser un modèle de comportement du système en présence des défaillances avec un niveau élevé de connaissance sur les défaillances (c'est à dire une connaissance de la structure et de la dynamique de la défaillance). L'obtention d'une estimation de la défaillance permet bien entendu de donner une image beaucoup plus précise de l'état du système.

2.3.3. La Supervision

C'est une macro-fonction regroupant des tâches de commande et de surveillance. La supervision doit piloter l'exécution de la séquence d'opération et assurer la gestion et la commande en temps réel des ressources nécessaires à cette exécution, et ceci, quel que soit le fonctionnement du système normal ou avec présence de défaillances :

♦ En fonctionnement normal, elle doit surveiller et contrôler le déroulement des opérations.

♦ En présence d'une défaillance, la supervision doit prendre les décisions nécessaires pour assurer un retour vers le comportement normal [Combacau *et al*., 00].

Malgré une automatisation accrue, l'opérateur se positionne encore comme le maillon 'intelligent' de la boucle de surveillance : il a en charge d'analyser la situation et de prendre la décision adéquate. De plus, suite à une défaillance ou une dérive, les moyens de réaction sont souvent manuels ou semi-automatiques. La tâche d'analyse de l'opérateur est facilitée par la mise en place d'interfaces ergonomiques donnant différentes vues du système sous forme de synoptiques. Ceux-ci reproduisent l'installation et affichent en temps réel les grandeurs mesurées. Citons comme exemples, les interfaces industriels [Mokhtari, 07] tel que : Vivale, Areal, Ordinal Technologies, Meta, Actors solutions…etc.

2.3.4. La correction

La **reprise** consiste à trouver le remède de la panne. Dans le cas le plus simple, c'est le remplacement de l'élément défaillant.

La **maintenance** [Mokhtari, 07] est la fonction qui permet le remplacement ou la réparation des équipements usagés ou défaillants. Nous distinguons deux types de maintenance :

1- *La maintenance préventive* imposée par la sûreté de fonctionnement. Elle peut être :

- ♦ Soit *systématique*, c'est-à-dire effectuée selon un échéancier établi suivant le temps d'usage ou à partir du nombre d'unités d'usage ;
- ♦ Soit *conditionnelle* : elle intervient, lors de la prédiction d'une future défaillance du système physique, c'est-à-dire lors d'une dégradation des performances du système.

2- *La maintenance corrective*, effectuée après défaillance. Elle intervient au cours de la fonction reprise. Elle est :

- ♦ Soit *curative* : Elle consiste en la remise en l'état initial, ce qui peut correspondre au remplacement du composant défectueux.

♦ Soit *palliative* : Dans ce cas, elle correspond à une solution de secours provisoire permettant au composant défaillant d'assurer au moins une partie de ses fonctionnalités. Elle doit, toutefois, être suivie d'une action curative dans les plus brefs délais.

2.4. Les méthodes de surveillance

La première question à se poser quant au choix d'une méthode de surveillance est la suivante : que savons-nous sur le système où apparaissent des défaillances ? Et plus exactement, possédons-nous un modèle permettant de connaître l'évolution de ce système ?

En fonction de la réponse, nous pourrons nous diriger vers l'une des deux familles de surveillance [Zwingelstein, 95] : les méthodes sans modèles (model-free methods) ou les méthodes avec modèles ou à base de modèles (model-based methods). Ces dernières sont différenciés par plusieurs critères : l'évolution de la dynamique du système (continu, discret ou hybride), la mise en place du système de surveillance (en ligne ou hors ligne), la nature de l'information (quantitative ou qualitative) et sa distribution (décentralisée, centralisée) [Derbel, 09]. Ces méthodes sont illustrées à la Figure 2.6 :

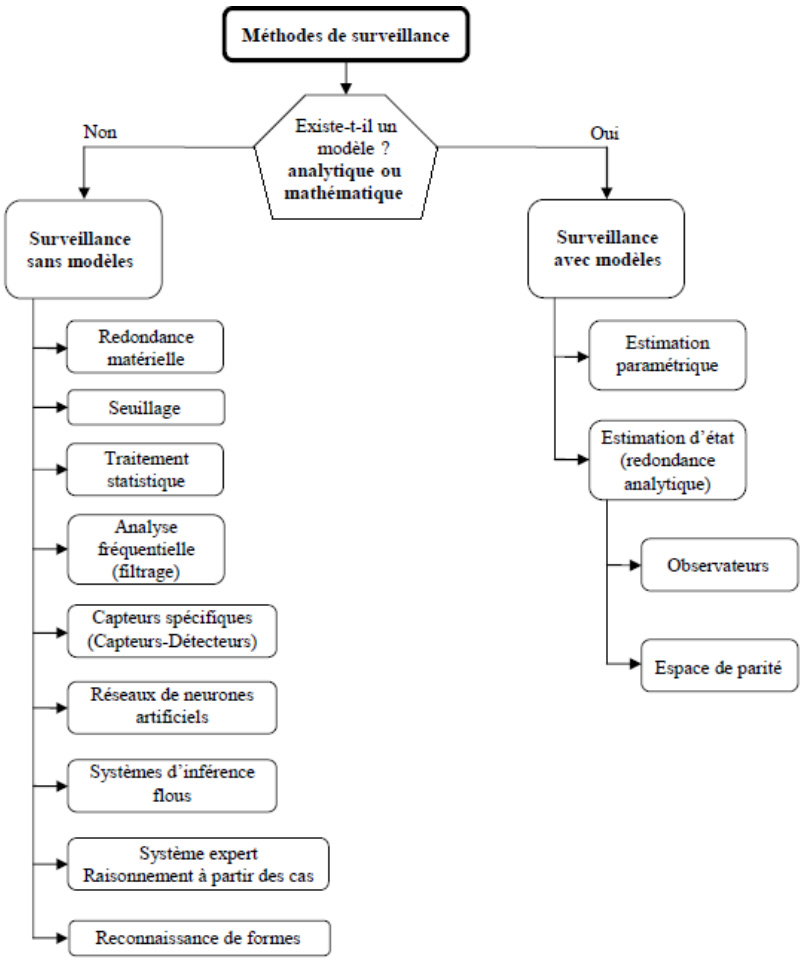


Figure 2.6. Différentes méthodes de surveillance.

Le principe général de ces méthodes est de confronter les données relevées au cours du fonctionnement réel avec la connaissance que l'on a du fonctionnement nominal (détection) ou des fonctionnements défaillants (diagnostic : localisation et identification).

2.4.1. Les méthodes de surveillance sans modèles

Il y a plusieurs systèmes dynamiques qui ne peuvent être modélisés. Différentes causes sont à l'origine de cette réalité, on cite par exemple la complexité du système ou bien la reconfiguration à plusieurs reprises en cours de fonctionnement. On ne peut appliquer sur ces systèmes que les méthodes de surveillance sans modèles. Ces méthodes nécessitent l'historique du procédé et s'appuient sur des règles heuristiques ou sur des exemples d'exécution. Parmi ces méthodes, on trouve :

A. La méthode de redondance matérielle

Cette méthode consiste à multiplier physiquement les capteurs critiques d'une installation. Un traitement des signaux, issus des éléments redondants, effectue des comparaisons et distingue l'élément défectueux en cas d'incohérence. Cette méthode est pénalisante en termes de poids, de puissance consommée, de volume, de coût (d'achat et de maintenance). Cette technique ne s'applique généralement que sur des capteurs.

B. La méthode de seuillage

Cette méthode s'appuie sur la comparaison entre les signaux fournis par le capteur avec des valeurs limites constantes ou adaptatives [Montmain, 92]. Le franchissement de ces valeurs seuils indique la présence d'une anomalie. La gravité de l'anomalie peut être déterminée par la mise en place de deux seuils de détection, le franchissement du premier correspond à la présence probable d'un défaut et le second caractérise sa gravité. Le seuillage comporte un inconvénient de son aspect catégorique.

C. Les méthodes statistiques

Ces méthodes supposent que les informations transmises par les capteurs possèdent certaines propriétés statistiques, sur lesquelles des tests de seuil sont effectués [Basseville, 98] [Zemouri, 03]. L'étude de l'évolution de la moyenne ou de la variance d'un signal peut favoriser la mise en évidence d'une anomalie. La prise de décision est généralement effectuée à l'aide d'un test d'hypothèses où deux hypothèses (par exemple le maximum de vraisemblance généralisée [Willsky, 76]) représentent le fonctionnement normal et anormal.

D. Méthodes d'analyse fréquentielle (Filtrage) :

Cette approche du traitement du signal repose sur l'analyse fréquentielle (transformée de Fourier). Elle est bien évidemment très utilisée pour la détection de phénomènes périodiques comme en analyse vibratoire. L'analyse du spectre des signaux issus des capteurs permet de

déterminer très efficacement l'état du système sous surveillance. Les signaux sont ici, tout d'abord, analysés en état normal de fonctionnement. Ensuite, toute déviation des caractéristiques fréquentielles d'un signal est reliée à une situation de défaillance. De plus, un échantillonnage fréquent est nécessaire [Lesecq *et al.*, 01] pour permettre de reconstituer le signal de départ tout en minimisant la perte de fréquence.

E. Capteurs spécifiques (Capteurs-Détecteurs)

Des capteurs spécifiques peuvent également être utilisés pour générer directement des signaux de détection ou connaître l'état d'un composant [Touaf, 05]. Par exemple, les capteurs de fin de course, d'état de fonctionnement d'un moteur ou de dépassement de seuils sont largement employés dans les installations industrielles.

F. Réseaux de neurones artificiels :

Quand la connaissance sur le système à surveiller n'est pas suffisante et que le développement d'un modèle de connaissance du système est impossible, l'utilisation de modèle dit "boîte noire" peut être envisagée. Pour cela des réseaux de neurones artificiels (RNA) ont été utilisés. Un RNA est en fait un système informatique constitué d'un nombre de processeurs élémentaires (ou nœuds) interconnectés entre eux qui traite -de façon dynamique- l'information qui lui arrive à partir des signaux extérieurs. Les RNA peuvent être utilisés pour le diagnostic des défaillances [Zemouri, 03].

G. Systèmes d'inférence flous :

Pendant les vingt dernières années, les systèmes d'inférence floue (SIF) – dont les bases relèvent de la théorie des ensembles flous de Zadeh [Zadeh, 65]– sont devenus très populaires. Les applications dans le traitement du signal, la modélisation, la surveillance, la supervision de systèmes et la prise de décision sont en effet autant d'applications qui démontrent la capacité des SIF à traiter des problèmes non linéaires grâce à l'utilisation de connaissances expertes.

H. Les systèmes experts (le raisonnement à partir des cas)

Le raisonnement à partir de cas (Case Based Reasoning) modélise l'expertise et les capacités de raisonnement de spécialistes qualifiés dans le domaine de pointe [Zwingelstein, 95], [Farreny, 89]. Ce raisonnement est qualifié de résoudre un problème en s'appuyant sur des expériences passées. La connaissance est emmagasinée sous forme de cas. Un cas est un morceau contextualisé de connaissances, représentant une expérience, qui peut être utilisé pour réaliser les buts du moteur de raisonnement. Le raisonnement à partir de cas est un raisonnement par analogie. Les attributs d'une situation sont employés en tant qu'index dans la bibliothèque de cas pour récupérer le meilleur, selon certains critères de similarité, et ainsi pour déterminer la solution au problème.

I. La reconnaissance des formes

Les méthodes de reconnaissance des formes (RdF) utilisent des algorithmes de classification des formes et des mesures [Dubuisson, 90], [Ondel, 06] en les comparants à des formes types. Ses applications interviennent dans de nombreux domaines tels que la reconnaissance vocale, la reconnaissance de caractères, l'automatisation industrielle, le diagnostic médical et la classification de documents.

2.4.2. Les méthodes de surveillance à base de modèles

Les méthodes de surveillances à base de modèles ont été introduites au début des années 70. Plusieurs travaux ont été réalisés sur ces méthodes [Frank, 90], [Patton and Chen, 91], [Patton, 94], [Frank, 96], [Dubuisson, 01] et [Hamscher *et al*., 92]. Depuis, de nombreux articles font régulièrement le point sur l'avancée des différentes approches. Les méthodes de surveillance utilisant un modèle reposent sur la génération et l'étude du résidu.

Classiquement, en Automatique, des modèles dits de bon fonctionnement sont utilisés. Ils caractérisent le comportement normal du système, c'est-à-dire lorsqu'aucune défaillance n'est présente. En surveillance, par contre, il est parfois nécessaire de compléter le modèle afin de caractériser le comportement défaillant du système. Trois niveaux de connaissance peuvent être considérés [Cocquempot, 04]:

- ***Le niveau 1*** est le niveau de connaissance le plus élémentaire. Il consiste à indiquer les équations décrivant le composant qui sont influencées directement par la défaillance, c'est à dire. Les équations du modèle (contraintes) qui ne sont probablement plus valides en cas de défaillances.
- ***Le niveau 2*** de connaissance est plus précis car il consiste à décrire, grâce à des variables supplémentaires (variables de défaillance), comment sont modifiées les équations de fonctionnement normal lorsqu'une défaillance survient. Les défaillances peuvent être additives ou multiplicatives suivant la manière dont les variables de défaillance influencent les équations du modèle.
- ***Le niveau 3*** de connaissance consiste à modéliser l'évolution dynamique de la défaillance.

Des équations supplémentaires liant les variables de défaillance sont ajoutées au modèle de bon fonctionnement. Pour obtenir ce modèle, soit une connaissance fine des phénomènes physiques est nécessaire, soit des données expérimentales du processus défectueux doivent pouvoir être utilisées.

Etant donné l'importance de méthodes de surveillances à base de modèles dans nos travaux la sous-section suivante leur est consacrée.

2.4.2.1. Estimation paramétrique

Les méthodes d'estimation paramétrique supposent l'existence d'un modèle paramétrique décrivant le comportement du système et la connaissance des valeurs des paramètres en fonctionnement nominal [Iserman, 84] [Willsky, 76]. Elles consistent alors à identifier les paramètres caractérisant le fonctionnement réel, à partir de mesures des entrées et des sorties du système.

On dispose ainsi d'une estimation des paramètres du modèle, réalisée à partir des mesures prises sur le système, et de leurs valeurs théoriques. Pour détecter l'apparition de défaillances dans le système, il faut effectuer la comparaison entre les paramètres estimés et les paramètres théoriques. Comme pour les méthodes de redondance analytique, la théorie de la décision sert alors à déterminer si l'écart observé est dû à des aléas normaux du fonctionnement ou à des défaillances. La différence entre les méthodes de redondance analytique et les méthodes d'estimation paramétrique est qu'on effectue, pour les premières, la comparaison entre l'état estimé et l'état théorique du système, alors que pour les secondes, on compare les paramètres estimés aux paramètres théoriques du système.

Le principe consiste à estimer en continu des paramètres du système en utilisant les mesures d'entrée/sortie et à évaluer la distance qui les sépare des valeurs de référence de l'état normal du système (Figure 2.7). L'estimation paramétrique possède l'avantage d'apporter de l'information sur la taille des déviations. Toutefois, un des inconvénients majeurs de la méthode réside dans la nécessité d'avoir un système physique excité en permanence. Ceci pose des problèmes pratiques dans le cas de systèmes dangereux ou fonctionnant en mode stationnaire. De plus, les relations entre les paramètres mathématiques et physiques ne sont pas toujours inversibles de façon unitaire, ce qui complique la tâche du diagnostic basé sur les résidus.

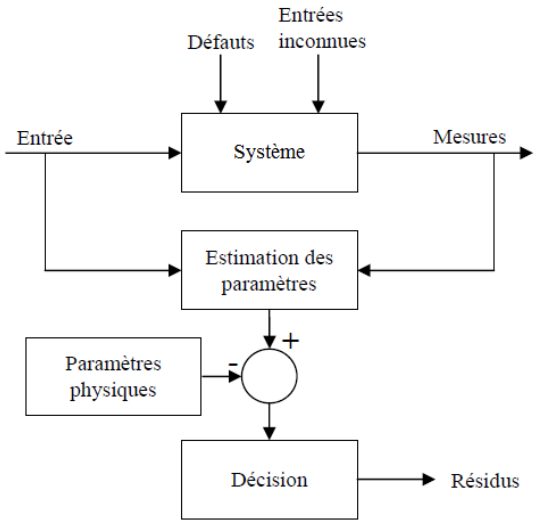


Figure 2.7 Estimation paramétrique pour la détection et le diagnostic des défauts.

La seconde catégorie de méthodes à base de modèles regroupe celles reposant sur l'estimation d'état. On y retrouve l'observateur et l'espace de parité.

2.4.2.2. Estimation d'état (redondance analytique)

Cette redondance[1] consiste à utiliser des informations supplémentaires issues, non plus de capteurs, mais de modèles permettant l'élaboration de grandeurs de même nature que celles issues des capteurs. L'intérêt est de permettre de remplacer un capteur physique par un capteur informationnel (résidu).

A. Observateurs

La méthode à base d'observateurs ou filtre est la plus couramment utilisée. Les premiers travaux datent des années 70 ([Willsky, 76] par exemple). Beard et Jones ont été en fait les premiers à proposer le remplacement de la redondance matérielle par des algorithmes de détection basés sur des observateurs.

Les observateurs ou filtres sont des outils bien connus des automaticiens à des fins de commande en boucle fermée reposant sur l'estimation d'état. Le principe général est de concevoir un système dynamique permettant de donner une image, ou estimation, de certaines variables, ou combinaisons de variables, nécessaires au bouclage. Lorsque le système est dynamique et que certaines variables (conditions initiales) sont inconnues, l'estimation n'est correcte qu'après un certain temps de convergence, fixé par la dynamique de l'observateur. Le principe général consiste à comparer des fonctions de sorties estimées avec les mêmes fonctions des sorties mesurées. L'écart entre ces fonctions est utilisé comme résidu.

Un observateur d'ordre réduit revient à ne considérer qu'une partie du système, donc à estimer une partie de l'état et à rejeter l'autre. Par ailleurs, l'élimination d'une partie du système peut être utilisée pour rejeter les perturbations. Les observateurs à entrées inconnues sont basés sur ce principe.

B. Espace de parité (approche par Relation de Redondance Analytique RRA).

L'approche à base de Relations de Redondance Analytique ou approche de l'espace de parité, a été une des premières méthodes employées [Gertler, 97]. Son nom provient du domaine de l'informatique où le contrôle de parité se faisait dans les circuits logiques. Son principe est de transformer, réécrire les équations du modèle de manière à obtenir des relations particulières appelées RRA: Relations de Redondance Analytique.

Ces relations ont pour propriété de ne lier que des grandeurs connues, disponibles en ligne. (C'est-à-dire la vérification de la consistance existant entre les entrées et les sorties du système à surveiller). Les résidus sont obtenus en substituant dans ces RRA les variables connues par leurs valeurs réelles, prélevées sur le système en fonctionnement. L'obtention hors-ligne des RRA est un problème général d'élimination de variables dans un système d'équations algébro différentielles. Lorsque le modèle est linéaire, l'élimination peut se faire par projection dans un

[1] Une définition du mot «redondance» trouvée dans Larousse de Bibliorom est : «Duplication d'équipements chargés d'assurer une fonction donnée, afin que l'un d'eux puisse se substituer à l'autre en cas de défaillance».

sous espace appelé ***espace de parité*** [Chow, 84]. Dans le cas non linéaire, des techniques d'élimination formelle peuvent être utilisées.

2.5. Surveillance des systèmes dynamiques hybrides

Le comportement dynamique d'un système hybride peut être représenté par une succession de modes. Chaque mode i ($i \in M; M = \{1, 2, ..., m\}$, où m est le nombre de modes) correspond à une configuration physique possible et caractérisé par une modalité de l'état discret, un ensemble de contraintes égalités (équations d'état par exemple) et la définition d'un domaine d'admissibilité (décrit par des contraintes inégalités). Les travaux concernant la détection, la localisation ou le diagnostic des défaillances sont peu nombreux [Narasimhan *et al.*, 00]. De plus, même en fonctionnement normal, le mode dans lequel se trouve le système (le mode courant) est à tout instant connu. La détermination du mode courant est donc une fonctionnalité supplémentaire que doit présenter la partie logicielle de surveillance.

Dans [Cocquempot *et al.*, 04] les auteurs proposaient d'utiliser les résidus de parité pour identifier le mode courant du système et estimer l'instant de transition. Les défaillances survenant sur un système hybride peuvent affecter le comportement du système au sein d'un mode ou affecter la séquence d'états discrets.

2.5.1. Approches de surveillance des systèmes dynamiques

Il existe trois approches différentes dans la littérature pour l'implémentation du système de surveillance [Rayhane, 04].

A. L'approche filtre

Le concept de cette approche est d'insérer un ou plusieurs filtres entre l'unité de commande et l'unité opérative du système comme l'illustre la Figure 2.8. Les informations instantanées émises par le capteur déterminent l'état réel du système. Le filtre est composé d'un filtre de commande et d'un filtre de valeurs capteur. Le rôle du premier est de tester la cohérence de l'instruction par rapport à l'état du système, le rôle du deuxième consiste à comparer les signaux transmis par les capteurs avec ceux correspondant au comportement normal du système [Nourelfath, 97].

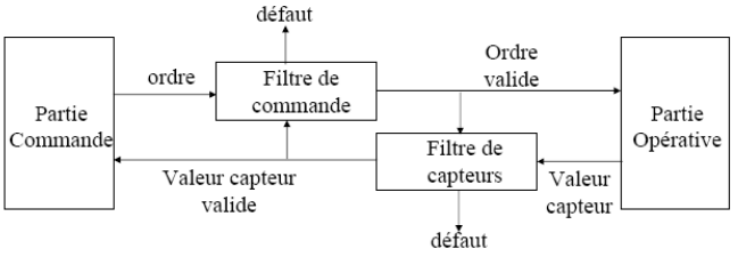


Figure 2.8. Approche filtre.

B. Approche comparateur

Cette approche repose sur la comparaison permanente de l'état réel du système déterminé à partir des informations des capteurs et de celui donné par le modèle de comportement du système (Figure 2.9). Tout écart entre l'état réel du système et celui donné par le modèle signale une défaillance. Le modèle du système est placé en tant que émulateur des évolutions normales de l'unité opérative. Son rôle est de calculer les fenêtres temporelles d'occurrence des comptes rendus émis par le système quand celui-ci est soumis à une commande particulière. Pour une consigne donnée, un bloc de comparaison permet de vérifier si un compte rendu émis par le procédé arrive bien à la date prévue par le modèle.

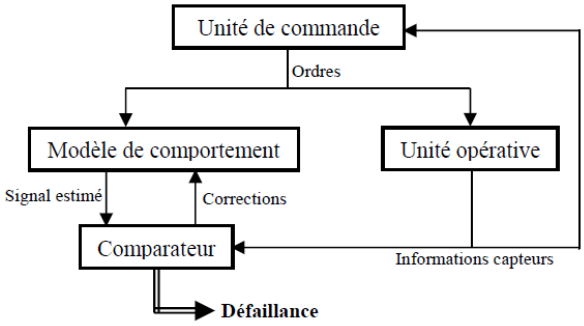


Figure 2.9. Approche comparateur.

C. Approche de modèle de référence

Cette approche nécessite l'existence d'un modèle de référence contenant tous les modèles de comportement normaux du système comme l'illustre la Figure 2.10. Avant l'envoi d'une instruction par l'unité de commande, celle-ci consulte le modèle de référence et s'il y n'a pas de concordance entre l'état du système et la nouvelle instruction alors une erreur de l'unité de commande est détectée. Le modèle de référence et l'unité opérative doivent évoluer simultanément, s'il y a un décalage entre les deux modèles il y aura toujours une défaillance. Cette approche à l'avantage de vérifier l'état du système avant l'exécution d'une nouvelle instruction.

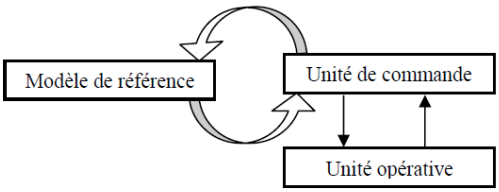


Figure 2.10. Approche par modèle de référence.

2.5.2. Choix de la méthode de surveillance et modèle de bon fonctionnement

Actuellement, on s'oriente vers les méthodes de surveillance à base de modèle comportementale, la plupart des travaux de surveillance à base de modèle qui ont été réalisés, se basent sur des modèles d'abstraction du système hybride afin de ramener la surveillance des SDH à un problème de surveillance soit de systèmes purement continus, soit de systèmes

purement discrets [Lunze, 00], [Koutsoukos *et al.*, 02], [Toguyéni *et al.*, 06]. La surveillance à base de modèle mixte a été également considérée dans les travaux de [Bergman et Larsson, 98], [Biswas *et al.*, 03] et [Domlan *et al.*, 04]. Dans les prochains chapitres, nous présenterons au début le modèle du système en comportement normal (sans présence des défauts), et puis le modèle d'un défaut unique et enfin le modèle global lorsqu'on souhaite localiser un défaut.

Nous avons choisi d'utiliser le formalisme des RdP hybrides et les automates hybrides pour représenter le comportement de la sous-classe des SDH considéré. Le modèle global sera un modèle de RdPH élémentaire translaté en utilisant notre technique de translation vers un automate hybride linéaire qui combine les avantages des deux outils. Certains avantages et inconvénients de ces outils dans le contexte du diagnostic des défaillances, l'approche de modélisation et la technique de translation seront présentés dans le reste du rapport.

2.5.3. Description et caractérisation des défaillances

Un système est dit défaillant lorsque son comportement réel ne correspond pas au modèle de bon fonctionnement. Plusieurs sortes de défaillances peuvent se produire sur un système hybride. En effet, les défaillances peuvent affecter soit l'évolution de l'état continu dans un mode, soit l'évolution discrète c'est à dire la séquence d'états discrets.

A. Défaillances affectant le comportement du système dans un mode

Un mode est entièrement défini par :

- ♦ Un ensemble de contraintes égalité (équations différentielles ou algébriques).
- ♦ Un domaine défini par un ensemble de contraintes inégalité.
- ♦ Une modalité de l'état discret, c'est-à-dire une configuration physique du système.

Une défaillance se produisant dans un mode peut affecter une de ces trois entités.

B. Défaillances affectant l'évolution discrète

L'évolution discrète du système correspond à un chemin (ou trajectoire) dans l'automate hybride, c'est-à-dire à une succession des sommets dans un ordre déterminé lorsque le système est en bon fonctionnement. Toute évolution dans l'automate hybride non conforme au comportement normal est considérée comme une défaillance. Trois types de défaillances peuvent être considérés :

Transition vers un mode non successeur : Si le système fonctionne correctement, seul un sous-ensemble $\Sigma(i)$ de modes (appelés successeurs) sont accessibles à partir d'un mode i. Une transition du mode i vers un mode n'appartenant pas à $\Sigma(i)$ est donc une défaillance.

Non transition : Ce type de défaillance se produit lorsque le système reste dans le mode courant alors que la condition de transition est vérifiée et que le système devrait normalement changer de mode.

Transition anormale vers un mode successeur : Ce type de défaillance se produit lorsque le système passe d'un mode i vers un successeur potentiel j alors que la condition normale de transition n'est pas vérifiée.

Ces trois types de défaillances peuvent être détectés en comparant l'évolution de l'état discret du système en fonctionnement avec l'évolution prévue si le système se comporte normalement. Ceci revient à comparer les trajectoires réelles et théoriques dans l'automate hybride.

2.6. Les modes de fonctionnement des systèmes dynamiques

La nature du mode de fonctionnement du système dynamique implique que la tâche qui lui est attribuée peut être exécutée totalement, partiellement ou non exécutée. Il existe plusieurs modes de fonctionnement [Rayhane, 04] qui sont :

1- Les modes de fonctionnement normaux : Ils comprennent tous les modes pouvant amener le système à exécuter sa tâche y compris le mode nominal qui permet d'exécuter parfaitement la tâche.
2- Les modes de fonctionnement anormaux : Dans ces modes le système ne peut exécuter sa tâche complètement ou même ne pas l'exécuter totalement. On peut les décomposer en :

- ♦ Modes critiques : dans ces modes le système fonctionne d'une façon particulière et souvent non souhaitée.
- ♦ Modes dégradés : dans ces modes le système réalise partiellement ses objectifs.
- ♦ Modes défaillants (de défaillance) : c'est un mauvais fonctionnement du système.
- ♦ Modes interdits : le système ne doit pas fonctionner dans ces modes pour des raisons de sécurité.

Rayhane a introduit par ses travaux la notion de comportement dégradé. Le système est en fonctionnement normal si le temps d'exécution de la tâche est dans un intervalle noté I_m, il est en mode dégradé si la durée de l'exécution de la tâche dépasse l'intervalle I_m mais reste contenue dans l'intervalle J_m. Trois valeurs de temps sont définis pour chaque tâche; $T^m_{\min}$, $T^m_{\max}$ et T^m_c. Pour une exécution normale de la tâche sa durée d'exécution est comprise dans l'intervalle de fonctionnement normal $I_m = \left[T^m_{\min} , T^m_{\max} \right]$.

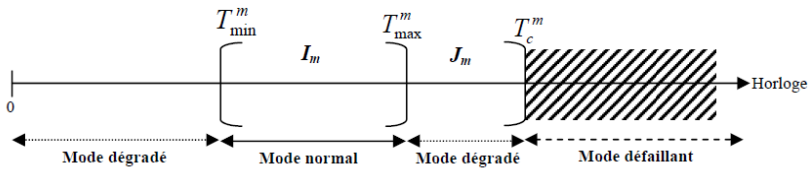


Figure 2.11. Durée d'exécution et modes de fonctionnement

L'intervalle $J_m = \left[T^m_{\max} , T^m_c \right]$ correspond à un intervalle de tolérance dans lequel le système est en mode dégradé. Cet intervalle comptabilise les retards qui peuvent être dus au vieillissement des machines, aux surcharges, aux problèmes mécaniques, …etc. Si la durée d'exécution dépasse la limite T^m_c, le système est considéré comme défaillant (Figure 2.11).

2.7. Conclusion

Dans ce chapitre nous avons présenté le concept général de la surveillance des systèmes dynamiques, ainsi plusieurs définitions nécessaires ont été citées. La surveillance regroupe deux fonctions principales : la détection et le diagnostic qui regroupe à son tour deux sous fonctions élémentaires telles que la localisation et l'identification.

Pratiquement, la nature des informations sur les systèmes dynamiques est différente d'un système à un autre, (des connaissances d'expertise, des modèles qualitatifs ou structurels). Cela influe directement sur la manière de procéder la surveillance. Par conséquent, deux grandes classes des méthodes de surveillance ont été citées, à savoir les méthodes sans modèle, et les méthodes avec modèle. Ces méthodes ont été classées selon plusieurs critères : l'évolution de la dynamique du système, la mise en place du système de surveillance, la nature de l'information et sa distribution.

A notre connaissance, il existe peu de contributions de la surveillance et du diagnostic des systèmes dynamiques hybrides. L'introduction du temps et des évènements non observables nous permet de rendre plus efficace les systèmes de surveillance.

CHAPITRE 3

MODÉLISATION DES SYSTÈMES DYNAMIQUES HYBRIDES À FLUX CONTINUS TOLÉRANTS AUX DÉFAUTS

Résumé : Ce chapitre est consacré aux modèles de base de la technique proposée pour la modélisation des systèmes dynamiques hybrides (SDHs) à flux continu tolérant aux défauts. La notion de la tolérance aux défauts et la modélisation des défauts possibles seront établies pour ces systèmes dans leurs différents types. Nous présenterons ensuite les deux modèles utilisés pour la représentation du système en comportement normal et en présence des défauts ; ce sont les réseaux de Pétri hybrides (RdPH) et les automates hybrides (AH). Ce chapitre constitue la principale contribution de ce travail de thèse.

3.1. Introduction

En raison d'une modernisation incessante des outils de production, les SDH deviennent de plus en plus complexes et sophistiqués. En parallèle, l'Automatique a permis à l'Homme de développer des méthodes de supervision telles la surveillance et le diagnostic et la commande tolérante aux défauts des systèmes. Ces deux notions définies dans le chapitre précédent, nécessitent un travail préliminaire, celui de la modélisation du système, c'est-à-dire une représentation mathématique du comportement dynamique du système.

Cependant, la modélisation des SDH requiert non seulement une connaissance précise des phénomènes intervenant sur ce système, mais également une aptitude à les représenter ; le système et les défauts qui l'affectent. Le problème principal réside l'utilisation d'un modèle simple mais suffisamment précis pour décrire fidèlement le comportement du système en présence des défauts.

Ainsi, sur la base de cette modélisation, un des objectifs de notre travail de thèse est d'élaborer une méthode de modélisation structurelle des SDH à flux continu tolérants aux défauts permettant la représentation puissante et détection des défauts dans ce type des systèmes. Il est nécessaire de synthétiser une méthode de détection des défauts robuste à l'environnement de ces systèmes afin d'éviter les fausses alarmes et les non détections des défauts. Dans certains cas, la détection et la localisation d'un ou de plusieurs défauts est nécessaire mais n'est pas suffisante pour connaitre la partie et/ou l'élément responsable du défaut car il est indispensable de réaliser l'identification pour éviter le même problème au futur.

Nous allons présenter dans ce chapitre les notions utilisées dans notre contribution de thèse, à savoir l'approche de modélisation des systèmes dynamiques hybrides à flux continu tolérant aux défauts.

3.2. La tolérance aux défauts

La tolérance aux défauts est la capacité d'un système en présence des défauts de préserver la capacité d'accomplir les missions souhaitées ou, dans le cas échéant, d'atteindre des nouveaux objectifs (réalisables) pour éviter dès que possible des trajectoires catastrophiques [Hoblos, 01].

Le système qui préserve la capacité d'accomplir ces missions est dit système tolérant aux défauts, nous allons définir cette notion tout de suite.

3.2.1. Le système tolérant aux défauts

Un système tolérant aux défauts possède la capacité de maintenir les objectifs nominaux en dépit de l'occurrence d'un défaut [Patton, 97], [Zhang et Jiang, 03a], [Zhang et Jiang, 08] et à s'en accommoder de manière automatique.

Un conventionnel gain de retour d'état peut s'avérer très limité et amener le système vers des comportements non désirés en présence d'un défaut. Pour pallier de telles catastrophes, de nouvelles approches de surveillance et diagnostic ont été développées dans le but précis de détecter, localiser et identifier le défaut au plus tôt possible afin de garantir le bon fonctionnement du système.

3.2.2. Techniques de tolérance aux défauts

La tolérance aux défauts est un ensemble de techniques combinant le traitement de défaut qui évite qu'il se reproduit et le traitement d'erreur qui tente de rattraper l'erreur avant qu'elle ne provoque une panne. Les travaux de recherche dans ce domaine sont nombreux mais obéissent plus ou moins à ce schéma général.

♦ Le traitement de défaut se fait en deux étapes :

- diagnostic de défaut : il consiste à identifier le défaut, en termes de nature ; origine et persistance.
- passivation de défaut : il vise à neutraliser la source de défaut de façon à l'empêcher de se reproduire.

♦ Le traitement d'erreur à son tour se fait de deux manières :

- compensation d'erreur: suppose que le système comprend une redondance qui lui permet de continuer à fonctionner en dépit de son état erroné, et donc la tolérance aux défauts est assurée grâce à la redondance interne du système qui peut être matérielle, logicielle ou bien temporelle.
- recouvrement d'erreur : transforme l'état erroné du système en un état correct

Ces techniques de tolérance aux défauts sont toutes réparties sur plusieurs phases, on en cite les principales: détection des défauts ; évaluation des conséquences ; traitement de l'erreur ; traitement de défaut et reprise du service.

Dans notre travail de thèse, nous ciblerons particulièrement la phase de détection des défauts pour effectuer le diagnostic.

Discutons maintenant les différents types des défauts.

3.3. Les différents types des défauts

Il n'y a aucune assurance que le comportement du système réponde aux objectifs prédéfinis après le lancement d'une application. Ce dysfonctionnement peut être à l'origine de plusieurs causes : apparition d'un défaut, une fausse information du capteur…. Les défauts sont des événements qui apparaissent à différents endroits du système. Dans la littérature, les défauts possibles dans un système sont classés en fonction de leur localisation :

A. Les défauts actionneurs

Les défauts actionneurs agissent au niveau de la partie opérative et détériorent le signal d'entrée du système. Ils représentent une perte totale (panne) ou partielle d'un actionneur agissant sur le système. Pour une perte totale d'un actionneur, citons comme exemples : tombé en panne d'une pompe ou d'un tapis roulant.

Les défauts actionneurs partiels sont des actionneurs réagissant de manière similaire au régime nominal mais en partie seulement, c'est-à-dire avec une certaine dégradation dans leur action sur le système (perte de puissance d'un moteur, fuite dans un vérin, . . .).

B. Les défauts capteurs

Ce type de défauts est la cause d'une mauvaise image de l'état physique du système. Un défaut capteur partiel produit un signal avec plus ou moins d'adéquation avec la valeur vraie de la variable à mesurer. Ceci peut se traduire par une réduction de la valeur affichée par rapport à la valeur vraie, ou de la présence d'un biais ou de bruit accru empêchant une bonne lecture. Un défaut capteur total produit une valeur qui n'est pas en rapport avec la grandeur à mesurer.

C. Les défauts composants ou systèmes

Ce type de défauts provient du système lui-même ; bien souvent les défauts n'appartenant pas aux deux premiers types sont classés de manière arbitraire dans cette catégorie. Néanmoins, un défaut composant résulte de la casse ou de l'altération d'un composant du système réduisant les capacités de celui-ci à effectuer une tâche. (Une chaufferie est cassée, un roulement est altéré, fuite dans un réservoir, . . .).

3.4. Modélisation des SDH pour la surveillance et le diagnostic

Comme nous avons indiqué au cours du chapitre précédent, une surveillance à base de modèles consiste à comparer le comportement prévu du système décrit par un modèle et celui réellement observé. Toute discordance entre les deux indique la présence d'au moins un défaut, que l'on cherchera à détecter. En effet, afin d'appliquer une telle méthodologie de surveillance, il faut disposer d'un modèle qui décrit le comportement nominal et de défaut du système. Ce modèle de défaut doit décrire les comportements défaillants possibles du système.

La modélisation d'un SDH consiste à établir des liens entre les différentes variables composant ce système. La nature et la complexité des modèles sont les principales différences qui peuvent différentier un modèle d'un autre. Ces modèles peuvent être prédictifs (équations d'état), qualitatifs (équation de confiance), structurels ou analytiques (graphes structurels), explicatifs (graphes causaux temporels) ou associatifs (système experts, reconnaissance des scénarios). Par contre la modélisation des défauts suppose l'acquisition d'une connaissance a priori des défauts que l'on souhaite diagnostiquer.

Généralement, en automatique, pour caractériser le comportement normal d'un système on utilise des modèles dits de bon fonctionnement, c'est-à-dire ne comportant aucun défaut. En surveillance, par contre, il faut généraliser le modèle pour qu'il puisse prendre en compte tous les états possibles y compris le comportement défaillant du système.

Comme nous avons indiqué dans le chapitre précédent, le niveau 3 de connaissance nécessite une modélisation de l'évolution dynamique de la défaillance. Des évolutions supplémentaires reliant les variables de défaillance sont ajoutées au modèle de bon fonctionnement. Une connaissance précise des phénomènes physiques est requise, et des données expérimentales du processus défectueux doivent pouvoir être utilisées.

L'importance de ce niveau de connaissance nous conduit à introduire un modèle mixte qui représente à la fois le comportement de bon fonctionnement (comportement en fonctionnement normal) et le comportement défaillant. Cette modélisation est illustrée dans la section 3.5.

Dans la définition 2.9 du Chapitre 2, nous avons souligné qu'une panne du système correspond à un état de dysfonctionnement, tandis qu'une défaillance, ou un défaut source d'une défaillance, correspond à un événement, qui peut mener vers un état de panne. Dans le contexte des SDH, l'apparition d'un défaut correspond au passage vers un état de dysfonctionnement. Ce passage peut être modélisé par une transition sur un défaut, une partition des états du système en états nominaux et états de dysfonctionnement est préalablement établie. Dans ce dernier cas, un système est déclaré en panne s'il atteint un état de dysfonctionnement.

S'il s'agit de pannes permanentes, le passage vers un état de début de panne se fait suite à l'occurrence d'un événement de défaut. Le système va évoluer ensuite vers d'autres états de pannes. S'il s'agit de pannes intermittentes, le système peut retourner vers un état de fonctionnement normal, suite à l'occurrence d'un événement de retour en fonctionnement normal.

Dans le cadre de notre travail, on s'intéresse à l'évolution du système en présence des défauts, donc, l'objectif est de représenter le système en comportement normal et en présence des défauts pour la sous classe des SDH considérée en modélisant avec une manière structurelle. La technique de modélisation sera détaillée dans la section suivante.

3.5. Modélisation des SDH à flux continus tolérant aux défauts

Beaucoup de travaux sur la modélisation des défauts dans les SDH ont été réalisés dans la littérature, et plusieurs techniques de modélisation ont été classées. Il y a les approches qui considèrent une modélisation à base d'événement [Lin et Wonham, 94], [Sampath *et al.*, 95]. D'autres considèrent une modélisation à base d'état [Zad *et al.*, 98], [Zad *et al.*, 99]. Une approche combinant les avantages de la modélisation à base d'événement et à base d'état est proposée dans [Sayed-Mouchaweh *et al.*, 08]. Un défaut peut également être représenté comme l'exécution d'un motif de supervision donné, qui est une propriété temporelle liée à l'occurrence

d'un ensemble de trajectoires/événements qui doit être diagnostiqué [Jéron *et al.*, 06]. Ces techniques utilisent des modèles contenant les comportements défaillants.

Il existe aussi des approches, qui utilisent des modèles sans défauts. Ils sont basés sur la comparaison des sorties des systèmes avec des sorties nominales des modèles. L'approche de modélisation sans défaut proposé par [Pandalai et Holloway, 00] utilise des modèles de condition pour déterminer si le système génère des événements dans le bon ordre ou au sein des retards de temps donnés. Dans [Sayed-Mouchaweh, 12] une connaissance experte est associée aux modèles de condition pour identifier les défauts liés à des événements manquants ou inattendus, et la surveillance progressive est utilisée pour réduire l'ensemble des défauts candidats après l'occurrence de nouveaux événements observables. Une autre approche de modélisation pratique sans défaut pour le diagnostic des défauts dans les systèmes de fabrication a été proposée dans [Roth *et al.*, 11].

En effet, il existe aussi des techniques qui utilisent le modèle RdP pour la modélisation des défauts dans un SDH tolérant aux défauts [Renganathan et Bhaskar, 11]. Parmi ces travaux, dans [Renganathan et Bhaskar, 13] les auteurs ont proposé une technique à base d'observateur pour détecter les défauts sur le système modélisé par les RdPH. Un autre travail a porté sur un diagnostic de défauts et le modèle d'analyse des causes utilisant une approche probante floue de raisonnement et les réseaux de Pétri flous adaptatifs dynamiques [Liu *et al.*, 13]. D'autres ont proposé de nouveaux algorithmes pour réaliser le diagnostic de défauts et le contrôle tolérant aux défauts avec un modèle basé sur l'outil de RdPH dans [David et Alla, 01]. Tous ces modèles sont essentiellement descriptifs et l'analyse se fait alors par simulation. Nous proposons dans notre travail d'apporter une analyse formelle en associant RdP et automates.

3.5.1. Le modèle du RDPH élémentaire

Nous avons basé notre approche sur la modélisation à base d'événements, celle-ci peut se mesurer par sa capacité à exploiter, d'une manière optimale, les deux aspects présentés par cette sous classe des SDH : l'aspect continu à travers les variables d'état continues et l'aspect discret à travers les événements discrets. Cet avantage reste valable en utilisant aussi bien le modèle RdPH élémentaire (Définition 1.13 du chapitre 1) ou un modèle d'une sous classe d'AH (Définition 1.15 du chapitre 1. En effet, il sera possible de modéliser un défaut sur l'aspect continu par une transition continue, et sur une variable discrète lorsque l'occurrence d'un événement inattendu se produit, ou lorsqu'un événement attendu ne se produit pas.

Pour modéliser les apparitions des événements, il faut connaître les liens temporels entre les différents évènements constituant les défauts. Ces liens définissent un ensemble de contraintes régissant l'occurrence des défauts.

- ♦ Ces contraintes sont mises comme des conditions temporelles sur les transitions discrètes modélisant les défauts sur la partie discrète (RdP T-temporel) du modèle RdPH élémentaire,
- ♦ Tout franchissement de ces transitions est le résultat d'un défaut qui change l'état du système,

♦ Chaque état est modélisé par un marquage particulier du RdP T-temporel. Ce marquage change à chaque fois qu'une transition est franchie.

En se basant sur ces principes, nous pouvons introduire les modèles utilisés dans notre méthodologie, à savoir les RdPH élémentaires et les AH linéaires. La question qui se pose maintenant est : lequel parmi ces deux modèles est le plus approprié pour la représentation de la sous-classe considérée des SDH en présence des défauts permanents ? C'est ce que nous verrons à la fin de ce chapitre après la présentation des deux modèles.

3.5.1.1. Le modèle du RDPH élémentaire en fonctionnement normal

Dans ce formalisme, le franchissement d'une transition continue décrit le flux continu, tandis que le franchissement d'une transition discrète modélise l'occurrence d'un événement qui peut changer l'état discret et ainsi par exemple modifier les vitesses de franchissement des transitions continues.

Le travail présenté dans cette thèse considère le formalisme des RdPH élémentaires, car il y a un découplage entre la partie discrète et la partie continue. Chaque partie peut influencer le comportement de l'autre, mais il n'y a pas de transformation du marquage discret en un marquage continu et vice-versa.

Ce modèle combine un RdP T-temporel et un RdP Continu à vitesse constante (RdPCC) [David et Alla, 10]. Une présentation formelle sera donnée dans la définition 3.1.

L'approche de modélisation est basée sur des techniques qui sont présentées ci-dessous. Nous verrons que cette construction est concise grâce à la modélisation par RdP, elle est structurelle et donc indépendante du marquage initial contrairement aux travaux de [Derbel *et al.*, 09].

Notre objectif est de construire le modèle complet (global) du système (modèle avec défauts), en deux modélisations indépendantes : 1) le système en fonctionnement normal, et 2) les défauts. Le modèle global s'obtient par composition structurelle de deux RdPH (fusion des transitions identiques).

Dans notre approche, les défauts seront considérés comme des événements non-observables, cela conduit à décomposer l'ensemble des événements en deux sous-ensembles : l'ensemble des événements observables et l'ensemble des événements non-observables ($\sum_o$ et $\sum_u$). Si un événement non-observable est associé avec une transition discrète, cette transition est dite non-observable.

Définition 3.1. Un RdPH élémentaire en fonctionnement normal est une structure $\boldsymbol{R_N}$ = (*P, T, Pré, Post, h,* $\sum$ *, I, V,* M_0), tel que :

-$P = \{P_1, P_2, \ldots, P_n\}$ est un ensemble de *n* places, $P = P^C \cup P^D$ avec :

P^C est l'ensemble fini de places continues (ou C-places) ;

P^D est l'ensemble fini de places discrètes (ou D-places) ;

- $T = \{T_1, T_2, \ldots, T_m\}$ est un ensemble de *m* transitions, $T = T^C \cup T^D$ avec :

T^C est l'ensemble fini de transitions continues (ou C-transitions) ;

T^D est l'ensemble fini de transitions discrètes (ou D-transitions) ;

-*Pré* : $P \times T \to N$ et *Post* : $P \times T \to N$ désignent respectivement les applications d'incidence avant et arrière; ces applications doivent satisfaire la condition suivante :

$$\forall (P_i, T_j) \in P^D \times T^C : Pré(P_i, T_j) = Post(P_i, T_j)$$

-*h* : $P \cup T \to \{D, C\}$ est une application qui désigne les nœuds discrets, *h(x)=D*, et les nœuds continus, *h(x)=C* ;

- $\sum$ est un ensemble fini d'événements ; $\sum = \sum_o \cup \sum_u$

$\sum_o$ est un sous ensemble des événements observables ;

$\sum_u$ est un sous ensemble des événements non-observables ;

$\sum \to Q^+ \times Q^+$ associe à chaque événement σ_j un intervalle d'occurrence $d_j = [\alpha_j \ \ \beta_j]$.

-*I*: $T^D \to \sum$ est une fonction qui associe à chaque transition discrète un événement de $\sum$;

-*V*: $T^C \to R^+$ est une application qui associe à chaque C-transition sa vitesse maximale de franchissement V_j :

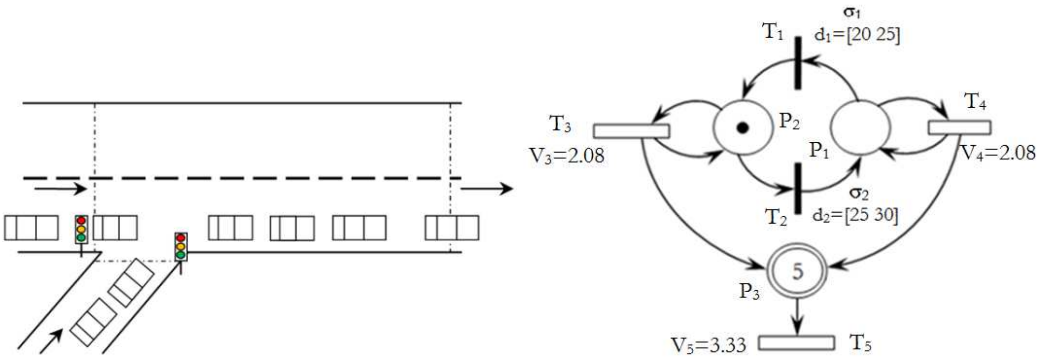
-M_0 est le marquage initial, tel que $M = (m_C \ \ m_D)^T$, m_C et m_D , le marquage continu et discret.

❑

Afin d'illustrer ce modèle, considérons l'exemple de la section de la route présenté auparavant, cette section peut contenir un nombre maximum de 150 véhicules (en supposant que la distance moyenne entre deux véhicules est de L= 4m).

Nous supposons que la section a une rampe d'entrée, les deux entrées de la section sont contrôlées par des feux de signalisation, le feu vert est allumé dans la première et la deuxième entrée, respectivement, après $d_1 = [20 \ \ 25]$ et $d_2 = [25 \ \ 30]$.

Les véhicules peuvent accéder à la section avec une vitesse moyenne de 30km/h à partir des deux entrées et sortent avec une vitesse moyenne de 48km/h.



(a) *(b)*

Figure 3.1. (a) Une section de la route. (b).RdPH élémentaire en fonctionnement normal.

La figure 3.1(b) représente le modèle du RdPH élémentaire de la section de la route. La partie continue est représentée par une double ligne, le place continue P_3 représente la section, son marquage correspond au nombre de véhicules. Initialement, il y a 5 véhicules dans la section.

Les transitions continues sont associées par leurs vitesses maximales de franchissement qui ont été calculés de la façon suivante :

$$\text{V}=\frac{\left(\text{Vitesse de entrée/sortie (k/h)}\times 1000\right)}{\left(\text{espace inter véhiculaire (m)}\times 3600\right)}$$

Alors :

$$V_3=\frac{\left(30km/h\times 1000\right)}{\left(4m\times 3600\right)}=2.08$$

$$V_4=\frac{\left(30km/h\times 1000\right)}{\left(4m\times 3600\right)}=2.08$$

$$V_5=\frac{\left(48km/h\times 1000\right)}{\left(4m\times 3600\right)}=3.33$$

La partie discrète est représentée par la ligne simple où les durées d'occurrence (d_1 et d_2) et les événements (σ_1 et σ_2) sont associés aux transitions discrètes. Ces durées correspondent aux délais d'occurrence des feux verts sur les deux entrées de la section. L'absence de feu vert correspond au feu rouge. On a mis un intervalle dans les durées pour assurer la coordination entre les deux entrées.

3.5.1.2. Le modèle d'un défaut unique

Notre objectif est de décrire le comportement des défauts indépendamment du comportement normal, et en ne décrivant que celui-ci. La modélisation des défauts sur la partie discrète n'est pas étudiée ici parce qu'elle est largement traitée dans la littérature. Seuls les défauts qui ont une conséquence sur la partie continue sont présentés.

L'occurrence des défauts sur l'aspect continu est ici modélisée par une transition continue contrôlée par l'occurrence d'un événement non-observable associé à une transition discrète. Ceci est défini ci-dessous. Nous traitons d'abord le cas d'un défaut simple (Figure 3.2).

Définition 3.2. Un modèle de défaut unique est un sous RdPH $\boldsymbol{R_F}$ qui contient :
- P_1, P_2, deux places discrètes ;
- T_1, est une transition discrète associée au défaut σ_f (événement non-observable) ;
- T_2, est une transition continue avec une vitesse de franchissement constante V_2 ;
- $Pré_f$: $P_2\times T_2\rightarrow N$ et $Post_f$: $P_2\times T_2\rightarrow N$ désignent respectivement les applications d'incidence avant et arrière de défaut tel que :

$$Pré_f(P_2,T_2)=Post_f(P_2,T_2)=1$$

- $M(P_1)=1$, $M(P_2)=0$, est le marquage initial.

❑

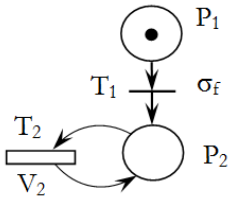


Figure 3.2. Modèle d'un défaut unique

Remarque 3.1. Si on veut avoir un flux additif à l'occurrence de défaut, un arc doit sortir de la transition T_2 vers une place continue P_3 et il est possible aussi d'avoir un flux soustractif. Pour cela, on doit inverser l'arc reliant la transition T_2 à une place continue P_3, comme le montre la figure suivante.

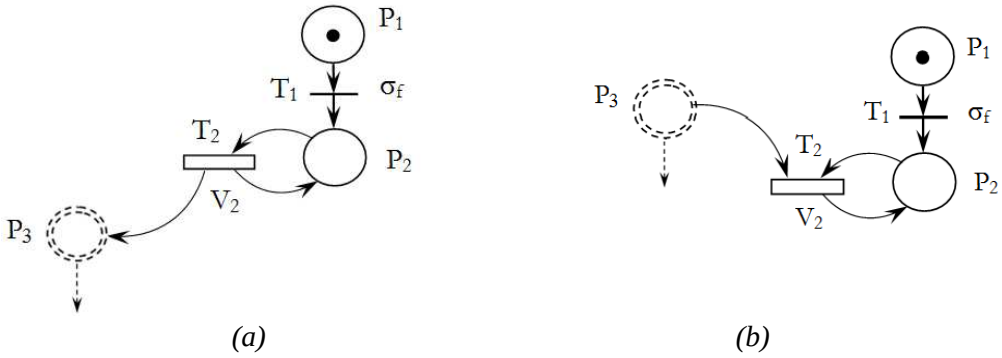
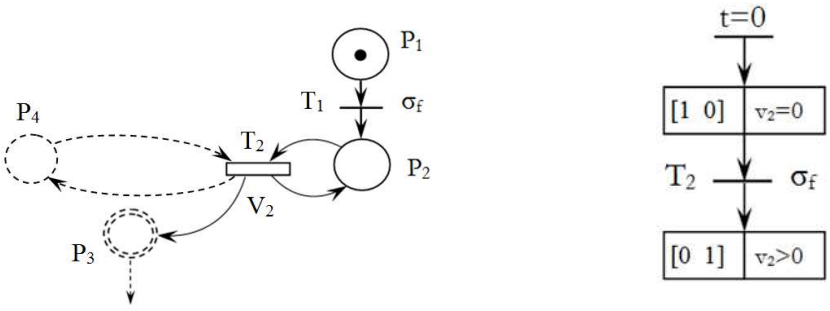


Figure 3.3. (a) Flux additif résultant d'un défaut unique. (b) Flux soustractif.

Complétant maintenant l'exemple de la section de la route. Une situation de défaillance du trafic peut être observée si le feu vert est allumé, dans les deux entrées. Le passage à cet état de défaillance est dû à l'occurrence de l'événement de défaut σ_f .

Le modèle RdPH de ce défaut est donné sur la Figure 3.4(a). Il modélise le défaut et son influence sur le comportement continu. Après l'apparition de défaut σ_f , P_2 est marquée et la transition continue T_2 est activée à sa vitesse de franchissement maximale V_2. Cela donne un flux additif à la place continue P_3. P_4 représente l'activation du feu vert dans la deuxième entrée. Les deux places P_3, et P_4 sont dessinées en pointillés car elles ne font pas partie du modèle de défaut. Le comportement dynamique du modèle de défaut est illustré par le graphe d'évolution donné à la Figure 3.4(b). Ce graphe d'évolution contient deux états. Le passage d'un état à un autre a lieu à l'apparition du défaut qui entraine une modification de la vitesse de franchissement de la transition continue. Cette vitesse devient strictement positive conduisant à un changement dans la dynamique du système qui pourra être avantageusement exploitée dans la synthèse d'un diagnostiqueur.



(a) *(b)*

Figure 3.4. (a) modèle de défaut par RdPH. (b) Graphe de marquage correspondant.

3.5.1.3. Le modèle du RDPH élémentaire en présence des défauts

Pour obtenir le modèle global, il est nécessaire d'effectuer une composition des deux modèles (modèles de fonctionnement normal et modèles de défauts). Cette composition est structurelle et indépendante du marquage initial. Elle s'obtient par la fusion des transitions identiques qui apparaissent dans les deux sous-modèles, celui du fonctionnement normal et celui du défaut. Le modèle global revient à ajouter au modèle en fonctionnement normal deux places à l'ensemble des places et deux transitions à celui des transitions. Cette composition est formellement définie ci-dessous.

Définition 3.3. Le modèle global $\boldsymbol{R_G}$ de RdPH obtenu par la composition des modèles $\boldsymbol{R_N}$ et $\boldsymbol{R_F}$ est un 8-uplet $\boldsymbol{R_G} = (P^G, T^G, Pré_g, Post_g, h, \sum_g, I, V, M_{0g})$, tel que :

- $P^G = \{P_1, P_2, \ldots, P_{n+2}\}$ est un ensemble global de $n+2$ places (P_1 et P_2 sont les deux places discrètes du $\boldsymbol{R_F}$) ;
- $T^G = \{T_1, T_2, \ldots, T_{m+2}\}$ est un ensemble global de $m+2$ transitions (T_1 et T_2 sont les transitions de défaut discrète et continue respectivement de $\boldsymbol{R_F}$) ;
- $Pré_g$ et $Post_g$ désignent respectivement les applications d'incidence globale avant et arrière;
- $h : P \cup T \rightarrow \{D, C\}$ est une application qui désigne les nœuds discrets, $h(x)=D$, et les nœuds continus, $h(x)=C$;
- $\sum_g$ est un ensemble fini d'événements ; $\sum_g = \sum_o \cup \sum_u + \{\sigma_f\}$
- I: $T^D \rightarrow \sum_g$ est une fonction qui associe à chaque transition discrète un événement de $\sum_g$;
- V: $T^C \rightarrow R^+$ est une application qui associe à chaque C-transition sa vitesse maximale de franchissement V_j :
- M_{0g} est le marquage initial, tel que $M_{0g} = [1 \;\; 0 \;\; M_0]^T$, ($M_0$ est le marquage initial de $\boldsymbol{R_F}$).

❑

Remarque 3.2.

- Pour des défauts multiples, la structure qui correspond à chaque type de défaut doit être dupliquée, comme il est illustré dans la figure ci-dessous.
- Un défaut qui correspond à un changement de structure conduit à un nouvel RdPH (ceci n'est pas formalisé dans notre travail).

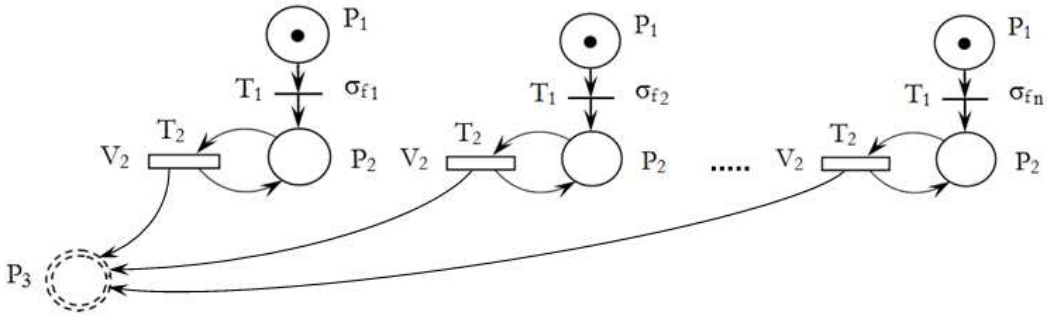


Figure 3.5. Flux additif résultant de défauts multiples.

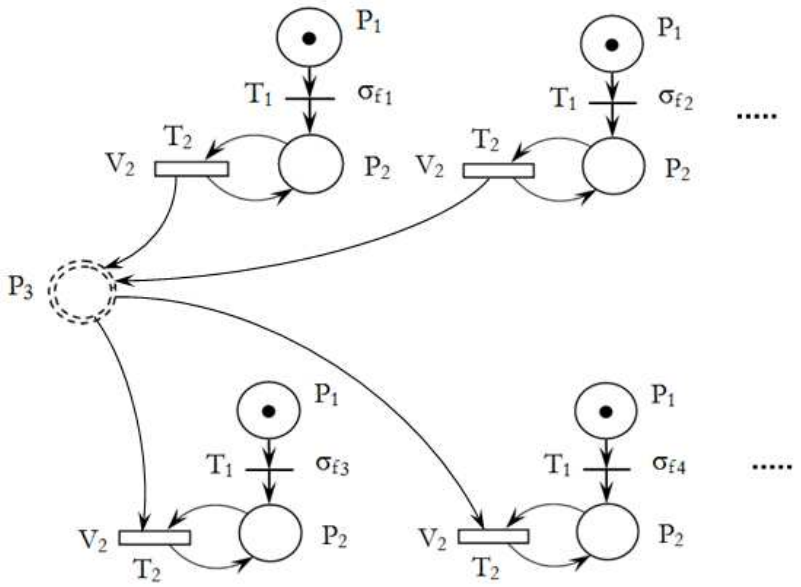


Figure 3.6. Flux additifs et soustractifs résultants de défauts multiples.

La Figure 3.7 présente le modèle global de la section de la route par RdPH élémentaire, issue de la composition du modèle en fonctionnement normal (sans défauts) et le modèle de défaut considéré. A partir du marquage de cette figure et en cas de défaut, la transition T_6 est franchissable. Deux flots de voitures s'écoulent alors dans le tronçon commun.

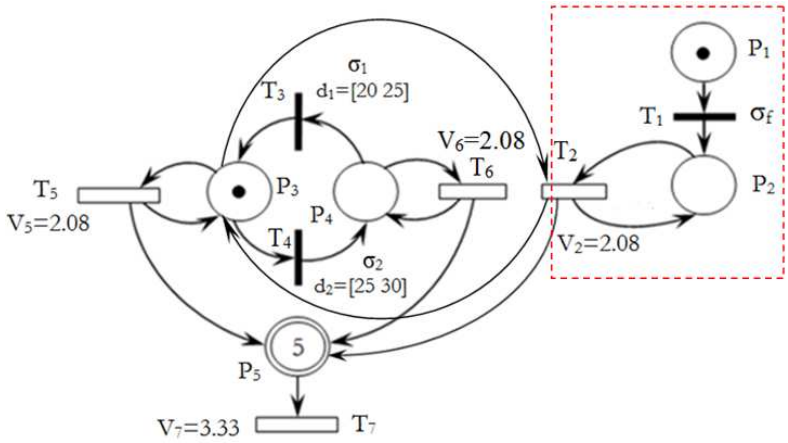


Figure 3.7. Modèle global de RdPH de section de la route.

3.5.2. Le modèle d'Automate hybride

Le RdPH est un outil concis et élégant de modélisation, mais son analyse nécessite souvent la simulation. L'automate hybride ou du moins certaines de ses sous-classes permettent une analyse formelle puissante. C'est pour cela que nous avons associé dans notre travail ces deux outils pour profiter des avantages apportés par chacun d'eux.

L'automate est un modèle à états discrets fini décrivant de manière explicite la dynamique d'un système à évènements discrets. L'automate hybride est une extension du modèle de base. Dans chaque sommet discret, la dynamique des variables continues est définie par des équations différentielles, et les transitions entre les états discrets dépendent des valeurs des variables continues. L'état d'un automate hybride à un instant *t* est déterminé par le couple (*q, x*) tel que *q* est un sommet discret et *x* est la valeur du vecteur d'état à l'instant considéré. A partir d'un état, le système peut évoluer : 1) soit en franchissant une transition discrète qui change le sommet actif et réinitialise certaines variables, 2) soit par la progression du temps dans le sommet courant ; cela entraîne un changement permanent de l'état continu conformément à la fonction d'évolution de ce sommet.

Nous définissons dans notre thèse seulement l'automate hybride linéaire car la dynamique d'un RdPH élémentaire correspond à cette sous-classe d'automate hybride. Et il s'avère que celle-ci permet également une analyse formelle, confortant ainsi notre démarche.

3.5.2.1. Les Automates Hybrides Linéaires (AHL)

L'automate hybride linéaire (Figure 3.8) est une sous-classe des automates hybrides.
Un automate hybride est dit linéaire si:

- ♦ La fonction d'évolution dans tous les sommets est une fonction linéaire de la forme: $\dot{x} = k$, avec *k* un vecteur constant.
- ♦ Les invariants des sommets ainsi que les gardes des transitions sont des prédicats linéaires de la forme : $Ax \prec b$; avec *A* un vecteur rationnel de dimension *n* et *b* une constante réelle. $\prec$ est une relation d'ordre $\prec \in \{=, <, >, \leq, \geq\}$.

Il s'agit ici d'une linéarité par rapport au temps, *i.e.* pour chaque variable, $x_i = k_i.t + x_0$. Les fonctions réinitialisation associées aux transitions sont des fonctions affines. i.*e.*, elles sont de la forme: $x' := Ax$, avec *A* une matrice carrée d'ordre *n*.

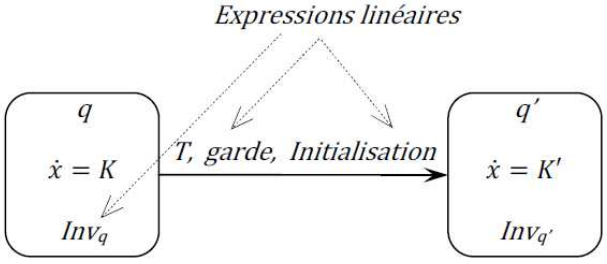


Figure 3.8. Automate hybride linéaire

L'intérêt des automates hybrides linéaires est le fait que tous les paramètres qui les définissent (invariants, gardes, conditions initiales, …) sont linéaires, et que si les régions qui défissent les espaces d'état atteignables, seront définis par la suite, sont convexes, alors leurs images par les fonctions linéaires d'évolutions continues et discrètes sont également des régions convexes.

Definition 3.4. L'automate hybride linéaire est un 7-uplet $H=(Loc, X, E, \delta, Dif, inv, q_0)$ tel que:
- *Loc* est l'ensemble fini des sommets (appelés aussi localités, situations),
- *X* est l'ensemble fini de variables réelles (vecteur d'état à composantes continues);
- *E* est l'ensemble fini d'étiquettes (i.e. ensemble d'actions événementielles liées aux franchissements de transitions),
- δ est un ensemble fini des transitions, chaque transition est un quintuple $T = (q, a, g, Init, q')$, tel que:
  - ♦ $q \in Loc$ est le sommet source;
  - ♦ $a \in E$ est l'événement associé au franchissement de *T*;
  - ♦ *g* est la garde de transition ;
  - ♦ *Init* est une fonction de réinitialisation qui affecte une expression linéaire aux variables de X en prenant la transition correspondante;
  - ♦ $q' \in Loc$ est le sommet cible;
- *Dif* fonction associant à chaque sommet $q \in Loc$ un ensemble de comportements continus (appelés aussi activités) *Dif*(*s*) :

$$\left.\frac{dx_i(u)}{du}\right|_t = x_i(t) = cste$$

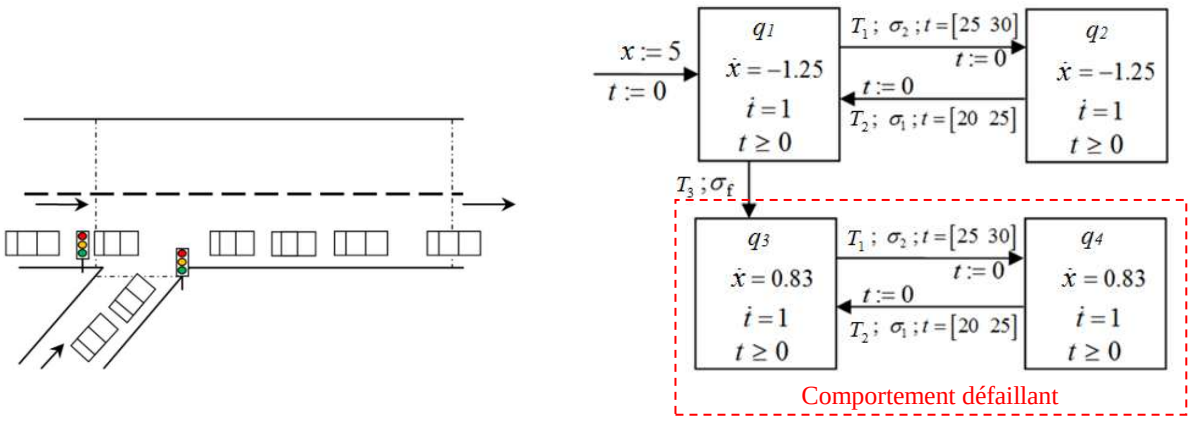
- *inv* est une fonction qui affecte à chaque sommet *q*, un prédicat linéaire *inv(q)* qui doit être satisfaite par les variables continues afin de rester dans le sommet *q*.
- $q_0 \in Loc$: sommet initial.

❑

Considérons cette fois aussi le même exemple de la section de la route avec la même situation de défaillance du trafic qui peut être observée si le feu vert est allumé dans les deux entrées. Le passage à cet état de défaillance est dû à l'occurrence de l'événement de défaut σ_f, il est modélisé par la transition T_3 associée à cet événement.

Ce défaut change directement le comportement continu du système. Ce changement est déclaré par un changement dans les valeurs des dérivés des vecteurs d'états (les valeurs des dynamiques continus) à l'instant considéré. La diférence entre deux dynamiques du même état représente le flux d'additif dans la section.

Le modèle d'AHL, en présence de ce défaut, est donné sur la Figure 3.9(b). Le système alors est dit en comportement défaillant, en rouge pointillés sur la même figure.



(a) *(b)*

Figure 3.9. (a) Une section de la route. (b) le modèle d'AHL correspondant.

3.5.2.2. Exécution d'un Automate Hybride Linéaire

Le comportement d'un automate hybride linéaire est défini par ses exécutions possibles qui sont des séquences de délais et de transitions discrètes. Nous pouvons définir formellement une exécution d'un automate hybride linéaire comme suit :

Définition 3.5. Une exécution φ est une séquence finie ou infinie de la forme :

$$\varphi = (q_0, x_0) \xrightarrow{t_0, f_0} (q_1, x_1) \xrightarrow{t_1, f_1} (q_2, x_2) \xrightarrow{t_2, f_2} \ldots (q_i, x_i) \xrightarrow{t_i, f_i} \ldots$$

Avec (q_i, x_i) est un état de l'automate hybride linéaire ; la fonction f_i est la fonction d'évolution dans le sommet q_i, et t_i est le temps de séjour dans le sommet q_i. Les conditions suivantes doivent être respectées :

- $f_i(0) = x_i$
- $f_i(t)$ vérifie $Inv(q_i)$ $\forall t \in [0, t_i]$
- $\forall i,\ \exists T = (q_i, a, g, Init, q_{i+1}) \in \delta$ tel que :

  $f_i(t_i)$ vérifie g ;

  $f_{i+1}(0) = Init(f_i(t_i))$.

❑

Un automate hybride est dit *déterministe* si, à partir d'un état initial donné, au plus une seule exécution (trajectoire) est possible. Il est dit non déterministe dans le cas contraire, *i.e.*, à partir d'un état initial, plusieurs exécutions sont possibles. La plupart des systèmes réels sont non déterministes. Le non déterminisme est une imprécision :

i. Dans les conditions initiales ;
ii. Dans les gardes des transitions ;
iii. Dans les fonctions d'évolution des sommets ;

Pour ces trois paramètres, les valeurs ne peuvent pas être précisément connues lors de la modélisation. Dans ce cas, les valeurs sont données sous la forme d'un intervalle comportant toutes les valeurs possibles. Le non déterminisme rend difficile l'analyse des SDH car, pour caractériser toutes les évolutions possibles, l'ensemble des trajectoires générées par le système doit être pris en compte. Pour un vecteur d'entrée constant, le problème n'est déjà pas facile à résoudre. Dans le cas où il y a une variation dans ce vecteur, même en utilisant des techniques de simulation numériques, il est difficile de simuler l'évolution du système pour toutes les valeurs du vecteur d'entrée. Les exemples 3.1 et 3.2 ci-après illustrent les notions de déterminisme et non déterminisme dans les SDHs.

Exemple 3.1. Considérons l'exemple classique d'un thermostat [Lygeros, 04] utilisé pour maintenir la température d'une chambre dans l'intervalle $\left[\theta_{min}, \theta_{max}\right]$. Ce procédé est composé d'un système de chauffage et d'un capteur de température. Le système de chauffage est en mode *ON* jusqu'à la détection du seuil supérieur θ_{max}, et il reste en mode *OFF* jusqu'au moment où la température descend en dessous d'un seuil inférieur θ_{min} .

Ce système peut être abstrait par un SDH où l'évolution continue est définie par la variation de la température *x* et l'évolution discrète par le passage du système de chauffage entre les états *ON* et *OFF*.

L'automate hybride en Figure 3.10(a) décrit le fonctionnement global de ce système. Dans l'état *OFF* l'évolution de la température correspond à un système du premier ordre $\dot{x} = -Ax$ avec $A > 0$, et dans l'état *ON* elle est décrite par $\dot{x} = -Ax + B$ avec $B > \theta_{max}$.

L'analyse d'un SDH revient à déterminer son espace d'état atteignable qui est déterminé par l'ensemble des exécutions possibles du système. Pour cet exemple simple, les solutions analytiques des équations différentielles peuvent être facilement trouvées. L'unique trajectoire de la température est présentée sur la Figure 3.10(c).

Exemple 3.2. Considérons à nouveau le système du thermostat, et supposons que le capteur ne soit pas précis et qu'il comporte un écart de mesure de valeur maximale ε . Cela implique une modification des gardes de transitions entre les états *ON* et *OFF* de l'automate hybride linéaire (figure 3.10(b)).

Les conditions de commutation d'un état vers l'autre de l'automate hybride linéaire, exprimées par les intervalles, signifient que le changement d'état peut se faire à n'importe quel instant dès que la température prend une valeur dans l'intervalle spécifié. Le comportement du système est non déterministe dans le sens ou pour une même condition initiale les trajectoires de la température peuvent être multiples (Figure 3.10(d)).

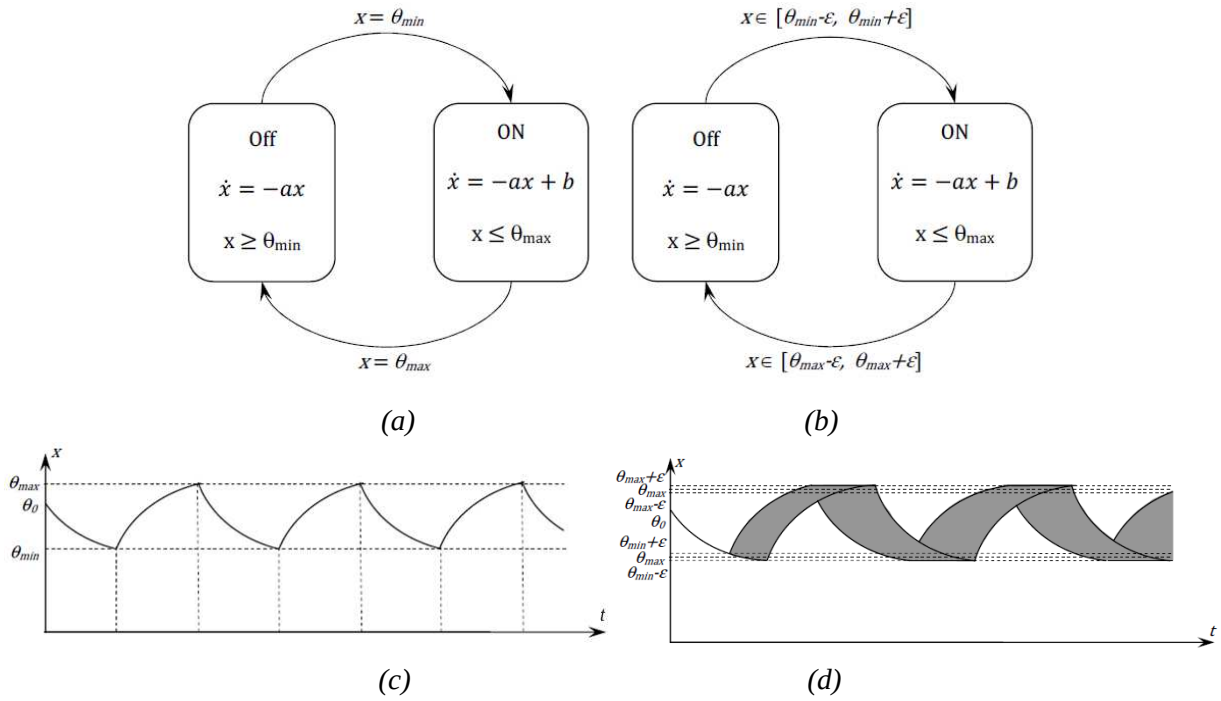


Figure 3.10. (a) automate hybride déterministe. (b) automate hybride non déterministe. (c) exécution de l'automate hybride de la figure 3.10(a). (d) exécution de l'automate hybride de la figure 3.10(b).

3.5.2.3. Analyse d'atteignabilité des Automates Hybrides Linéaires

L'espace d'états atteignables par un automate hybride linéaire, à partir d'un espace d'état initial (q, ϑ), où q est un sommet et ϑ un espace d'état, est défini comme étant l'ensemble des états visités par toutes les exécutions commençant à partir de (q, ϑ). Il existe deux types d'évolutions à partir d'un état initial, à savoir : une évolution continue, en restant dans le même sommet et en laissant le vecteur d'état évoluer suivant la fonction d'évolution du sommet, ou de manière discrète en franchissant une transition discrète.

Définition 3.6. Soit un ensemble d'état (q, ϑ) où $q \in \boldsymbol{Loc}$ et ϑ un espace d'état continu. On définit l'ensemble des successeurs continus de (q, ϑ), qu'on note $R_{cont}(q, \vartheta)$, comme suit:

$$R_{cont}(q, \vartheta) = \left\{ (q, x') / \exists x \in \vartheta, \exists t > 0, (q, x) \overset{t,\, f_q}{\rightarrow} (q, x') \right\}$$

❑

La Figure 3.11(a) ci-après représente le successeur continu de l'ensemble d'état (q, ϑ). Le comportement d'un automate hybride dans un sommet discret q est contraint par l'invariant de ce sommet délimité par un trait pointille sur la figure. Dans cette figure, l'état (q, z) ne fait pas partie des successeurs continus de l'ensemble (q, ϑ) tandis que l'état (q, y) en fait partie.

Définition 3.7. Soit une transition *T*, dont la garde est *g* et la fonction de réinitialisations *Init*, reliant *q* à *q'* et ϑ un espace d'état continu, où *q*, *q'*$\in$ ***Loc***. On définit l'ensemble des successeurs discret de (q,ϑ), par rapport à la transition *T*, qu'on note $R_{dis}(q,\vartheta)$, comme suit :

$$R_{dis}(q,\vartheta) = \{(q',x') / \exists x \in \vartheta \cap g \wedge x' \in Init \cap Inv(q')\}$$

❑

Considérons la Figure 3.11(b), le successeur discret de l'ensemble d'état (q,ϑ) est représenté par l'espace d'état gris. Cet espace est l'intersection de la garde *g*, représentée par un rectangle et de l'invariant du sommet but *q'*. Il est supposé ici que le vecteur d'état n'est pas réinitialisé.

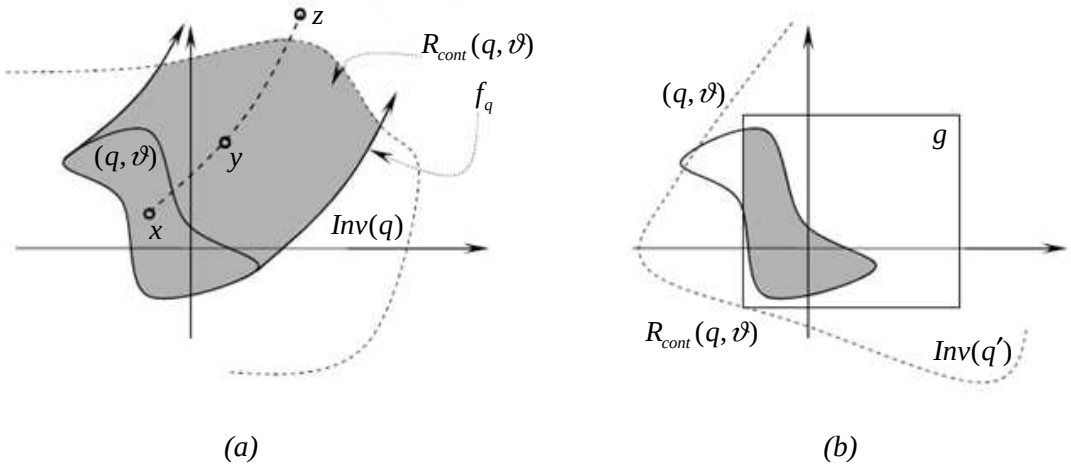


Figure 3.11. (a) Successeur continu. (b) successeur discret.

À l'exception de quelques classes très simples d'automates hybrides (les AHL en font partie), le calcul de l'espace d'états atteignables est généralement complexe. Le problème d'accessibilité (ou atteignabilité) constitue un problème fondamental pour l'analyse et la vérification des systèmes modélisés par automates hybrides, il est prouvé être non décidable [Alur *et al.*, 95], [Henzinger *et al.*, 95] (voir ANNEXE A). Le calcul de l'espace atteignable peut être effectué à travers les automates des régions proposées dans [Alur et Dill, 94], [Alur, 99]. Ce formalisme regroupe les différentes valeurs possibles des variables dans les sommets sous forme de polyèdres. Cette construction peut être infinie si les variables divergent.

Les automates hybrides rectangulaires initialisés [Henzinger et Rusu, 98] constituent aussi une classe décidable pour le problème d'atteignabilité. Cependant, le problème d'atteignabilité est non décidable pour les automates rectangulaires d'une manière générale [Puri et Varaiya, 94]. Dans [Asarin *et al.*, 12], les auteurs considèrent une classe particulière des AHL ayant des trajectoires continues qu'ils appellent PCD (Piecewise Constant Derivative, en anglais). Ils prouvent que le problème d'atteignabilité est décidable pour les PCD de dimension 2. Cependant, le problème devient non décidable pour les PCD de dimension supérieure ou égale à trois. Le problème d'atteignabilité reste ouvert pour cette classe.

Le calcul d'atteignabilité est complexe, cela oblige souvent à calculer une sur-approximation de cet espace. L'algorithme de base pour le calcul de l'espace des états accessibles est un calcul de point fixe. À partir de l'espace des états initiaux, on ajoute l'espace des successeurs continus et l'espace des successeurs discrets jusqu'à ce que cet espace se stabilise, et donc un point fixe est atteint. Le logiciel PHAver (voir ANNEXE B) permet de déterminer l'espace des états accessibles. Lorsque l'algorithme de calcul converge, il donne cet espace pour chaque sommet sous la forme d'inégalités entre les différentes variables continues. Cette formalisation analytique sera utilisée plus tard pour le calcul des nouvelles gardes dans la synthèse de l'automate atteignable. Le problème d'atteignabilité est un facteur intéressant qui sera exploité dans le dernier chapitre afin de justifier le modèle utilisé pour terminer la partie diagnostic.

3.5.3. Présentation de l'approche de surveillance et diagnostic

Afin de résoudre la problématique présentée au début de cette thèse, nous avons développé une méthode de représentation des défauts dans les systèmes hybrides à flux continu tolérant aux défauts. Nous avons pu voir dans ce chapitre que les événements non observables jouent un rôle d'un facteur principal qui caractérise en général l'occurrence des défauts et en particulier les transitions non observables qui peuvent conduire le système vers des états de dysfonctionnement ou de défaillance. Notre approche de modélisation se base essentiellement sur ce facteur, dans laquelle on élabore un modèle global qui représente à la fois tous les comportements possibles en présence du type de défauts considéré.

Nous considérons que les systèmes étudiés peuvent évoluer dans plusieurs modes de fonctionnement différents, comme illustré par la Figure 3.12. Chacun de ces modes à une dynamique propre. Par ailleurs, on peut distinguer parmi eux quatre catégories :

♦ Les modes de fonctionnement initial: ces modes de fonctionnement sont ceux à partir desquels le système démarre. Les paramètres qui caractérisent l'évolution du système sont initialisés dans ces modes.
♦ Les modes de fonctionnement normal: ce sont les modes de fonctionnement prévu par l'opérateur. La présence du système dans ces modes, le conduit inexorablement vers une exécution correcte et sans violation du cahier des charges.

♦ Les modes de dysfonctionnement: ce sont les modes où le système évolue avec des dynamiques pouvant l'amener soit à une défaillance soit à une violation du cahier des charges. La présence du système dans ces modes de fonctionnement n'est pas toujours synonyme de défaillance. Parfois on a recours à ce type de fonctionnement pour corriger les effets de certains imprévus qui apparaissent en cours de fonctionnement. Donc, la présence du système de ce mode de fonctionnement ne déclenche pas systématiquement l'alarme. Nous pouvons citer comme exemple deux applications ; la première celle d'un moteur qui voit sa vitesse de rotation varier de sa vitesse nominale de fonctionnement, soit elle augmente soit elle diminue, suite à un dysfonctionnement ou même à une mauvaise manipulation de l'opérateur. La deuxième application est celle d'une vanne dont le débit peut varier selon la présence ou non de résidus dans les tuyaux.

♦ Les modes de blocage: ce sont les modes où le système s'arrête complètement. Dans certains systèmes, il existe qu'un seul mode de blocage.

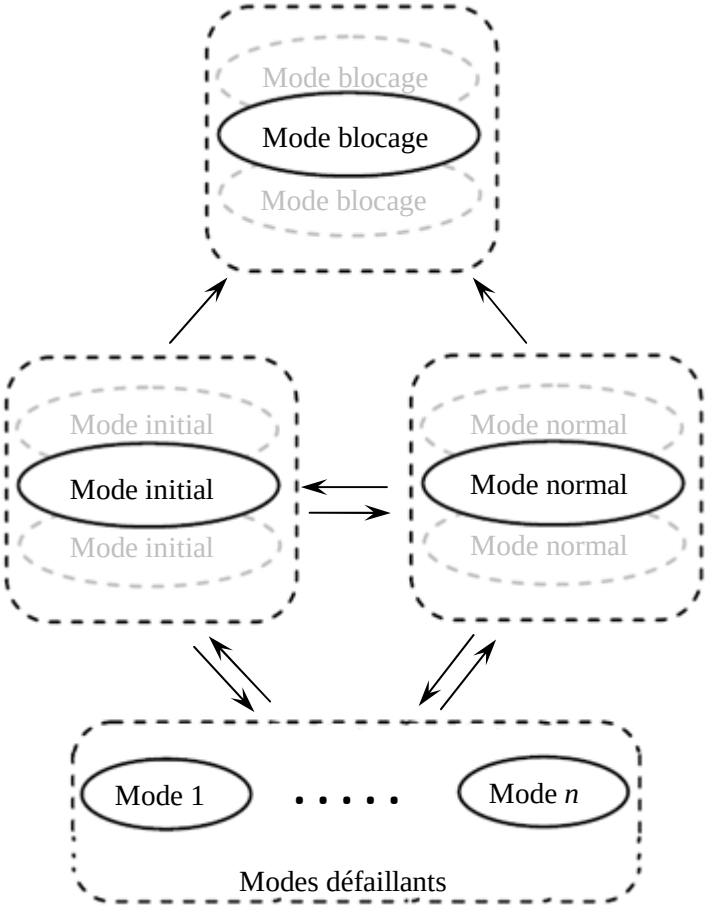


Figure 3.12. Modes de fonctionnement d'un système hybride tolérant aux défauts

L'élaboration d'une méthode de modélisation des systèmes dynamiques hybrides tolérants aux défauts a l'avantage d'avoir un champ d'application élargi. Nous allons tout de suite expliquer la démarche de modélisation que nous avons élaborée [Tolbi *et al.*, 16].

Dans un premier temps nous modélisons le comportement du système en fonctionnement normal en utilisant le modèle de RdPH élémentaire. On établit les modèles caractérisant chaque défaut qui peut apparaître sur le système en utilisant une modélisation indépendante des uns des autres avec le même outil. Puis, on effectue une composition des deux modèles (modèles de fonctionnement normal et modèles de défauts). Cette composition est structurelle et indépendante du marquage initial du RdPH élémentaire. Par la suite, on applique la procédure de translation proposée [Tolbi *et al.*, 16] pour avoir un modèle d'AHL modélisant le système en présence des défauts. Finalement, on applique une procédure d'analyse et synthèse à cet automate pour garder uniquement les trajectoires qui satisfont les propriétés obligatoires en utilisant le logiciel PHAVer. Les sommets de l'automate représentent les états atteignables du système, les équations différentielles relatives à chaque sommet reflètent la situation du système dans cet état.

Le passage d'un sommet à un autre est réalisé par le franchissement de la transition qui les lie. Ce franchissement est conditionné par l'occurrence de l'événement correspondant à cette transition quel que soit observable ou non observable. L'espace des états atteignables synthétisé dans chaque sommet comporte toutes les situations correspondant au comportement normal et défaillant.

La combinaison entre le pouvoir de modélisation des événements non observables qui caractérisent les défauts et le pouvoir de l'analyse d'atteignabilité de l'automate constitue l'originalité de notre approche de modélisation. Ceci nous permet de déterminer des espaces temporels décrits par des inégalités algébriques relatives à chaque sommet de l'automate. La violation de ces espaces permet de détecter les défauts.

Cette approche de modélisation représente la grande partie de notre approche globale du diagnostic illustrée sur le schéma de la Figure 3.13. Le reste de l'approche globale sera présenté dans le dernier chapitre.

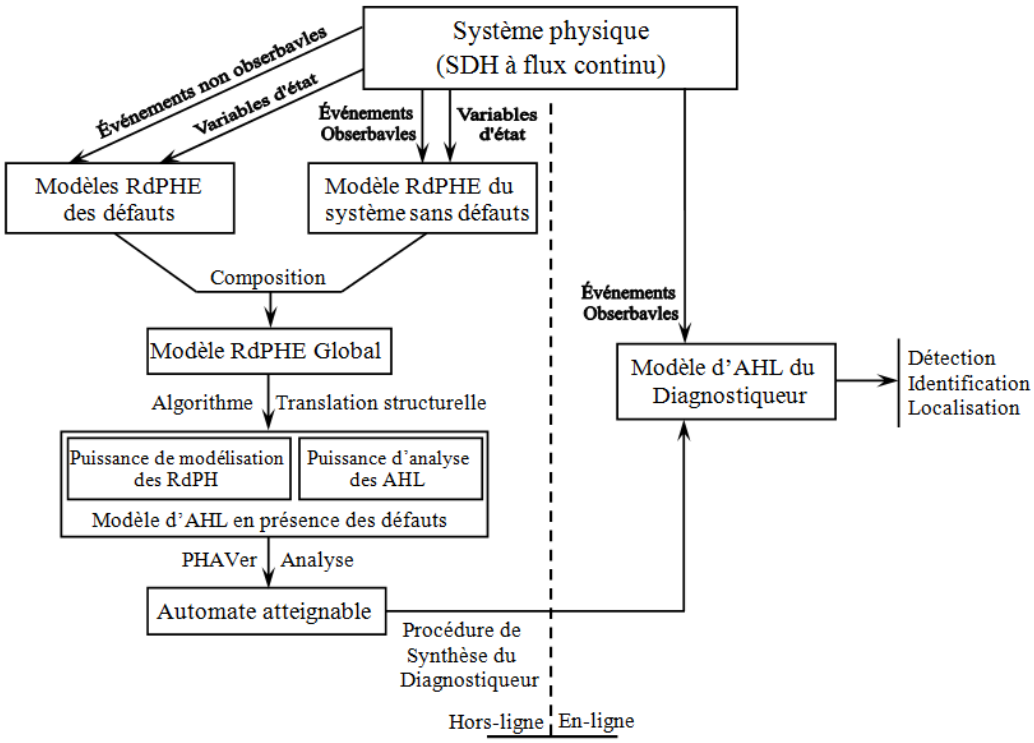


Figure 3.13. Schéma de notre approche globale de diagnostic à base d'automates hybrides linéaires.

3.6. Conclusion

Dans ce chapitre, nous avons présenté en premier volet la terminologie de la tolérance aux défauts en présentant le système tolérant aux défauts et les différentes techniques de tolérance qui existent dans la littérature. Les différents types de défauts qui peuvent se produire dans un SDH ont été ensuite décrits avec une étude des techniques de modélisation. Cela nous a permis de choisi le chemin pour modéliser le SDH considéré dans notre travail.

Dans un deuxième volet, nous avons présenté les deux outils de base que sont les RdPH et les AH. Les RdPH ne nécessitent pas une énumération exhaustive de l'espace d'états et permettent de représenter de manière finie un système dont l'espace atteignable est infini. Il comprend dans le même formalisme, la représentation des comportements dynamiques discrets et les comportements à événements discrets. Toutefois, la tâche d'analyse formelle du modèle est rarement possible, seule la simulation est accessible.

Les automates hybrides constituent un outil d'analyse puissant, cet outil permet une analyse formelle des SDH par un calcul analytique de l'espace atteignable. Cela nous a donné l'idée de coupler la puissance d'analyse des AH à la puissance de modélisation des RdPH pour avoir un approche combinant les avantages des deux modèles. Cela répond à la question précédemment posée dans la section 3.5.1: lequel parmi ces deux modèles est le plus approprié pour la représentation de la sous-classe considérée des SHD en présence des défauts permanents ?. L'association des deux modèles est réalisée en effectuant une translation structurelle du RdPH élémentaire en présence des défauts vers un AHL correspondant. Cette translation sera détaillée dans le chapitre suivant.

CHAPITRE 4

DU RÉSEAUX DE PÉTRI HYBRIDES ÉLÉMENTAIRES VERS LES AUTOMATES HYBRIDES LINÉAIRES : TRANSLATION ET BISIMILARITÉ

Résumé : Dans ce chapitre nous présentons une méthode de translation structurelle des RdPH élémentaires en automates hybrides linéaires. Nous présentons d'abord une introduction pour justifier les raisons d'être de cette translation. Nous présenterons par la suite quelques techniques de translation des RdP en automates rencontrés dans la littérature. Ensuite, nous illustrons le principe de notre méthode de translation de manière intuitive, puis nous proposerons un algorithme permettant une translation systématique. Une technique de bisimilarité temporelle sera présentée et prouvée pour démontrer la similarité des deux modèles en termes d'un système de transitions temporisé. Enfin, nous illustrons le principe de cette méthode sur un exemple illustratif en présence de défauts.

4.1. Introduction

L'approche de modélisation que nous proposons dans notre travail de recherche allie la capacité de modélisation du formalisme RdPH à la puissance d'analyse de l'automate hybride. Ainsi, une étape importante dans notre approche consiste à construire un automate hybride qui modélise le même comportement global que celui du RdPH élémentaire pour le système considéré.

Les RdPH présentent l'avantage d'avoir une modélisation claire et ne nécessitent pas une énumération exhaustive de l'espace d'état. Les automates (discrets, temporisés ou hybrides) sont des modèles formels qui permettent une manipulation facile, cependant ils sont mal adaptés à la modélisation. D'où l'idée de coupler la puissance d'analyse des automates à la puissance de modélisation des RdPH.

En général, la translation d'un RdPH à un automate hybride est complexe à cause de la forte interaction qui existe entre la dynamique discrète et la dynamique continue. Dans notre travail, notre choix s'est porté sur les RdPH élémentaires qui séparent la dynamique de la partie discrète de la dynamique de la partie continue, il n'est y a pas de transformation de marquage, du discret vers le continu ou vice-versa. Ce modèle est suffisamment général pour permettre l'étude d'une grande classe de SDH, de plus il permet une construction structurelle de l'automate hybride comme nous le verrons dans la suite de ce chapitre. De ce fait, nous proposons une méthode structurelle permettant de traduire un RdPH élémentaire en un automate hybride linéaire. Le RdPH élémentaire est un modèle issu du RdPH, il combine un RdP continu à vitesse constante (RdPCC) et un RdP T-temporel. Les comportements dynamiques des parties continue et discrète s'influencent mutuellement à travers des boucles sur les transitions.

Nous passons d'abord n revue des techniques similaires utilisées en littérature. Nous présenterons par la suite les spécificités du modèle RdPH élémentaire. Ensuite, nous illustrons le principe de notre méthode d'une manière intuitive. Nous proposerons, dans ce qui suit, un algorithme permettant de traduire de manière systématique un RdPH élémentaire en AHL. Nous réserverons la section qui suit pour démontrer que les deux modèles ont formellement le même comportement, et ce, en termes de bisimularité temporelle. Une preuve mathématique pour cette dernière sera présentée. Enfin, nous illustrons le principe de notre méthode de translation sur un exemple classique simple et nous analyserons l'atteignabilité de l'automate hybride résultant de cette translation générée avec le logiciel PHAVer.

4.2. Translation des RdP en Automates

La relation entre les réseaux de Pétri et les modèles basés sur les automates a été largement étudiée en littérature. Une comparaison en termes de puissance de modélisation est effectuée dans [Berthomieu *et al.*, 06], [Haar *et al.*, 00], [Haar *et al.*, 02], [Bérard *et al.*, 08], [Srba, 08]. La relation entre les RdP temporels et les automates temporisés a été étudiée, en premier lieu, dans [Sifakis et Yovine, 96] où les auteurs étudient une sous-classe des réseaux de Pétri temporels à flux dont le réseau sous-jacent est sauf (STPN). Dans ce modèle, un intervalle de temps [*a* , *b*]

est associé à l'arc reliant une place à une transition. Un jeton arrivant dans la place est indisponible pendant une durée comprise entre *a* et *b* avant d'être disponible pour le tir de la transition. Les auteurs proposent de translater de tels RdP en automates temporisés en associant une horloge à chaque place du RdP. L'horloge est réinitialisée à chaque fois que la place reçoit un jeton. Les sommets et les transitions de l'automate se déduisent directement à partir du graphe des marquages du RdP. Les intervalles associés aux arcs définissent les gardes et les invariants de l'automate.

Dans [Bornot *et al.*, 98], Bornot, Sifakis et Tripakis s'intéressent à une catégorie de RdP saufs et étendus avec des horloges appelés RdP à échéances (PND). Un PND est translaté en un modèle automate équivalent appelé automate temporisé à échéances (TAD). Le graphe des marquages du réseau de Pétri à échéances définit la structure discrète du TAD. On impose les mêmes contraintes temporelles pour les transitions du TAD et du PND. Le nombre d'horloges des PND et TAD est également le même. Les auteurs proposent une translation des réseaux de Pétri temporels en TAD avec une horloge par arc entrant du réseau de Pétri temporel initial. Les automates temporisés à échéances pouvant être considérés comme des automates temporisés standards. La méthode fournit ainsi, une translation des RdP temporels dont le RdP sous-jacent est sauf vers les automates temporisés. Le résultat possède un nombre d'horloges supérieur ou égal au nombre de transitions du réseau de Pétri initial.

Sava et Alla proposent dans [Sava, 01; Sava et Alla, 06] de translater un réseau de Pétri T-temporel(TPN) borné en automate temporisé (TA). Cette méthode se base sur une analyse dynamique du comportement du TPN en analysant le graphe de marquage du TPN ainsi que l'espace d'état de chaque marquage. Chaque marquage correspond à un sommet du TA et chaque franchissement d'une transition du TPN correspond à l'exécution d'une transition du TA. On associe une horloge dans le TA à chaque transition du TPN. L'intervalle de franchissement des transitions du TPN définit les gardes des transitions ainsi que les invariants de sommet du TA. Bien que la complexité de cet algorithme dépende exponentiellement du nombre d'horloges du TPN, cet algorithme s'avère très adéquat pour les analyses et les vérifications du TA obtenu. D'ailleurs, les auteurs utilisent le TA obtenu pour définir la synthèse de la commande des systèmes à événements discrets temporisés modélisés par un TPN borné.

Dans [Guillaume *et al.*, 03; Guillaume *et al.*, 06], les auteurs ont proposé une méthode pour appliquer le calcul du graphe des régions des automates temporisés aux réseaux de Pétri T-temporels bornés, en utilisant des DBM (Matrice de différences, en anglais Difference Bound Matrices). Ce calcul est utilisé pour générer un automate temporisé ayant un comportement temporel bisimilaire à celui défini par le graphe de marquages du RdP. L'algorithme de translation est inspiré de l'algorithme défini par Sava et Alla dans [Sava, 01; Sava et Alla, 06]. Cet automate possède un sommet par marquage et une horloge par transition. Les auteurs appliquent les algorithmes de réduction du nombre d'horloges de Daws et Yovine [Daws et Yovine, 96], ce qui réduit le nombre d'horloges final.

Cassez et Roux ont développé une méthode de translation structurelle d'un réseau de Pétri T-temporel en un automate temporisé [Cassez et Roux, 03 ; Cassez et Roux, 04]. Chaque transition

du RdP est modélisée indépendamment par un automate temporisé avec variables. Ensuite, tous les automates ainsi obtenus sont combinés avec un superviseur à travers un produit synchronisé d'automates temporisés. Cette méthode est définie d'une manière structurelle, et est par conséquence applicable à une large catégorie des réseaux de Pétri T-temporel y compris ceux non bornés. Cependant, l'analyse et la vérification des automates obtenus à travers cet algorithme s'avèrent très compliquées : en effet, la structure de produit du résultat final fait qu'on combine plusieurs horloges dans le calcul, ce qui est très complexe à gérer. En plus, le produit d'automates possède une horloge par transition du réseau de Pétri T-temporel et ce nombre ne peut être réduit par des algorithmes comme celui de Daws et Yovine [Daws et Yovine, 96] à cause de la structure de produit du résultat.

Dans [Balaguer et Chatain, 12], les auteurs proposent une méthode similaire à la translation structurelle de TPN, dont le RdP sous-jacent est borné, qui minimisent le nombre d'horloge global de l'automate. Cette méthode reste spécifique à des systèmes particuliers qui peuvent être décomposés (sur le plan du procédé, ainsi qu'au niveau du modèle) et ne peut, par conséquent, être généralisée au TPN en général. Dans [Lime et Roux, 06], Lime et Roux proposent une extension du graphe de classe d'états sous forme d'automate temporisé bisimilaire appelé graphe de classe étendu ce qui permet de vérifier les propriétés temporelle du TPN sur l'automate obtenu.

4.2.1. Translation des RdPH en AH

Dans [Allam, 98], Allam présente l'idée d'associer la capacité de modélisation du RdP hybride à la puissance des automates. Ceci est matérialisé par une approche d'analyse quantitative du comportement du RdP hybride temporisé à travers le modèle automate hybride en proposant un algorithme permettant de construire systématiquement l'automate hybride à partir d'un RdPH temporisé donné.

En s'inspirant du travail de Allam [Allam, 98], Demongodin et Rouibia [Demongodin et Rouibia, 03] ont présenté une procédure qui permet de systématiser le passage des RdP lots en automates hybrides. L'automate résultat a un sommet pour chaque IB-état du RdP lots. Si ce dernier vérifie les conditions de bornitude et de rationalité des temporisations associées aux transitions discrètes, les auteurs montrent la convergence de l'algorithme. La méthode d'analyse en avant est ensuite utilisée pour calculer la région atteignable, à partir de la région initiale définie par le marquage initial. Le calcul de la région atteignable, permet de décrire le fonctionnement périodique du RdP lots.

Dans [Alla et David, 98] est présenté un algorithme permettant la construction de l'automate hybride équivalent au RdPH. L'automate résultant a autant de sommets que d'IB-états du RdPH, les variables d'états étant les marquages des places continues et les horloges mesurant le temps pour les transitions validées. L'automate résultat est linéaire et déterministe. Pour assurer la convergence de cet algorithme de translation, le RdPH doit être borné.

Ghomri a proposé dans sa thèse [Ghomri, 12], un algorithme de translation des RdPH D-élémentaires en automates hybrides. L'algorithme procède en deux étapes élémentaires :

translation des RdP T-temporels en automates temporisés et translation des RdPCC en automates hybrides. Le nombre de sommets de l'automate hybride résultant de l'algorithme est linéaire.

S'inspirant des travaux de [Sava et Alla, 06], les auteurs dans [El Touati *et al.*, 09] ont présenté une procédure qui permet de systématiser le passage des RdP temporels étendus (RdP-TE) en automates hybrides linéaires (AHL). Le comportement temporel de l'AHL obtenu est similaire au comportement du RdP-TE puisque le nombre de sommets est égal au nombre de marquages du graphe de marquage. Les horloges actives dans un sommet reflètent le comportement des transitions validées. Les variables des tâches sont aussi représentées dans les sommets et leurs valeurs sont préservées par les affectations associées à l'AHL.

Plusieurs problèmes, liés à l'analyse des propriétés des automates hybrides ont pu être exprimés comme le problème de l'accessibilité. L'existence d'outils informatiques permettant l'analyse du problème d'accessibilité pour quelques classes d'automates hybrides fait que l'analyse de plusieurs formalismes de systèmes hybrides est faite après leur translation en automates hybrides [Sava, 01] , [Ghomri, 12] et [El Touati, 13].

Dans notre approche de translation, nous définissons un modèle spécifique d'automates hybrides linéaires, associé au modèle du RdPH élémentaire qui a été défini dans le chapitre précédent. Nous allons d'abord fournir les spécifications de ce dernier, puis le nouveau modèle sera présenté dans la section suivante.

4.2.2. Spécifications du RdPH élémentaire

Dans [Ghomri, 12] est présenté le RdPH D-élémentaire combinant un RdPCC et un RdP T-temporel. Dans ce modèle, le comportement de la partie discrète commande la partie continue sans que cette dernière puisse influencer la partie discrète. Le non déterminisme associé à ce modèle est représenté par les intervalles de franchissement associés aux transitions discrètes.

Le modèle RdPH que nous considérons dans cette thèse est appelé RdPH élémentaire et diffère de celui proposé dans [Ghomri, 12] par le fait que les deux parties interagissent entre elles. Les spécifications de ce modèle sont illustrées à travers l'exemple ci-dessous. Dans ce modèle, nous considérons qu'un événement n'a pas une date exacte d'occurrence mais un intervalle dans lequel l'événement a la possibilité de se produire.

Exemple 4.1. Considérons le système de réservoirs illustré par la Figure 4.1. Ce système Comporte 3 réservoirs, et 5 vannes. Les réservoirs 1 et 2 sont approvisionnés (remplis) par la vanne 1 et la vanne 2 avec un débit de 2 et 5 litres/sec respectivement. Le réservoir 1 (le réservoir 2) et le réservoir 3 sont reliés à l'aide de la vanne 3 (vanne 4) ayant un débit de 3 litres/sec (6 litres/sec). Le réservoir 3 est vidé par l'intermédiaire de la vanne 5 avec un débit de 7 litres/sec.

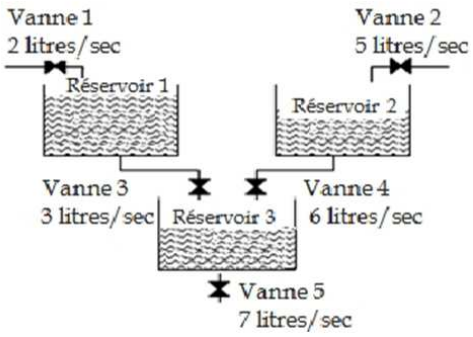


Figure 4.1. Système des trois réservoirs.

Le modèle RdPH élémentaire de la Figure 4.2 modélise le système des réservoirs de la Figure 4.1. Les places (transitions) continues sont représentées par un double cercle (trait) pour les distinguer des places et transitions discrètes. Le franchissement des transitions T_5, T_6, T_7, T_8, T_9 représente le flux traversent la vanne1, la vanne 2, la vanne 3, la vanne 4 et la vanne 5 respectivement. Les places P_5, P_6, P_7, représentent les quantités d'eau dans le réservoir 1, le réservoir 2 et le réservoir 3 respectivement. Le franchissement des transitions continues T_2 et T_6 est conditionné par le marquage des places discrètes P_2 et P_4 (influence de la partie discrète sur la partie continue).

Les vannes 3 et 4 peuvent être dans l'état ouvert représenté par les places P_2 et P_4 et l'état fermé représenté par les places P_1 et P_3. Les transitions discrètes T_2 et T_3 représentent le passage de l'état ouvert à l'état fermé qui prend 3 sec à 5 sec. Ce passage à l'état fermé des vannes 3 et 4 est conditionné par la quantité d'eau dans les réservoirs 1 et 2 respectivement. Ce sont les boucles T_2-P_5 et T_3-P_6 (influence de la partie continue sur la partie discrète).

D'autre part, le passage de l'état fermé à l'état ouvert a lieu après 10 sec de la dernière action d'ouverture, l'intervalle de temps [10, 10] est associé aux transitions discrètes T_1 et T_4. On suppose que, les réservoirs 1, 2 et 3 contiennent initialement 26, 10 et 12 litres respectivement.

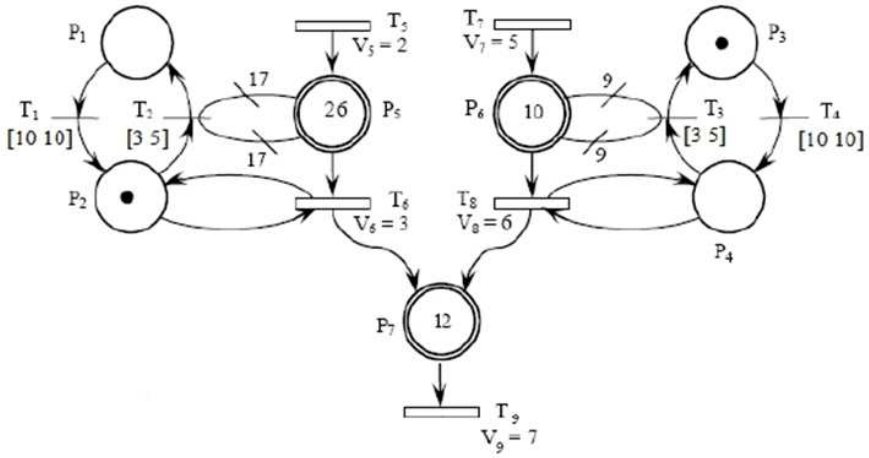
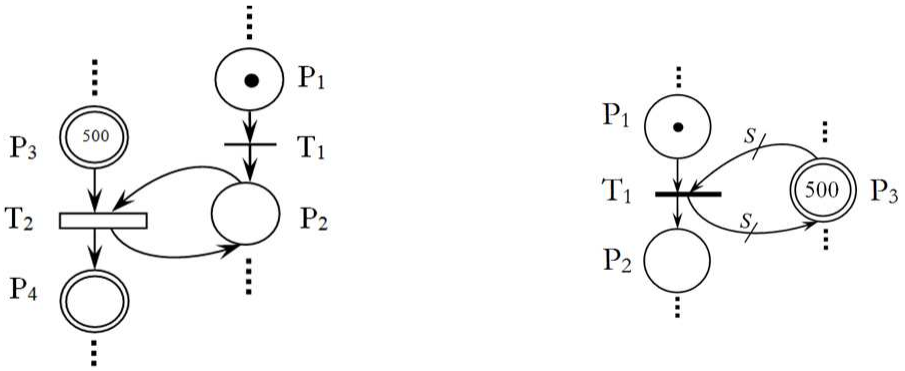


Figure 4.2. RdPH élémentaire modélisant le système des 3 réservoirs.

De manière générale dans un RdPH élémentaire, les parties discrètes et continues sont connectées à travers des boucles reliant les C-transitions à des D-places, le franchissement des C-transitions est alors conditionné par le marquage des D-places (Figure 4.3(a)). La partie continue

peut donc être contrôlée par la partie discrète. De manière symétrique le franchissement d'une D-transition peut aussi dépendre du marquage continu d'une C-place. Cela est représenté par une boucle reliant cette D-transition à cette C-place, c'est-à-dire que le franchissement de cette D-transition n'est possible que si le marquage de cette C-place est supérieur ou égal à ***S*** (Figure 4.3(b)).



(a) (b)

*Figure 4.3. (a) Validation de C-transition T_j si la D-place P_i est marquée. (b) Validation de D-transition T_j si la C-place atteint un seuil **S**.*

L'état d'un RdPH élémentaire, à un instant *t* est un couplage de l'état du RdP T-temporel et l'état du RdPCC. Les événements discrets dont l'occurrence donne un changement d'état sont :

1. Le marquage d'une C-place s'annule (C1-événement),
2. Le franchissement d'une D-transition (D1-événement) :
3. Le degré de validation d'une D-transition qui change à cause du marquage d'une C-place (D2-événement).

Par conséquent, la Définition 3.1 du chapitre 3 qui décrit le modèle $\boldsymbol{R_N}$ sera modifiée maintenant comme suit.

Définition 4.1. Un RdPH élémentaire en fonctionnement normal est une structure $\boldsymbol{R_N}$ avec une modification au niveau de l'assertion suivante:

- $\boldsymbol{T} = T^C \cup T^D$, T^C et T^D sont les ensembles des transitions continus et discrètes, sachant que :
  $T^D = T^{D1} \cup T^{D2}$, T^{D2} est l'ensemble des transitions discrètes ayant une C-place d'entrée (amont) ;
  $T^{D1} = T \setminus T^{D2}$;
  $T^C \cap T^{D1} \cap T^{D2} = \phi$.

❑

4.2.3. Principe de la méthode de translation

Avant de présenter le principe de notre méthode de translation, nous allons introduire le nouveau modèle d'AHL. Ce modèle est obtenu on ajoutant les spécifications présentées ci-dessous. Par la suite, nous détaillerons de manière intuitive la méthode de translation.

♦ l'activité d'une variable est décrite par sa première dérivée du type $\dot{x} = k$, où k est une constante donnée.
♦ L'espace d'état continu se compose de variables réelles qui modélisent le marquage de places continues du RdPH et les horloges qui correspondent aux transitions discrètes actives dans le RdPH.
♦ Les gardes de la relation de transition continue dans l'automate sont sous forme de prédicats linéaire sur l'ensemble des variables qui modélisent les marquages.
♦ L'espace initial est défini par la donnée d'un sommet initial et une région initiale réduite à un point.

Le nouveau modèle d'automate hybride linéaire est décrit formellement dans la définition suivante.

Définition 4.2. L'automate hybride linéaire associé au RdPH est un 6-uplet HA = (*Loc, x, E, δ, F, inv*) tel que :
- *Loc* est l'ensemble fini des sommets,
- *x* est l'espace d'état continu, tel que :

  ♦ x_C est le vecteur des variables de valeurs réelles qui modélisent le marquage des places continues ;
  ♦ x_D est le vecteur des horloges qui correspondent aux transitions actives. Une valuation est une fonction qui assigne une valeur $v(x) \in R$ à chaque variable $x_i \in x$.

- *E* est l'ensemble des événements ;
- δ est un ensemble fini des transitions, chaque transition est un quintuplet *T = (q, a, g, init, q')*, tel que:

  ♦ $q \in Loc$ est le sommet source ;
  ♦ $a \in E$ est l'événement associé au franchissement de *T* ;
  ♦ *g* est la garde de transition, c'est un prédicat linéaire sur *x*; une transition peut être franchie lorsque sa garde est satisfaite. ;
  ♦ *init* est une fonction de réinitialisation qui affecte une expression linéaire aux variables de *x* ;
  ♦ $q' \in Loc$ est le sommet cible ;

- *F* est la fonction qui assigne à chaque sommet un vecteur linéaire continu sur *x*. Les variables continues $m_i \in m_C$ évoluent en fonction d'une équation différentielle de la forme $\dot{m}_i = B_i$, où $B_i \in R$ est le bilan dynamique du marquage de la place continue P_i. Les horloges $\dot{t}_j \in x_D$ évoluent suivant l'équation différentielle $\dot{t}_j = 1$.
- *inv* est une fonction qui affecte à chaque sommet *q*, un prédicat linéaire *inv(q)* qui doit être satisfait par les variables continues afin de rester dans le sommet *q*.

❑

La partie discrète du RdPH élémentaire a un comportement indépendant de la partie continue, son évolution ne dépend que de son marquage initial et de la variable continue indépendante qui est le temps. Elle peut donc être étudiée seule, cela justifie la translation structurelle que nous proposons dans ce travail. Pour chaque marquage accessible par le RdP T-temporel, correspondra une configuration du RdPCC. Le modèle RdPH élémentaire peut donc être étudié d'une manière hiérarchique. La partie discrète est d'abord considérée, ensuite, pour chacun de ses marquages accessibles, la configuration continue est construite.

La procédure de translation proposée dans cette thèse comporte trois principales étapes :

Étape 1 : Isoler le RdP T-temporel du modèle hybride et construire l'automate initial à base du graphe de marquage accessible du RdP autonome sous-jacent au RdP T-temporel.

En supprimant les liens qui relient la partie discrète et la partie continue, ces deux parties deviennent momentanément indépendantes. Cette première étape de la procédure de translation consiste à construire le graphe de marquage accessible du RdP autonome sous-jacent. Il correspond à l'évolution autonome du RdP T-temporel. À partir de ce graphe, nous déduisons un automate temporisé en y associant les horloges correspondant aux transitions. L'automate résultat a un nombre fini de sommets car le RdP discret est borné. Nous allons par la suite détailler cette étape à travers des exemples intuitifs en tenant compte différents cas possibles.

Étape 2 : Associer les dynamiques aux sommets et les gardes aux transitions.

Nous désignons par sous-sommets, les sommets de l'automate temporisé qui résulte de l'étape précédente, chacun d'eux va correspondre à plusieurs sommets dans l'automate hybride final. Chaque sous-sommet correspond à un marquage du RdP T-temporel, et donc à une configuration du RdPCC, dépendant du marquage des D-places. On construit l'évolution du RdPCC indépendamment de celle du modèle discret. Ensuite, on ajoute l'influence ce de dernier. Celle-ci se traduira par le fait que certaines C-transitions peuvent être franchissables ou infranchissables selon l'état de la partie discrète. Une transition infranchissable sera éliminée de l'évolution du RdPCC. Enfin, nous associons les dynamiques à leurs sommets correspondants et les gardes à leurs transitions appropriées. Cette étape sera détaillée par la suite à travers un exemple intuitif.

Étape 3 : Construire l'automate hybride final

Après l'application de la deuxième étape principale de la translation, on obtient une forme hiérarchique d'un automate hybride comportant des sous-sommets. Chacun de ces sous-sommets comporte un automate hybride qui décrit la dynamique continue du sous sommet. Dans cette étape, nous remplaçons les transitions entre les sous-sommets par des transitions entre leurs sommets internes en adoptant la notion du macro-marquage. Cette étape sera détaillée aussi dans ce qui suit.

Nous allons détailler ci-dessous ces étapes principales. Comme dans un RdPH élémentaire il n'y a pas de transformation de marquage, du discret vers le continu ou du continu vers le discret.

On peut donc étudier les deux parties de manière indépendante. La première étape de translation du RdPH élémentaire en automate hybride proposée dans ce travail est détaillée dans la section qui suit.

Nous adopterons les notions suivantes :

$S_k = L_k$ un sommet.

L_{k1} et L_{k2} des sous-sommets.

m_0^* macro-marquage initial.

L_{f0} sommet initial de l'AHL final.

W matrice d'incidence du RdPCC.

V vecteur des vitesses maximales de franchissement.

$v(t)$ vecteur des vitesses de franchissement instantanées [David et. Alla, 92].

L_l et L_{l+1} sommets internes.

A. Influence de la partie continue sur la partie discrète

Comme il a été mentionné précédemment en section 4.2.2, une D-transition T_j peut avoir comme condition de franchissement le marquage d'une C-place P_i qui atteint un seuil $\boldsymbol{S}$. Graphiquement, ceci est représenté par une boucle (un arc de P_i vers T_j et un arc de T_j vers P_i) dont le poids est $\boldsymbol{S}$. Nous supposons dans cette partie que le modèle du système $\boldsymbol{R_N}$ (Définition 4.1) satisfait la relation suivante :

$$\forall P_i \in P^C,\ card\left(T^{D2} \cap P_i^{\circ}\right) \leq 1$$

Cela signifie que, quel que soit $P_i \in P^C$, il existe au plus une transition $T_j \in \left\{T^{D2} \cap P_i^{\circ}\right\}$ dans le modèle du système $\boldsymbol{R_N}$.

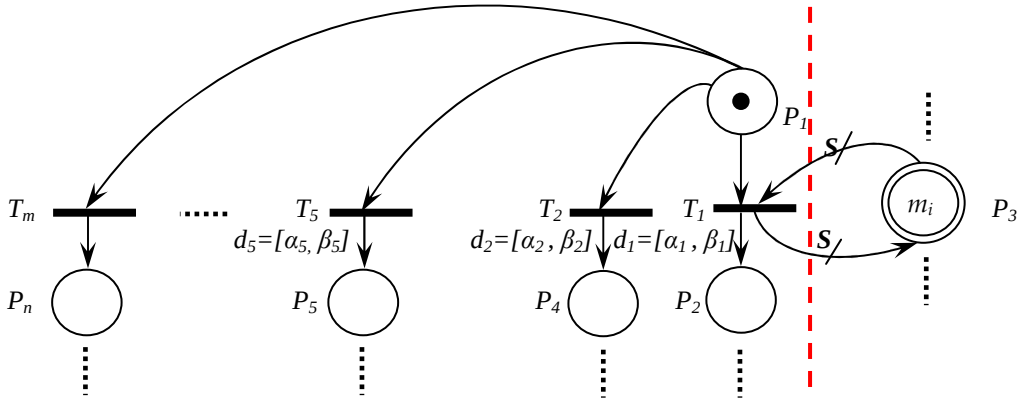
Généralement le marquage du RdP discret représente un sommet, ainsi que chaque D-transition représente une transition dans l'automate. Cela n'est pas toujours vrai dans un RdPH élémentaire où il existe deux types de transitions (simples appartiennent à T^{D1} et des transitions ayant une C-place d'entrée, appartiennent à T^{D2}). Les événements susceptibles de changer l'état d'un RdP T-temporel dans cette étape correspondent respectivement, aux transitions qui ne sont franchissables que si le marquage $m_i=\boldsymbol{S}$ ($T_j \in T^{D2}$) et aux franchissements de D-transitions simples ($T_j \in T^{D1}$).

La D-place source de la D-transition $T_j \in T^{D2}$ peut avoir plusieurs transitions de sortie (Figure 4.4(a)), et donc chaque Transition T_j correspond à un sommet ou deux sous-sommets tout dépend du type de T_j. Pour cela, nous avons proposé la procédure de translation suivante :

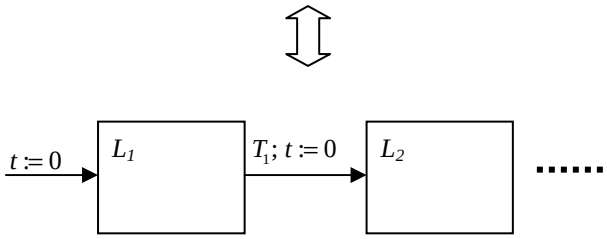
1- isoler le RdP T-temporel ;
2- Construire le graphe de marquage accessible du RdP autonome sous-jacent ;
3- éclater chaque sommet qui correspond à $T_j \in T^{D2}$ en deux sous-sommets L_{k1} et L_{k2}, un qui représente la validation du T_j et l'autre le franchissement du T_j ;

4- associer l'horloge du temps au sommet L_{k2} avec initialisation à son entrée ;
5- relier L_{k1} et L_{k2} pour chaque sommet avec une transition et lui associer la garde m_i=***S*** ;
6- créer une transition de sortie pour chaque L_{k2} et lui associer la garde $d_j=[\alpha_j , \beta_j]$;
7- relier la transition de sortie de chaque L_{k2} avec le sommet qui suit ;
8- créer pour chaque transition de sortie $T_j \in T^{D1}$ un sommet destination ;
9- relier chaque sommet créé par deux transitions de L_{k1} et L_{k2}, et lui associer la garde $d_j=[\alpha_j , \beta_j]$.

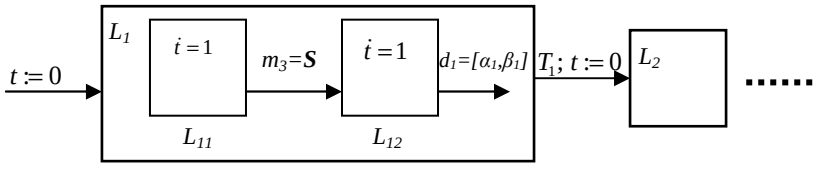
Considérons l'exemple de la Figure 4.4(a), où $T_1 \in T^{D2}$ et P_1 a plusieurs transitions de sortie (T_2 ,T_5,.... T_m), l'automate obtenu est présenté sur la Figure 4.4(d).



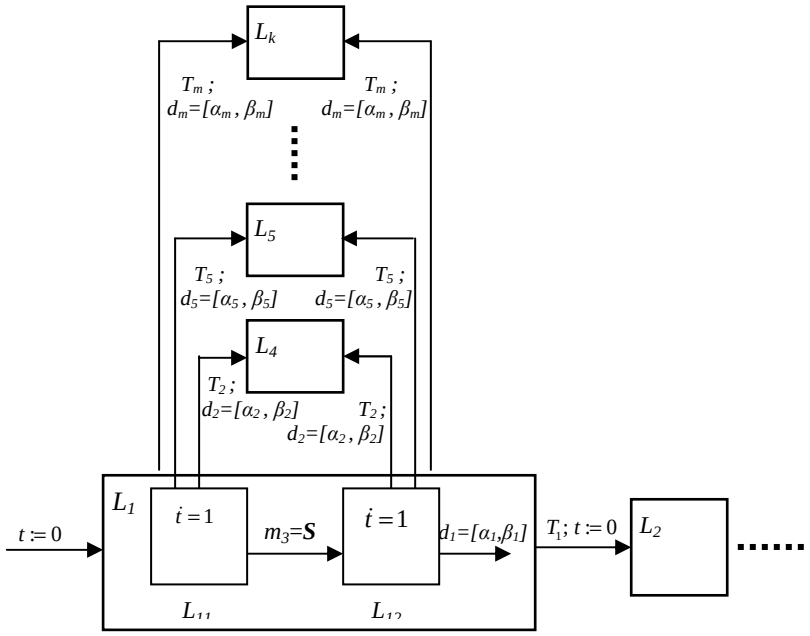
(a)



(b)



(c)



(d)

Figure 4.4. (a) RdPH élémentaire (b) graphe de marquage accessible du RdP autonome sous-jacent (c) et (d) étapes de translation du cas général.

Durant notre travail, dans cette partie de translation, nous avons rencontré plusieurs configurations particulières pour lesquelles nous avons proposé une translation spécifique.

A.1. Configuration 1

La D-place P_i source de la transition $T_j \in T^{D2}$ n'a aucune transition de sortie $T_j \in T^{D1}$, et la durée de la transition $T_j \in T^{D2}$ est $d=[\alpha, \beta]$, La procédure de translation dans ce cas est faite de la manière suivante :

1- isoler le RdP T-temporel ;
2- construire le graphe de marquages accessibles du RdP autonome sous-jacent ;
3- éclater le sommet source de la T_j en deux sous-sommets L_{k1} et L_{k2}, un qui représente la validation du T_j et l'autre le franchissement du T_j ;
4- associer l'horloge au sommet L_{k2} avec initialisation à son entrées ;
5- relier L_{k1} et L_{k2} pour chaque sommet avec une transition et lui associer la garde m_i=**S** ;
6- créer une transition de sortie pour chaque L_{k2} et lui associer la garde $d=[\alpha, \beta]$;
7- relier la transition de sortie de chaque L_{k2} avec le sommet qui suit ;

Considérons l'exemple de la Figure 4.5(a), où $T_1 \in T^{D2}$, l'automate obtenu est présenté sur la Figure 4.5(d).

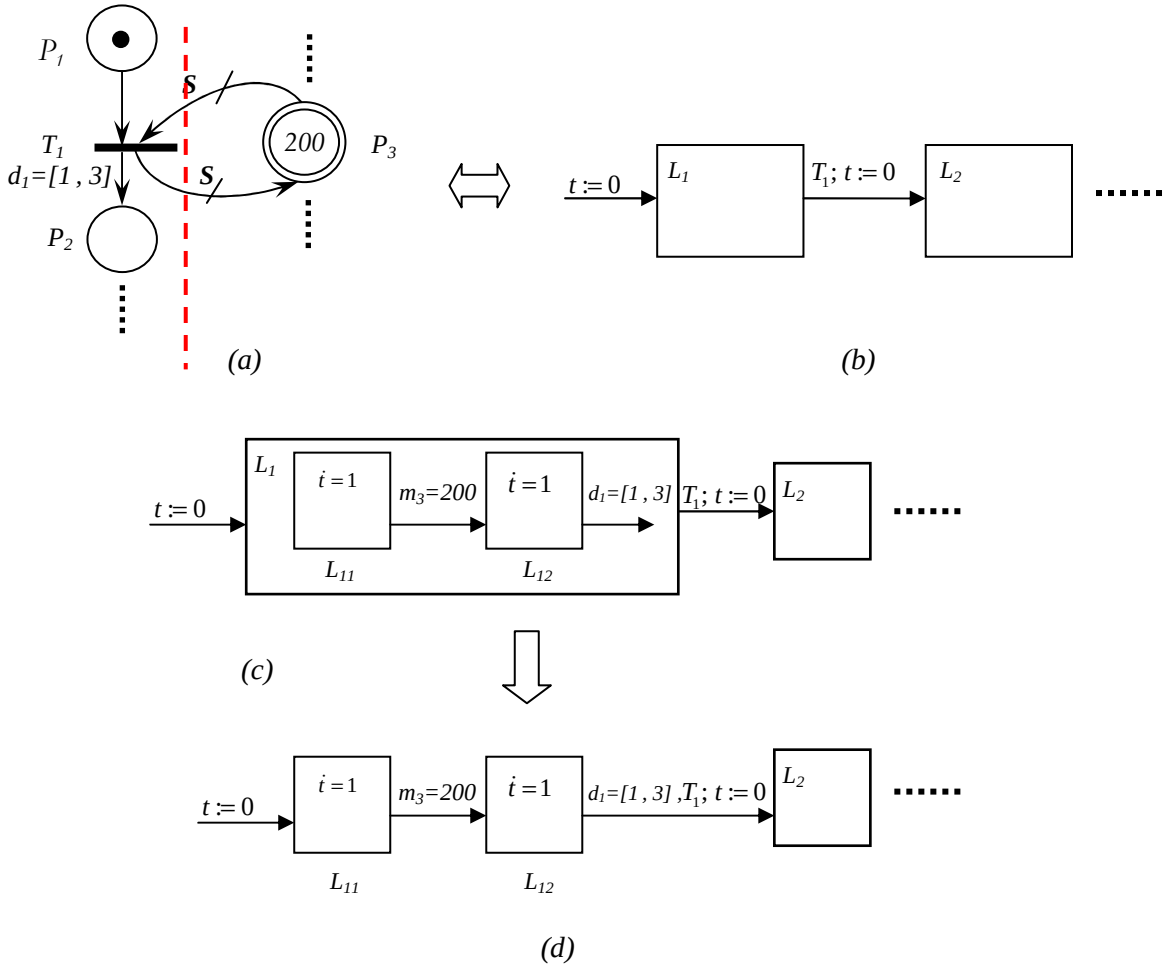


Figure 4.5. (a) RdPH élémentaire (b) graphe de marquage accessible du RdP autonome sous-jacent (c) et (d) étapes de translation pour la configuration 1.

A.2. Configuration 2

La D-place P_i source de la transition $T_j \in T^{D2}$ n'a aucune transition de sortie $T_j \in T^{D1}$, mais la durée de la transition $T_j \in T^{D2}$ est $d=[0\ ,\ 0]$, alors la procédure de translation dans ce cas est faite de la manière suivante :

1- isoler le RdP T-temporel ;
2- construire le graphe de marquages accessibles du RdP autonome sous-jacent ;
3- éclater le sommet source de la T_j en deux sous-sommets L_{k1} et L_{k2}, un qui représente la validation du T_j et l'autre le franchissement du T_j ;
4- associer l'horloge au sommet L_{k2} avec initialisation à son entrée ;
5- relier L_{k1} et L_{k2} pour chaque sommet avec une transition et lui associer la garde $m_i=\boldsymbol{S}$;
6- créer une transition de sortie pour chaque L_{k2} et lui associer la garde $d_j=[\alpha_j\ ,\ \beta_j]$;
7- éliminer le sommet L_{k2} associé à la garde $d_j=[0\ ,\ 0]$ pour chaque sommet éclaté ;
8- relier chaque sommet L_{k1} au sommet qui suit, en gardant la garde $m_i=\boldsymbol{S}$

Considérons à nouveau l'exemple de la Figure 4.5(a), avec $d_1=[0\ ,\ 0]$, l'automate obtenu est présenté sur la Figure 4.6.

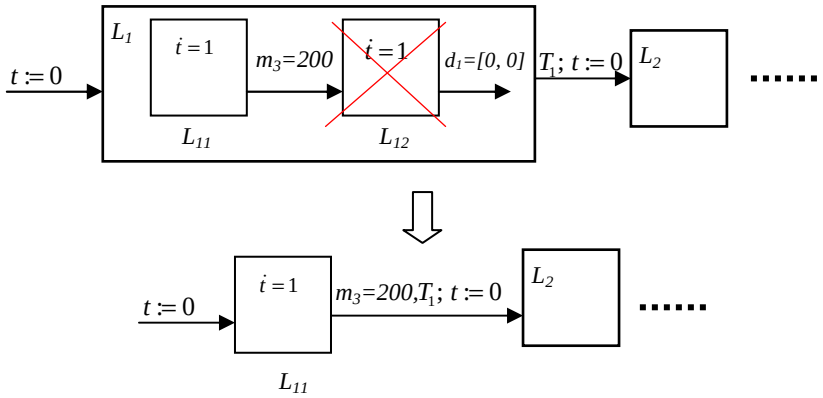


Figure 4.6. Étapes de translation en appliquant la procédure pour la configuration 2.

A.3. Configuration 3

La D-place P_i a plusieurs transitions T_j de durées $d_j=[\alpha_j\ ,\ \beta_j]$, et en même temps ∃ au plus $T_j \in T^{D2} \cap P_i^{\cdot}$ avec une durée $d=[0\ ,\ 0]$, alors la procédure de translation dans ce cas est faite de la manière suivante :

1- isoler le RdP T-temporel ;
2- Construire le graphe de marquages accessibles du RdP autonome sous-jacent ;
3- éclater le sommet source de la T_j en deux sous-sommets L_{k1} et L_{k2}, un qui représente la validation du T_j et l'autre le franchissement du T_j ;
4- associer l'horloge au sommet L_{k2} avec initialisation à son entrée ;
5- relier L_{k1} et L_{k2} pour chaque sommet avec une transition et lui associer la garde m_i=$\boldsymbol{S}$;
6- créer une transition de sortie pour chaque L_{k2} et lui associer la garde $d_j=[\alpha_j\ ,\ \beta_j]$;
7- éliminer le sommet L_{k2} associé à la garde $d_j=[0\ ,\ 0]$ pour chaque sommet éclaté ;
8- relier chaque sommet L_{k1} au sommet qui suit, en gardant la garde m_i=$\boldsymbol{S}$;
9- créer pour chaque transition de sortie $T_j \in T^{D1}$ un sommet destination ;
10- relier chaque sommet destination créé au sommet L_{k1}, par une transition et lui associer la garde $d_j=[\alpha_j\ ,\ \beta_j]$.

Considérons à nouveau l'exemple de la Figure 4.5(a), mais avec les spécifications présentées sur la Figure 4.7(a), l'automate obtenu est présenté sur la Figure 4.7(c)

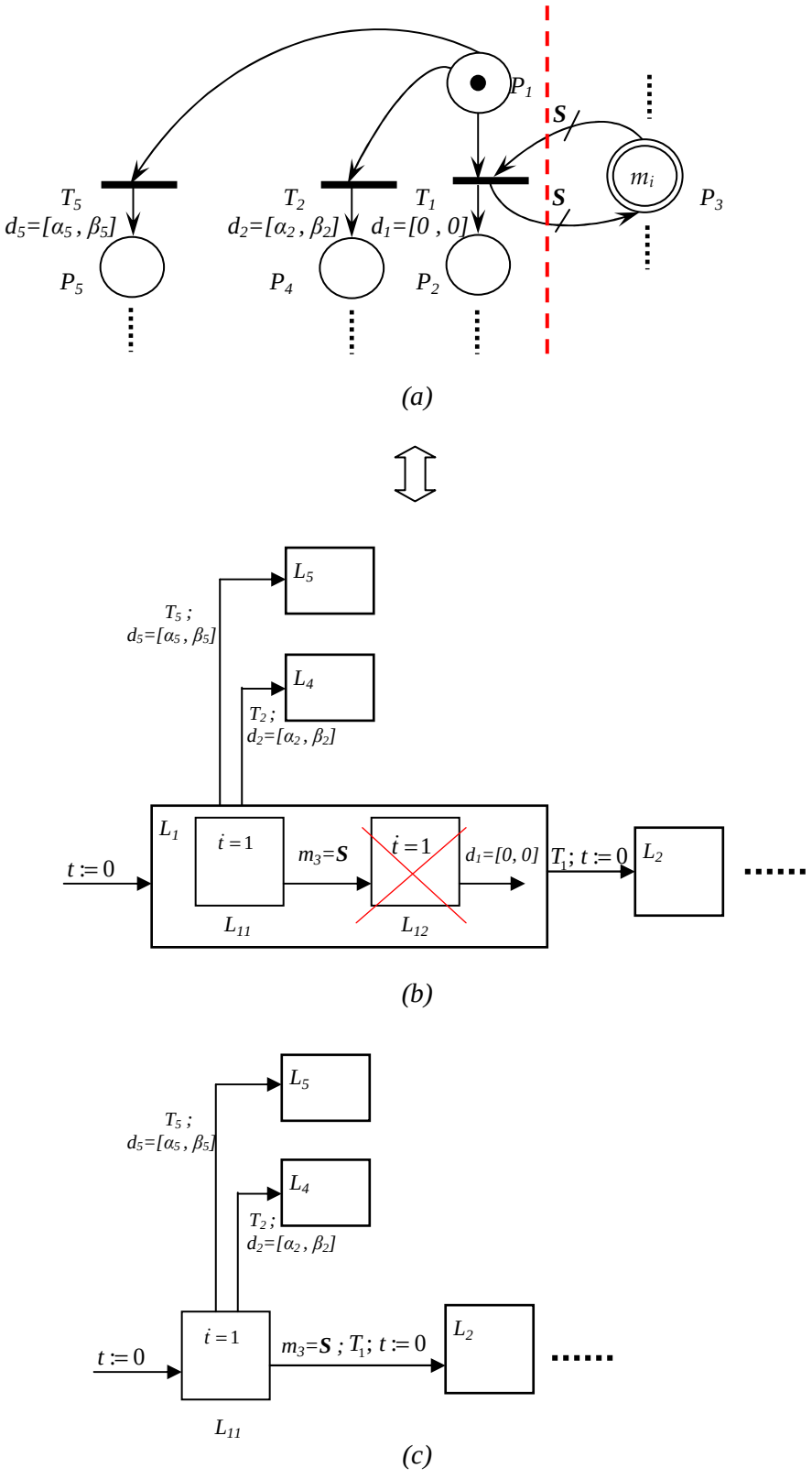


Figure 4.7. (a) RdPH élémentaire (b) et (c) étapes de translation en appliquant la procédure pour la configuration 3.

Nous allons maintenant détailler l'étape principale 2 dans la section suivante.

B. Influence de la partie discrète sur la partie continue

Comme il a été mentionné précédemment, c'est le RdP T-temporel qui contrôle le comportement du RdPCC via des boucles connectant certaines D-places à certaines C-transitions (Figure 4.8). Cela signifie que ces dernières ne sont pas validées et par conséquent ne peuvent être franchies que si les D-places sont marquées.

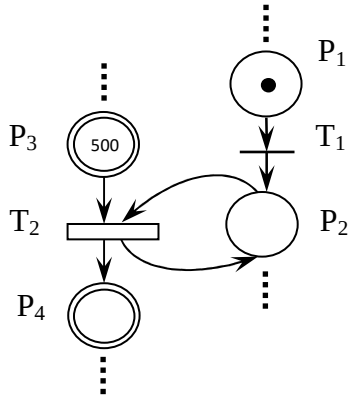


Figure 4.8. Validation de T_j si P_i est validée dans un RdPH élémentaire.

Pour pouvoir avoir une construction structurelle, il faut que celle-ci soit indépendante du marquage initial de la partie continue du RdPH. Il faut donc être capable de construire l'automate hybride correspondant au RdPCC pour tout marquage de la partie continue. Pour cette raison, nous allons déterminer une configuration structurelle du RdPCC, et ceci en supposant que le marquage initial où toutes les places continues sont marquées. Cela correspond à un macro-marquage dont le support est tout P^C. La notion de macro-marquage, a été introduite par David et Alla [David et. Alla, 05] dans le but de représenter d'une manière finie le nombre de marquages infini accessibles par un RdP continu autonome.

Le but de cette étape est de compléter l'automate hybride modélisant les comportements des RdPCC correspondant à chaque sommet du RdP T-temporel et donc à une configuration du RdPCC. L'évolution d'un RdPCC est généralement modélisée par son graphe d'évolution. Le graphe d'évolution d'un RdPCC est assimilable à un automate hybride linéaire, dans lequel les variables d'état correspondent au marquage des C-places et les transitions correspondent aux événements. Pour un RdP continu, autonome ou non, à n places, le nombre maximum de macro-marquages accessibles est 2^n, ceci est dû à la nature booléenne de l'état d'une place dans un macro-marquage.

Remarque 4.1 : Pour un RdPCC, le nombre de macro-marquages atteignables est maximal pour le macro-marquage initial qui a pour support P^C (l'ensemble de toutes les C-places).

❑

Sachant que le seul événement pouvant changer l'état d'un RdPCC est que le marquage d'une C-place s'annule (C1-événement), l'occurrence de cet événement n'est possible que pour les C-place marquées et dont le marquage est décroissant. Un macro-marquage m^* dans lequel la C-

place P_i n'est pas marquée, est atteignable depuis le macro-marquage m_0^*, dans lequel P_i est marquée. Le contraire n'est pas possible.

Pour cette raison et pour prendre en compte tous les marquages accessibles par les RdPCC, nous allons considérer que le macro-marquage initial a pour support P^C et construire l'automate hybride modélisant le comportement de chaque RdPCC sans considération pour son marquage initial. Pour ce faire, nous procédons comme suit :

1- Considérer que le macro-marquage initial a pour support P^C ,

$$m_0^* = P^C \Rightarrow \begin{cases} \text{- tous places } P_i \text{ initialement marquées} \\ \text{- tous les transitions sont franchies à leurs vitesses maximales} \end{cases}$$

2- Créer le sommet initial L_{f0} de l'automate hybride final et lui associer l'activité

$$\dot{M} = W.V$$

3- Créer depuis le sommet initial une transition pour chaque place P_i dont le marquage est initialement décroissant, et lui affecter la garde $m_i = 0$.

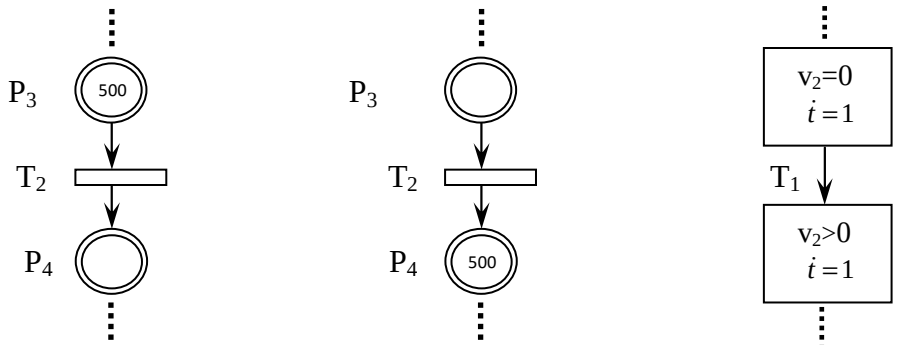
4- Pour chaque transition construite précédemment, créer un sommet destination et lui affecter l'activité :

$$\dot{M} = W.v(t)$$

5- Fusionner les sommets ayant la même activité.

6- Vérifier si pour chaque place P_i dont le marquage est initialement décroissant, le marquage a atteint un bilan nul ($\dot{m}_i = 0$). Si oui arrêter l'étape, sinon refaire l'étape 3 pour chaque sommet créé en étape 4.

La Figure 4.9 représente un exemple intuitif de la configuration du RdPCC pour chaque marquage du RdP T-temporel du RdPH élémentaire de la Figure 4.8.



(a) *(b)* *(c)*

Figure 4.9. (a) RdP continu C correspondant au marquage discret […1 0…]^T
(b) RdP continu C correspondant au marquage discret […0 1…]^T.
(c) L'automate correspondant.

La dernière étape consiste à remplacer les transitions entre les sous-sommets par des transitions entre les sommets internes. Les trois événements susceptibles de changer l'état d'un RdPH élémentaire sont : le franchissement d'une D-transition, le marquage d'une C-place qui s'annule ou le marquage d'une C-place qui atteint le seuil ***S***. Ils correspondent respectivement, dans la forme hiérarchique de l'automate hybride que nous avons obtenu dans cette étape, aux transitions entre les sous-sommets, aux transitions entre les sommets internes et sous-sommets et aux transitions ayant des gardes de types m_i=***S***.

Pour détailler cette construction, considérons la notion de macro-marquages, un macro-marquage d'un RdPCC de *n* places, peut être caractérisé par un vecteur booléen $\bar{M}$ de dimension *n*, avec :

$$\bar{M}_l = \begin{cases} 1 \text{ si } P_i \text{ est marquée} \\ 0 \text{ sinon} \end{cases}$$

Cette étape est garantie à condition que le vecteur caractéristique du sommet source soit inférieur ou égal (composante à composante) au vecteur caractéristique du sommet cible. Cela signifie que le franchissement d'une transition discrète dans un RdPH élémentaire ne peut pas marquer une place continue vide (il n'y a pas de transformation de marques). Nous pouvons alors fixer les deux étapes suivantes.

1- Calculer le vecteur booléen $\bar{M}$ qui caractérise le macro-marquage pour chaque sommet interne construit
2- Remplacer les transitions entre toute paire de sommets par une transition qui relie leurs sommets internes quand la condition suivante est satisfaite : le vecteur caractéristique du sommet source est Supérieur ou égal (composante à composante) au vecteur caractéristique de son sommet but :

$$\bar{M}_l \geq \bar{M}_{l+1}$$

Nous pouvons résumer toute les étapes détaillées de la procédure de translation en tenant compte tous les cas possibles du RdPH élémentaire sur un algorithme informel et systématique, ce dernier est décrit dans la section suivante.

4.2.4. L'algorithme de translation

Dans cette section, tous les résultats des étapes principales ont été organisées dans l'algorithme suivant.

Algorithme 4.1 : Algorithme de translation

1: «Initialisation» ;
2: isoler la partie du RdP T-temporel du modèle hybride de la partie continue (RdPCC);
3: construire le graphe de marquage accessible du RdP autonome sous-jacent ;
4: **tant que** $\forall P_i \in P^C,\ card\left(T^{D2} \cap P_i^{\circ}\right) \leq 1$, **faire**

5: **si** la place $P_i \in P^D$ source de la transition $T_j \in T^{D2}$ **n'a aucune** transition de sortie $T_j \in T^{D1}$, **faire**
6: éclater chaque sommet qui correspond à $T_j \in T^{D2}$ en deux sous-sommets L_{k1} et L_{k2};
7: associer l'horloge du temps au sommet L_{k2} avec initialisation à son entrée ;
8: relier L_{k1} et L_{k2} pour chaque sommet avec une transition et lui associer la garde $m_i = \boldsymbol{S}$
9: créer une transition de sortie pour chaque L_{k2} et lui associer la garde $d_j=[\alpha_j , \beta_j]$;
10: **si** $d_j=[0 , 0]$ correspond à $T_j \in T^{D2}$ **faire**
11: éliminer le sommet L_{k2} pour chaque sommet éclaté ;
12: relier chaque sommet L_{k1} au sommet qui suit, en gardant la garde $m_i = \boldsymbol{S}$;
13: **sinon**
14: relier la transition de sortie de chaque L_{k2} avec le sommet ou le sous-sommet qui suit
15: **fin si**
16: **sinon**
17: éclater chaque sommet qui correspond à $T_j \in T^{D2}$ en deux sous-sommets L_{k1} et L_{k2};
18: associer l'horloge du temps au sommet L_{k2} avec initialisation à son entrée ;
19: relier L_{k1} et L_{k2} pour chaque sommet avec une transition et lui associer la garde $m_i = \boldsymbol{S}$;
20: créer une transition de sortie pour chaque L_{k2} et lui associer la garde $d_j=[\alpha_j , \beta_j]$;
21: **si** $d_j=[0 , 0]$ correspond à $T_j \in T^{D2}$ **faire**
22: éliminer le sommet L_{k2} pour chaque sommet éclaté ;
23: relier chaque sommet L_{k1} au sommet qui suit, en gardant la garde $m_i = \boldsymbol{S}$;
24: créer pour chaque transition de sortie $T_j \in T^{D1}$ un sommet destination ;
25: relier chaque sommet destination créé au sommet S_{k1}, par une transition et lui associer la garde $d_j=[\alpha_j , \beta_j]$;
26: **sinon**
27: relier la transition de sortie de chaque L_{k2} avec le sommet qui suit;
28: créer pour chaque transition de sortie $T_j \in T^{D1}$ un sommet destination ;
29: relier chaque sommet destination créé par deux transitions de L_{k1} et L_{k2} et lui associer la garde $d_j=[\alpha_j , \beta_j]$;
30: **fin si.**
31: **fin si.**
32: **fin tant que.**
33: «configuration_RdPCC», configurer la partie continue du RdPCC pour chaque marquage discret ;
34: mettre $m_0^* = P^C$;
35: créer L_{f0}, pour chaque sommet et lui associer l'activité $\dot{M} = W.V$;
36: « vérification_marquage » ;
37: **Si** le marquage pour chaque C-place P_i est décroissant **Alors**
38: créer depuis L_{f0}, une transition pour chaque place P_i et lui affecter la garde $m_i = 0$;
39: pour chaque transition construite, créer un sommet destination et lui affecter l'activité $\dot{M} = W.v(t)$;
40: fusionner les mêmes sommets ayant la même activité ;
41: **Si** $\dot{m}_l = 0$, pour chaque P_i dont le marquage est initialement décroissant **Alors**

42: aller « comparaison_vecteurs » ;
43: **sinon**
44: retourner « vérification_marquage » ;
45: **Fin Si**
46: **Fin Si**
47: « comparaison_vecteurs », remplacer les transitions en comparant les vecteurs booléens ;
48: calculer le vecteur booléen $\bar{M}$ pour chaque sommet interne ;
49: **Si** $\bar{M}_l \geq \bar{M}_{l+1}$ **Alors**
50: remplacer chaque transition entre sommets par une transition entre leurs sommets internes L_l et L_{l+1}.
51: **fin Si**
52: **Fin**

L'algorithme de translation des RdP élémentaires en automates hybrides donne lieu à un automate hybride linéaire avec un certain nombre de spécificités.

1. Les variables d'état continues sont de deux types :

- Les horloges t_i : elles correspondent aux durées de validation des D-transitions dans le modèle RdPH élémentaire;
- Les variables m_i : elles correspondent aux marquages des C-places dans le modèle RdPH élémentaire ;

Ainsi x le vecteur d'état est de la forme $x = \begin{pmatrix} x_C \\ x_D \end{pmatrix}$, tel que: $x_C = (m_1, m_2, ..., m_{nC})^T$ est un vecteur de nC variables réelles et $x_D = \{t_1, t_2, ..., t_j\}$ est le vecteur d'horloges.

2. Les horloges peuvent être réinitialisées lors de franchissement de transitions, tandis que les variables réelles m_i ne le sont jamais, elles n'effectuent donc jamais de sauts.

3. Une garde de chaque transition ne dépend que d'une seule variable réelle. Elle peut être sous une des deux formes suivantes *:*

- $m_i = 0$, si la transition modélise l'annulation du marquage d'une C-place dans le RdPH élémentaire ;
- $m_i = \boldsymbol{S}$, si la transition modélise le marquage d'une C-place qui atteint le seuil $\boldsymbol{S}$.
- $\alpha_j \leq t_j \leq \beta_j$, si la transition modélise une D-transition dans le RdPH élémentaire.

Remarque 4.2.

- Ce travail peut être étendue en RdPH général, mais l'algorithme de translation sera difficile, surtout le découplage continu-discret sera perdu et donc le caractère structurel de la translation. Cependant, ce problème pourrait être étudié comme une perspective d'avenir.

- Lorsqu'on choisit un marquage initial et que l'on utilise PHAVer, la terminaison n'est pas garantie comme dans le cas d'un AHL

Ces remarques sont importantes, elles constituent les éléments de départ pour la synthèse du diagnostiqueur.

4.2.5. Analyse de l'algorithme de translation

L'algorithme converge pour les RdPH élémentaires bornés. L'algorithme se termine lorsqu'il n'y a aucune nouvelle visite d'un sommet à comparer. La visite d'un sommet est complètement caractérisée par l'IB-état du RdPH élémentaire. Ainsi l'algorithme converge si le nombre de sommets et le nombre de fois que chaque sommet est visité est fini.

Le RdPH élémentaire est constitué d'un RdP T-temporel et d'un RdPCC. Plusieurs algorithmes ont été proposés pour la translation des RdP T-Temporels, la convergence est conditionnée par le caractère borné du RdP T-temporel. Tandis que la translation du RdPCC en automate hybride converge toujours, même si le RdPCC n'est pas bornée. Ceci est dû au fait que l'on caractérise une place non pas par son marquage mais par son macro-marquage qui est une grandeur booléenne. Ainsi l'AHL associé au RdPH élémentaire dans la classe considérée a un nombre fini de sommets, alors le nombre de transitions est aussi fini.

La pluparts des travaux récents dans la littérature n'ont pas fournis des preuves pour leurs techniques de translation des RdP en automates. Pour cela, nous allons introduire une preuve mathématique pour la procédure de translation proposée dans cette thèse. Dans un premier temps, nous définissons les deux modèles (RdPH élémentaire et AHL) comme Systèmes de Transitions Temporisés (STT) en basant sur leurs sémantiques d'évolution. Par la suite, nous présenterons une preuve pour la similarité temporelle de leurs STT. La section suivante présente les notions nécessaires pour cette preuve.

4.3. Les systèmes de transitions temporisés

Un système de transitions temporisé (TTS[1]) permet de décrire l'ensemble des états ainsi que des transitions entre ces états [Larsen *et al.*, 95]. Deux types de transitions sont possibles dans un TTS : une transition continue ou de temps qui décrit un écoulement du temps ou une évolution continue, et une transition discrète décrivant l'évolution suite à une action discrète (événement). Si l'on isole un état initial dans un TTS, les différents chemins partant de cet état représentent les différentes évolutions possibles pour le système modélisé. Dans ce travail, nous avons proposé que le marquage d'une C-place qui s'annule (C1-événement) est considéré comme une transition discrète.

[1] Pour Time Transition System TTS, en anglais.

Définition 4.3. (Système de Transitions Temporisé-STT).
Un système de transitions temporisé (STT) sur un ensemble d'actions Ψ (ou alphabet) est un quadruplet $S_{TT} = (Q, q_0, \Psi, \rightarrow)$ avec:

- Q est l'ensemble des états;
- q_0 est l'état initial;
- Ψ est l'ensemble des actions (ou alphabet);
- $\longrightarrow \subseteq Q \times (\Psi \cup R^+) \times Q$ est une relation de transition.

Une relation $(q, e, q') \in Q$ est notée également $q \xrightarrow{e} q'$ [2]. La relation de transition se décompose en une relation de transition continue $\xrightarrow{d \in R^+}$ et une relation de transition discrète $\xrightarrow{a \in \Psi}$.

Un STT admet les propriétés suivantes :

- Propriété du zéro-délai : si $q \xrightarrow{0} q'$ alors $q = q'$.
- Propriété d'additivité temporelle: si $q \xrightarrow{\delta} q'$ et $q' \xrightarrow{\delta'} q''$ alors $q \xrightarrow{\delta+\delta'} q''$ avec $\delta, \delta' \in R^+$
- Propriété de continuité temporelle : si $q \xrightarrow{\delta} q'$ alors $\forall \delta, \delta' \in R^+$, tel que $\delta = \delta' + \delta''$, $\exists q'' \in Q$ tel que $q \xrightarrow{\delta'} q''$ et $q'' \xrightarrow{\delta''} q'$.
- Propriété du déterminisme temporel : si $q \xrightarrow{\delta} q'$ et $q \xrightarrow{\delta} q''$ alors $q' = q''$.

❑

Remarque 4.3. Dans certains travaux, on retrouve la notion de l'action vide (slutter event, en anglais). Cette notion est utilisée surtout lors du produit synchrone entre plusieurs TTS. Nous ne présentons par cette notion dans le cadre de notre travail.

Nous présentons dans la Définition 4.4 la notion d'exécution (run).

Définition 4.4. (Exécution dans un STT - Run)

Une exécution η d'un STT $S_{TT} = (Q, q_0, \Psi, \rightarrow)$ est une séquence finie ou infinie de transitions continues et discrètes de S_{TT}.

$$\eta = q_0 \xrightarrow{e_0} q_1 \xrightarrow{e_1} q_2 \xrightarrow{e_2} q_3 \ldots . q_{n-1} \xrightarrow{e_{n-1}} q_n \ldots .$$

Avec $e_i \in \Psi \cup R^+, \forall i$. L'ensemble des exécutions d'un système de transitions temporisé S_{TT} est noté $[\![S_{TT}]\!]$.

Une exécution η est une **alternance** de transitions continues et discrètes si elle peut s'écrire sous la forme :

$$\eta = q_0 \xrightarrow{d_0} q'_0 \xrightarrow{a_0} q_1 \xrightarrow{d_1} q'_1 \xrightarrow{a_1} q_2 \ldots . \xrightarrow{d_{n-1}} q'_{n-1} \xrightarrow{a_{n-1}} q_n \ldots .$$

[2] Dans un contexte où plusieurs STT sont définies, une transition (q, e, q') d'un STT S_{TT} est notée également $q \xrightarrow{e}_{S_{TT}} q'$

Avec $a_i \in \Psi$, $d_i \in R^+, \forall i$.

Dans ce cas, une notation plus concise serait la suivante :

$$\eta = q_0 \xrightarrow{(d_0,a_0)} q_1 \xrightarrow{(d_1,a_1)} q_2 .... \xrightarrow{(d_{n-1},a_{n-1})} q_n ....$$

❑

Dans la Définition 4.4, d_i correspond à la durée entre les états q_i et q_i'. Ainsi, les informations temporelles sont retenues d'une manière relative dans les exécutions. Dans la Définition 4.5, nous présentons la notion de trace d'exécution où le temps sera considéré d'une manière absolue.

Définition 4.5. (Trace d'exécution - Trace)

La trace d'une exécution $\eta = q_0 \xrightarrow{d_0} q_0' \xrightarrow{a_0} q_1 \xrightarrow{d_1} q_1' \xrightarrow{a_1} q_2 ... \xrightarrow{d_{n-1}} q_{n-1}' \xrightarrow{a_{n-1}} q_n ...$ est le mot temporisé :

$$trace(\alpha) = (a_0, \gamma_0)(a_1, \gamma_1)....(a_n, \gamma_n)....$$

$$\text{avec } \begin{cases} \gamma_0 = d_0, \\ \gamma_i = \sum_{j=0}^{i} d_i. \end{cases}$$

❑

Un système de transitions temporisé est une manière mathématique pour illustrer l'évolution du comportement système par rapport au temps. Une exécution sur un STT correspond à une trajectoire possible dans le STT. La notion de trace d'exécution peut être utilisée dans le contexte des langages temporisés décrit dans la section suivante. La notion de langage temporisé constitue une deuxième alternative pour décrire le comportement d'un système, et ce, à travers la définition de l'ensemble des mots temporisés qu'il peut générer.

4.3.1. Notion de langages temporisés

Soit Ψ un ensemble dénotant un alphabet. Ψ^* représente l'ensemble des suites (mots) finies d'éléments de Ψ [3]. Ψ^ω est l'ensemble des suites infinies d'éléments de Ψ . $\Psi^\infty = \Psi^* \cup \Psi^\omega$.

Définition 4.6. (Mot temporisé) Un mot temporisé ω sur un alphabet Ψ est une séquence finie ou infinie :

$$\omega = (a_0, d_0)(a_1, d_1)...(a_n, d_n)...$$

telle que $\forall i \geq 0, a_i \in \Psi$, $d_i \in R^+, d_{i+1} \geq d_i$, représente l'ensemble des tous les mots finis et infinis construits sur Ψ .

❑

La valeur d_i d'un mot temporisé $\omega = (a_0, d_0)(a_1, d_1)...(a_n, d_n)...$ donne la date absolue de l'événement a_i , et ce, en considérant que l'instant initial est à la date zéro.

[3] $\Psi^* = \bigcup_{i=0}^{\infty} \Psi^i$, avec Ψ^i correspond aux mots de longueur *i* des symboles de Ψ .

Nous notons par $Untimed(\omega) = a_0 a_1 ... a_n ...$, la partie non temporisée de ω. La durée de ω est notée par $Duration(\omega) = sup_i d_i$

Définition 4.7. $TW^*(\Psi)$ dénote l'ensemble des mots temporisés finis. $TW^{\omega}(\Psi)$ dénote l'ensemble des mots temporisés infinis $TW^{\infty}(\Psi) = TW^*(\Psi) \cup TW^{\omega}(\Psi)$. Un langage temporisé Lan sur Ψ est un sous-ensemble de $TW^{\infty}(\Psi)$.

❑

Lors de l'analyse des systèmes, on est souvent amené à comparer le comportement de deux systèmes modélisés chacun par un STT. La notion de bisimilarité temporelle présentée dans la section suivante répond à cette question.

4.3.2. Similarité et bisimilarité temporelle

Une relation de bisimulation permet de définir une équivalence de comportement entre deux systèmes de transitions temporisés. Une bisimilarité entre deux systèmes de transitions assure que toute action de l'un des systèmes peut être simulée par l'autre dans le sens qu'elle a un comportement équivalent dans l'autre système. Nous commençons par définir la notion de similarité temporelle.

Définition 4.8. (Relation de similarité temporelle) Soient deux systèmes de transitions temporisés $S_{TT1} = (Q_1, q_0^1, \Psi, \longrightarrow_1)$ et $S_{TT2} = (Q_2, q_0^2, \Psi, \longrightarrow_2)$. Soit Φ une relation binaire sur $Q_1 \times Q_2$. Nous écrivons $s \; \Phi \; s'$ pour $(s, s') \in \Phi$. Φ est une relation temporelle forte de S_{TT1} par S_{TT2} si les assertions suivantes sont vérifiées :

$$\begin{cases} \text{si } s_1 \in q_0^1, \text{ alors } \exists s_2 \in q_0^2 \text{ tel que } s_1 \; \Phi \; s_2 ; \\ \text{si } s_1 \xrightarrow{\delta}_1 s_1', \text{ avec } \delta \in R^+ \text{ et } s_1 \; \Phi \; s_2 \text{ alors } \exists s_2 \xrightarrow{\delta}_2 s_2' \text{ tel que } s_1' \; \Phi \; s_2' ; \\ \text{si } s_1 \xrightarrow{a}_1 s_1' \text{ avec } a \in \Psi \text{ et } s_1 \; \Phi \; s_2 \text{ alors } \exists s_2 \xrightarrow{a}_2 s_2' \text{ tel que } s_1' \; \Phi \; s_2' ; \end{cases}$$

❑

Définition 4.9. (Relation de bisimilarité temporelle) Soit S_{TT1} et S_{TT2} deux systèmes de transitions temporisés. S_{TT1} et S_{TT2} sont en relation de bisimilarité temporelle s'il existe une relation de similarité forte Φ (au sens de la Définition 4.8) de S_{TT1} par S_{TT2} et si Φ^{-1} est aussi une relation de similarité forte de S_{TT1} par S_{TT2}. La relation de bisimilarité est notée $S_{TT1} \approx S_{TT2}$.

❑

D'une manière générale, les STT sont utilisés pour donner la sémantique et la description d'un modèle. Cependant, ils ne sont pas efficaces pour manipuler l'ensemble des comportements des systèmes. Ainsi, ils ne sont pas utilisables directement pour l'analyse et la surveillance ainsi que la modélisation. Les réseaux de Pétri et les automates sont des classes de modèles de plus

haut niveau, qui sont plus appropriés pour la modélisation, l'analyse et la surveillance. Nous présentons dans ce qui suit l'équivalence de comportement entre le STT des RdPH élémentaires et le STT des AHL, ainsi que la preuve mathématique pour cette dernière.

4.4. La bisimilarité temporelle entre les RdPH élémentaires et les AHL

Dans cette section, une preuve de justesse de l'algorithme de translation est donnée. Cela garantira que la translation est correcte et se termine. Il sera prouvé que la sémantique d'un RdPH élémentaire et son AHL translaté sont temporellement bisimilaires (Définition 4.9). Pour cela, nous donnons des sémantiques formelles pour le RdPH élémentaire et l'AHL (Définition 4.10 et 4.11) en termes de STT. Par conséquent, nous prouvons la bisimilarité entre le STT du RdPH élémentaire et de l'AHL. Une bisimilarité entre deux systèmes de transitions assure que toute action d'un des systèmes peut être simulée par l'autre dans le sens d'un comportement équivalent dans un autre système.

Pour prouver la bisimilarité entre les deux STT, nous prouvons que le STT du RdPH élémentaire peut être simulé par le STT de l'AHL et ce dernier peut être simulé par STT du RdPH élémentaire dans le sens inverse. Les sémantiques des RdPH élémentaires et les AHL sont donnés ci-dessous pour introduire la preuve.

Définition 4.10. (Sémantique du RdPH élémentaire) La sémantique d'un RdPH élémentaire est un système de transitions temporisé $S_{EHPN} = \left(S^E, S_0^E, \sum, \rightarrow_E\right)$ tel que:

- $S^E = \left(m_D, m_C, t, v\right)$ est l'ensemble des états ;
- $S_0^E = \left(m_{D0}, m_{C0}, \overline{0}, v_0\right)$ est l'état initial ;
- $\sum$ est l'ensemble des actions (événements), $\sum = T^D \cup \left\{m_i = 0\right\}$;
- $\longrightarrow_E \in S^E \times \left(\sum_g \cup R^+\right) \times S^E$ est une transition continue ou discrète, tel que:

A- La transition discrète $\left(m_D, m_C, t, v\right) \xrightarrow{a} _E \left(m'_D, m'_C, t', v'\right)$ avec $a \in \sum$ est divisée en deux formes :

- $a \in T^D$ (**le franchissement d'une transition T_j dans le RdP T-temporel**) est la première forme de la transition discrète :
  t_j est le temps associé à la transition T_j ; t_k est le nouveau temps associé aux transitions nouvellement activées (T_k) après le franchissement de T_j.
  $\left(m_D, m_C, t, v\right) \xrightarrow{a} _E \left(m'_D, m'_C, t', v'\right)$ si:

$$\begin{cases} M \geq Pre(.,T_j), T_j \in T^{D2} \text{ and } \begin{cases} m'_D = m_D - Pre(.,T_j) + Post(.,T_j) \\ m'_C = m_C \end{cases} \\ \alpha(T_j) \leq t_j \leq \beta(T_j) \\ t'_k = \begin{cases} 0 \text{ if } \uparrow enabled\ (T_k, M, T_j) \\ t_k \end{cases} \\ \forall (P_k \in T_i^{\circ}), T_j \in T^{D2}, \{T_i, T_{i+1}\} \in T^C, and\ \ v'_i = 0 \\ \forall (P_j \in T_j^{\circ}), T_j \in T^{D1}, and\ \ v'_{i+1} = V_{i+1} \end{cases}$$ [4]

- $a \in \{m_i = 0\}$ (**le vidage d'une place continue** P_i) est la deuxième forme de la transition discrète:

  $(m_D, m_C, t, v) \xrightarrow{a}_E (m'_D, m'_C, t', v')$ si:

  $$\begin{cases} v' = f(m_C, t) \\ m_i = 0 \end{cases}$$ [5]

B- La transition continue est le temps écoulé :

$(m_D, m_C, t, v) \xrightarrow{d}_E (m'_D, m'_C, t', v')$ si:

$$\begin{cases} m'_C = m_C + B.d \\ t' = t + d \\ \forall T_k \in T^D \ \ si\ \ m \geq Pre(.,T_k)\ \ Alors\ \ t'_k \leq \beta(T_k) \\ \forall P_i \in P^C \ \ si\ \ m_i > 0 \qquad Alors\ \ m'_i > 0 \end{cases}$$

❑

Définition 4.11. (Sémantique de l'AHL) La sémantique d'un AHL est définie comme un système de transitions temporisé $S_{HA} = (S^A, S_0^A, E, \rightarrow_A)$ tel que:

- $S^A = (q, x)$ est l'ensemble des états;
- $S_0^A = (q_0, x_0)$ est l'état initial;
- E est l'ensemble des actions (événements);
- $\longrightarrow_A \in S^A \times (E \cup R^+) \times S^A$ est une transition qui peut être continue ou discrète, tel que:

A- Transition discrète: franchissement d'une transition T_j dans un AHL

$(q, x) \xrightarrow{T_j}_A (q', x')$ si :

$$\begin{cases} \exists (q, a, Init, q) \in \delta \ \ pour\ que \\ g(x) = true\ ,\ \ inv(q)(x) = true \\ et\ \ inv(q')(x') = true \end{cases}$$

[4] Pour calculer les vitesses de franchissement instantanées, nous invitons le lecteur à consulter [David et Alla, 10]

[5] Les vitesses de franchissement instantanées sont calculées selon l'algorithme présenté en détail dans [David et Alla, 10]

B- La transition continue: le temps écoulé.

$(q, x) \xrightarrow{d}_A (q', x')$ si :

$$\begin{cases} q = q' \\ x' = F(d) \\ \forall\ 0 \le d' \le d,\ inv(q)(x + d') = true \end{cases}$$

❑

Après avoir définit les sémantiques des deux modèles, nous adopterons maintenant ces derniers pour introduire la notion de bisimilarité temporelle entre les RdPH élémentaires et les AHL. la définition suivante est lui consacrée.

Définition 4.12. Relation de similarité et bisimilarité temporelle entre les RdPH élémentaires et les AHL.

- Soit $S_{EHPN} = \left(S^E, S_0^E, \Sigma, \rightarrow_E\right)$ le STT du RdPH élémentaire et $S_{HA} = \left(S^A, S_0^A, E, \rightarrow_A\right)$ le STT de l'AHL; soit Φ une relation binaire sur $S^E \times S^A$. Nous écrivons $s\,\Phi\,s'$ pour $(s, s') \in \Phi$. Φ est une relation temporelle forte de S_{EHPN} by S_{HA} si les assertions suivantes sont vérifiées :

$$\begin{cases} \text{si } s_1 \in S_0^E, \text{ alors } \exists s_2 \in S_0^A \text{ tel que } s_1\ \Phi\ s_2\,; \\ \text{si } s_1 \xrightarrow{\delta}_E s_1', \text{ avec } \delta \in R^+ \text{ et } s_1\ \Phi\ s_2 \text{ alors } \exists s_2 \xrightarrow{\delta}_A s_2' \text{ tel que } s_1'\ \Phi\ s_2'\,; \\ \text{si } s_1 \xrightarrow{a}_E s_1' \text{ avec } a \in \Sigma \text{ et } s_1\ \Phi\ s_2 \text{ alors } \exists s_2 \xrightarrow{a}_A s_2' \text{ avec } a \in E \text{ tel que } s_1'\ \Phi\ s_2'\,; \end{cases}$$

- $\forall (S_{EHPN}\ \Phi\ S_{HA}) \wedge (S_{HA}\ \Phi^{-1}\ S_{EHPN})$, alors $S_{EHPN} \approx S_{HA}$. $\approx$ est une relation de bisimilarité temporelle entre S_{EHPN} et S_{HA}.

- $\forall\ S_1 = (q_1, x_1) \in S^E$ est un état de S_{EHPN}, $S_2 = (q_2, x_2) \in S^A$ est un état de S_{HA} et $S_1\ \Phi\ S_2 \Leftrightarrow m_D = \nabla(q_2) \wedge x_1 = x_2$, où ∇ est une fonction qui associe à chaque marquage discret un sommet donné.

❑

Remarque 4.4. La notion de similarité et bisimilarité temporelle entre le modèle $\boldsymbol{R_N}$ (Définition 4.1) et l'AHL reste valable pour le modèle global avec défauts $\boldsymbol{R_G}$ (Définition 3.3).

Théorème 4.1. Le STT du RdPH élémentaire (S_{EHPN}) et le STT de l'AHL (S_{HA}) sont temporellement bisimilaires,

$S_{EHPN} \approx S_{HA}$. ($\approx$ est une relation de bisimilarité temporelle).

❑

Preuve.

$\forall\ S_1=(q_1,x_1)\in S^E$ et $S_2=(q_2,x_2)\in S^A$ tel que $S_1\ \Phi\ S_2$, nous écrivons $S_1\ \Phi\ S_1'$ pour $(S_1,S_1')\in\Phi$. $S_{EHPN}\ \Phi\ S_{HA}$ si les assertions suivantes sont vérifiées:

$$\begin{cases}\text{si } S_1 \xrightarrow{d\in R^+}_E S_1', \text{alors} \exists\ S_2 \xrightarrow{d\in R^+}_A S_2', \text{ tel que } S_1'\ \Phi\ S_2' \\ \text{si } S_1 \xrightarrow{a\in\Sigma}_E S_1', \text{ alors } \exists\ S_2 \xrightarrow{a\in\Sigma}_A S_2', \text{ tel que } S_1'\ \Phi\ S_2'.\end{cases}$$

Nous prouvons tout d'abord si : $S_{EHPN}\ \Phi\ S_{HA}$??

1- Transitions continus:

Selon la sémantique du RdPH élémentaire (Définition 4.10), la transition continue $\xrightarrow{d\in R^+}_E$ est le temps écoulé. t' et m_C' vérifient : $t'=t+d$; $m_C'=m_C+B.d$ et $m_D'=m_D$ $\Rightarrow q_1'=q_1$, $x_1'=F(d)$.

Selon notre algorithme, la transition $\xrightarrow{d\in R^+}_E$ est translate à une transition $d=\left[\alpha_j\ \ \beta_j\right]$ entre sous sommets (Algorithme 4.1, ligne 9) ou entre sommets (ligne 29), dont seule la dynamique qui se change durant ce temps, alors $x_2'=F(d)$ et $\forall\ m_D'=m_D \Leftrightarrow \nabla(q_2')=\nabla(q_2)\Rightarrow q_2'=q_2$.

$$\left.\begin{array}{l}S_1\ \Phi\ S_2\ ,(q_1'=q_1)\wedge(q_2'=q_2)\Rightarrow q_1'=q_2' \\ x_1'=F(d) \text{ et } x_2'=F(d)\Rightarrow x_1'=x_2'\end{array}\right\}\ \Rightarrow S_1'\ \Phi\ S_2' \qquad (1)$$

2. Transitions discrètes:

Selon la sémantique du RdPH élémentaire, la transition discrète $\xrightarrow{a\in\Sigma}_E$ est:

a) le franchissement d'une transition T_j dans le RdP T-temporel est la transition discrète $\xrightarrow{a\in T^D}_E$:

si $M\geq Pre(.,T_j)\Rightarrow M=g(x_1)=true$, et si T_j est franchie, alors :

$m_C'=m_C$ et $m_D'=m_D-Pre(.,T_j)+Post(.,T_j)$

$\Rightarrow inv(q_2')(x_1')=true$ et $inv(q_2)(x_1)=true$

Selon notre algorithme (Algorithme 4.1, ligne 5-9), T_j correspond à la transition de la garde $m_i=S\Rightarrow m_i=g(x_2)$ si $g(x_2)=true$ et $inv(q_2)(x_2)=true$, alors $inv(q_2')(x_2')=true$.

$g(q_2')(x_1')=true$ et $g(q_2')(x_2')=true$ tel que $S_1\ \Phi\ S_2$ alors $x_1'=x_2'\Leftrightarrow S_1'\ \Phi\ S_2'$ (2)

b) le vidage d'une place continue P_i est la transition discrète $\xrightarrow{a\in\{m_i=0\}}_E$:

elle est considérée comme un événement discret de type (C1 événement), si $m_i=0$, $g(x_1)=0$ $\Rightarrow m_i=g(x_1)=true\Rightarrow m_D'=m_D$ et $m_C'=m_C-Pre(.,P_i)$ alors $g(q_2')(x_1')=true$. Ce vidage est translaté dans ce travail (Algorithme 4.1, ligne 38) à une transition avec la garde $m_i=0$ $\Rightarrow g(x_2)=0\Rightarrow m_i=g(x_2)$.

Si $g(x_2) = true$ et $inv(q_2)(x_2) = true$, alors $g(q_2')(x_2') = true$, tel que $S_1 \Phi S_2$ alors :

$$x_1' = x_2' \Rightarrow S_1' \Phi S_2' \quad (3)$$

De (1)-(3) nous écrivons: pour toute S_1 et S_2 en relation.

$$S_{EHPN} \Phi S_{HA} \quad (4)$$

Maintenant, nous prouvons si : $S_{HA} \Phi^{-1} S_{EHPN}$??

Considérons : $S_1' = (q_1', x_1'), S_2' = (q_2', x_2')$ tel que $S_2' \Phi^{-1} S_1'$

1. Transitions continues :

Selon la sémantique de l'AHL (Définition 4.11), la transition continue $\xrightarrow{d \in R^+}_A$ avec la garde $d = [\alpha_j \ \beta_j]$ est le temps écoulé. Alors, durant ce temps, le sommet ne change pas, seulement le vecteur d'état qui change ($x_2 = F(d)$) et $q_2' = q_2$.

Selon notre algorithme, la transition $\xrightarrow{d \in R^+}_A$ avec la garde $d = [\alpha_j \ \beta_j]$ correspond à une transition $\xrightarrow{d \in R^+}_E$ du RdP T-temporel, alors seulement le marquage continu qui va changer. Ce changement correspond au changement du vecteur d'état durant ce temps, alors $x_1 = F(d)$ et $m_C' = m_C + B.d$.

Nous avons $m_D = \nabla(q_2)$ (Définition 4.12) et $q_2' = q_2 \Leftrightarrow \nabla(q_2') = \nabla(q_2) \Leftrightarrow m_D' = m_D \Rightarrow q_1' = q_1$.

on peut conclure que $\forall \ S_2' \Phi^{-1} S_1'$, $(q_2' = q_2) \wedge (q_1' = q_1) \Rightarrow S_2 \Phi^{-1} S_1$ (5)

2. Transitions discrètes:

Selon la sémantique de l'AHL, après le franchissement d'une transition T_j: x_2 et $g(x_2')$ vérifient: si $g(x_2') = true$, $\Rightarrow inv(q_2')(x_2') = true$, nous distinguons deux cas :

a) **si** $g(x_2') = m_i$ et $m_i = S \Rightarrow g(x_2') = M = true$, alors $inv(q_2)(x_2) = true$ et selon notre algorithme (Algorithme 4.1, ligne 8), $m_i = S$ **correspond au franchissement d'une T_j dans le RdP T-temporel.**

si $M \geq Pre(.,T_j)$; $g(x_1') = M$ et T_j est franchie, alors $m_C = m_C'$ mais le marquage discret va se changer $m_D = m_D' - Pre(.,T_j) + Post(.,T_j) \Rightarrow inv(q_2)(x_1) = true$.

On peut conclure que $\forall \ S_2' \Phi^{-1} S_1'$, $inv(q_2)(x_2) = true$ et $inv(q_2)(x_1) = true \Rightarrow x_2 = x_1$

$$\Rightarrow S_2 \Phi^{-1} S_1 \quad (6)$$

b) **si** $g(x_2') = 0$ et $m_i = 0 \Rightarrow g(x_2') = 0 = true$, alors $inv(q_2)(x_2) = true$ et selon notre algorithme (Algorithme 4.1, ligne 38), $m_i = 0$ **correspond au vidage d'une place continue P_i .**

si $m_i = 0 \Rightarrow g(x_1') = 0 = true$, $m_D = m_D'$ et $m_C = m_C' - Pre(.,P_i)$, alors $inv(q_2)(x_1) = true$

On peut conclure que $\forall\ S_2'\ \Phi^{-1} S_1'$, $inv(q_2)(x_2) = true$ et $inv(q_2)(x_1) = true \Rightarrow x_2 = x_1$

$$\Rightarrow S_2\ \Phi^{-1} S_1 \qquad (7)$$

De (5)-(7) nous écrivons : pour toute S_2 et S_1 en relation.

$$S_{HA}\ \Phi^{-1}\ S_{EHPN} \qquad (8)$$

Finalement, nous écrivons: $\forall\ (S_{EHPN} \Phi\ S_{HA}) \wedge (S_{HA}\ \Phi^{-1}\ S_{EHPN})$

$$S_{EHPN} \approx S_{HA} \qquad (9)$$

$\approx$ est une relation de bisimilarité temporelle entre S_{EHPN} et S_{HA}

Nous avons présenté une procédure de translation structurelle d'un RdPH élémentaire à un AH. Nous allons appliquer cette translation à la modélisation de systèmes tolérants aux fautes à travers un exemple. Nous considèrerons le modèle en comportement normal $\boldsymbol{R_N}$, puis celui avec fautes $\boldsymbol{R_G}$.

4.5. Application de l'algorithme de translation

Dans cette section, un système de chauffage de liquides est considéré, l'AHL de l'exemple est construit à partir de son modèle du RdPH élémentaire de sorte que les comportements des deux modèles sont dans une correspondance un-à-un en utilisant l'algorithme proposé. L'AHL résultant de la translation et le modèle du RdPH élémentaire sont temporellement bisimilaires selon ce qui a été déjà présenté.

Exemple illustratif.

Le système de chauffage de liquides, illustré dans la Figure 4.10(a), comporte deux vannes V_1 et V_2, un bac, une résistance, un thermostat et deux capteurs de niveau : le capteur C_1 surveille le niveau maximal et le capteur C_2 surveille le niveau minimal. Le liquide à chauffer est introduit par la vanne V_1 avec un débit 5 u.v/u.t (unité de volume par unité de temps). Quand le niveau de ce liquide atteint le niveau maximal 500 u.v, le capteur C_1 génère un événement s_1 de notification au contrôleur pour qu'il commande la fermeture de la vanne V_1.

Par la suite, le liquide est chauffé pendant une durée de 40 u.t. Le liquide est ensuite évacué à travers la vanne V_2 avec un débit 8 u.v/u.t jusqu'à ce que le bac soit vide. A ce moment, le capteur C_2 génère un événement s_2 de notification, le contrôleur commence alors un nouveau cycle de chauffage.

Nous supposons que le fonctionnement de ce système peut être affecté par deux défauts non observables :

• le premier défaut correspond à l'existence d'une fuite dans le bac avec un débit 0.5 u.v/u.t. Ce défaut est représenté par l'événement non observable σ_1 ;

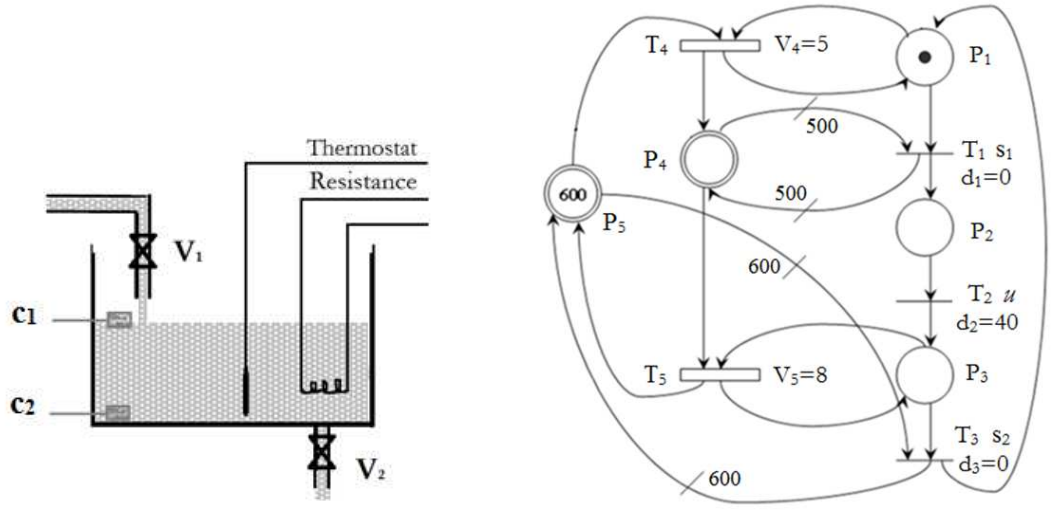
• le deuxième défaut correspond au blocage de la vanne V_1 en position fermée. Ce défaut est représenté par l'événement non observable σ_2 .

Les événements s_1 et s_2 sont observables et correspondent aux notifications générées, respectivement, par les capteurs C_1 et C_2. L'événement *u* est non observable. Cet événement a été introduit pour permettre la représentation d'une transition autonome. À l'état initial, *t=0*, le bac est vide.

Le modèle du RdPH élémentaire de ce système en fonctionnement normal est présenté sur la Figure 4.10(b), les places discrètes P_1, P_2 et P_3 représentent respectivement l'état de remplissage, chauffage et vidange. La place continue P_4 représente le bac, son marquage représente le volume de liquide dans le bac. La place P_5 est une place complémentaire à P_4 son marquage correspond à la partie vide du bac. Le volume du bac est de 600 u.t.

Aux transitions discrètes T_1, T_2 et T_3 sont associées respectivement les durées $d_1=0$, $d_2=40$ et $d_3=0$ des événements s_1, u, s_2, sachant que d_1 et d_3 sont des durées de synchronisation. Les transitions continues T_4 et T_5 représentent respectivement la vanne V_1 et V_2 ; il leur est associé leurs vitesses de franchissement maximales qui représente le débit d'entrée et de sortie.

Au début, la transition T_4 est franchie à sa vitesse maximale (5 u.v/u.t), le bac commence à se remplir à travers la vanne V_1. Lorsque le bac atteint le niveau maximal 500 u.v, il y a franchissement de la transition discrète T_1, et donc le système passe de l'état remplissage à l'état de chauffage. Après 40 u.t il passe à l'état de vidange à travers la vanne V_2, et la transition continue T_5 est activée à sa vitesse maximale (8 u.v/u.t) jusqu'à l'occurrence de l'événement observable s_2.



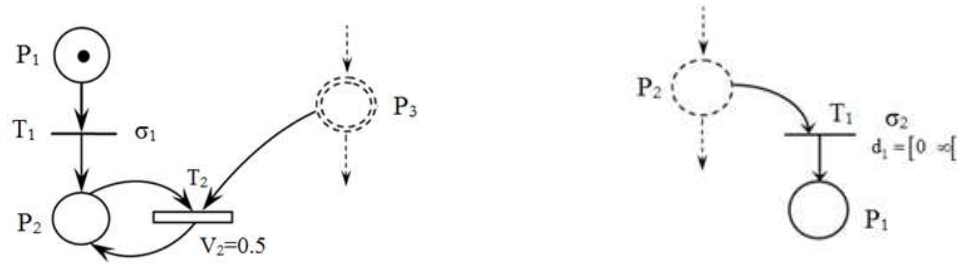
(a) *(b)*

Figure 4.10. a. Système de chauffage de liquides, b. RdPH élémentaire décrivant le comportement normal.

L'occurrence de défaut de fuite concerne la partie continue, il est modélisé par la transition continue T_2 attachée à la place continue P_3 avec un arc de la place à la transition T_2. Cette transition est contrôlée par l'occurrence de l'événement non observable σ_1 associé à la transition discrète T_1 (Figure 4.11(a)).

Le défaut du blocage de la vanne V_1 en position fermée est modélisé par la transition discrète T_1 attachée à la place discrète P_2 avec un arc de la place à la transition T_1. Cette transition est contrôlée par l'occurrence de l'événement non observable σ_2 associé à la transition discrète T_1. Ce défaut peut survenir à tout moment pendant le processus de remplissage (Figure 4.11(b)).

Pour obtenir le modèle global (modèle décrivant le comportement normal et défaillant) donné à la Figure 4.12, il est nécessaire d'effectuer une composition structurelle du modèle en fonctionnement normal avec les modèles des défauts.



(a) *(b)*

Figure 4.11. (a) le modèle décrivant la fuite (b) le modèle décrivant le blocage de la vanne

Le comportement d'un RdPH est généralement modélisé par un graphe d'évolution dont les nœuds représentent les IB-états et les arcs sont étiquetés par les événements (dont l'occurrence fait changer d'IB-état) avec leurs dates d'occurrence. Le graphe d'évolution du RdPH élémentaire global possède 6 IB-états comme présenté en Figure 4.13. Le graphe d'évolution du comportement en présence de fuite est donné par la Figure 4.14.

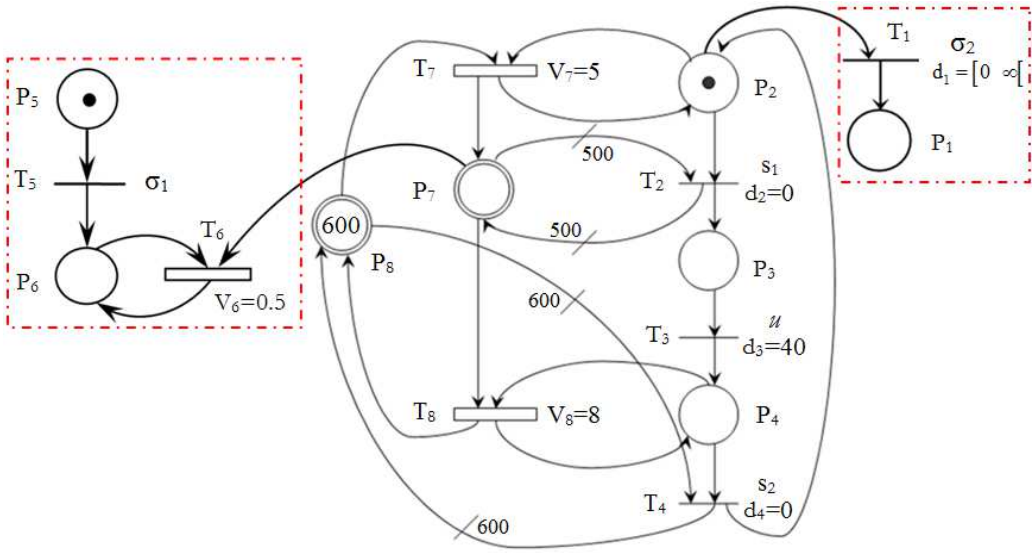


Figure 4.12. Le modèle du RdPH élémentaire global.

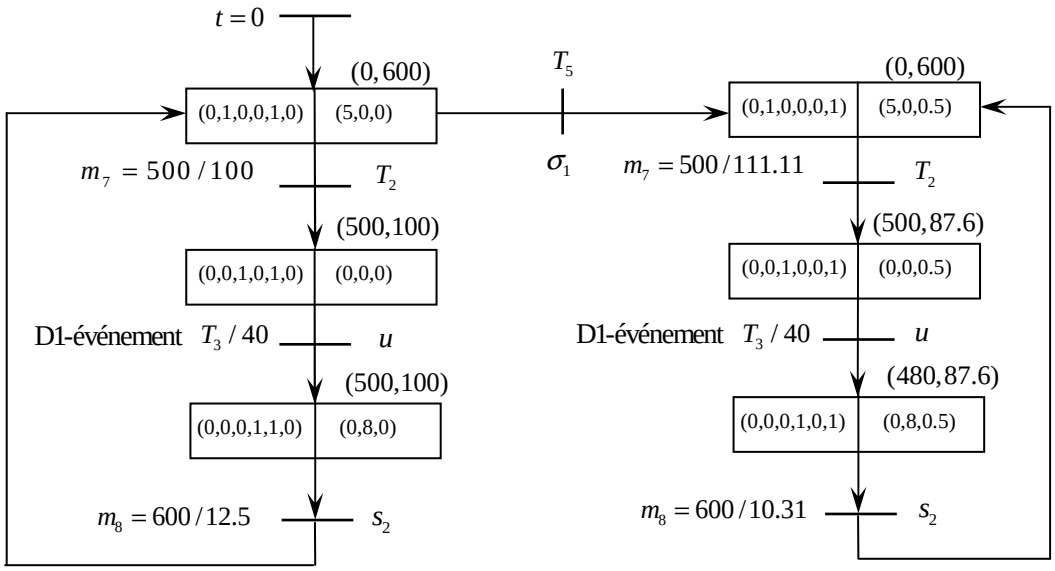


Figure 4.13. Graphe d'évolution du RdPH élémentaire global.

En se basant sur les valeurs des bilans de marquages[6], nous pouvons constater que l'occurrence d'une fuite affecte les débits d'entrée/sortie du liquide durant les phases de remplissage, de chauffage et de vidange ce qui implique un changement dans les dynamiques. Ce changement correspond au changement du marquage des places continues P_7 et P_8 pour chaque marquage discret du RdP T-temporel. Pour cela, nous avons ajouté une transition continue (T_6) reliée à la place continue P_7 par un arc. Cette transition peut être validée par l'occurrence de l'événement non observable σ_1 associé à la transition discrète T_5. Le changement qui suit cet événement affecte les dates d'occurrence des événements observables (s_1 et s_2). Aussi le blocage de la vanne V_1 se produit après l'occurrence de l'événement non observable σ_2. Cet événement est associé à la transition discrète T_1, reliée à la place discrète P_2 qui modélise l'état initial du système (phase de remplissage).

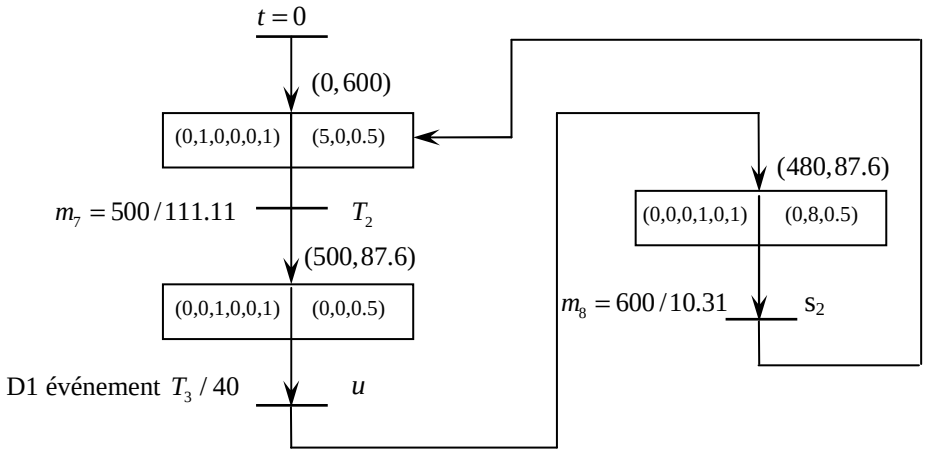


Figure 4.14. Graphe d'évolution du RdPH élémentaire en présence de fuite.

[6] La balance de marquage correspond à la différence entre le flux l'entrée et le flux de sortie pour chaque place continue P_i d'un RdPCC.

Le modèle avec défauts est translaté directement en utilisant l'algorithme présenté précédemment. L'exécution de l'algorithme 1, dans cet exemple illustratif, est de la forme décrite sur l'algorithme 2 pour mieux expliquer la procédure de translation du comportement normal. Seul l''état de blocage a été ajouté ici car il correspond à un sommet unique. Bien sûr, de la même manière l'algorithme fonctionne pour le comportement en présence de fuite. Les étapes structurelles de translation sont présentées ci-après.

Algorithme 4.2 : l'algorithme 1 exécuté pour l'exemple

1: «Initialisation» ;
2: isoler la partie du RdP T-temporel du modèle hybride de la partie continue (RdPCC);
3: construire le graphe de marquage accessible du RdP autonome sous-jacent ;
4: **tant que** $\forall\{P_7, P_8\} \in P^C$, $card\left(T^{D2} \cap P_i^\circ\right) = 2$, **faire**
5: $\{P_2, P_4\} \in P^D$ sources des transitions $\{T_2, T_4\} \in T^{D2}$ **ont une** transition de sortie $T_1 \in T^{D1}$, **faire**
6: éclater les sommets L_1 et L_3 correspond aux T_2, T_4 et en deux sous-sommets L_{11}, L_{12} et L_{31}, L_{32} ;
7: associer les horloges aux sommets L_{12} et L_{32} avec initialisation à leurs entrées;
8: relier L_{11}, L_{12} et L_{31}, L_{32} avec transitions, et lui associer les gardes $m_7 = 500$ et $m_8 = 600$
9: créer une transition de sortie pour L_{12} et L_{32} et lui associer les gardes $d_2 = 0$ et $d_4 = 0$;
10: $d_2 = 0$ et $d_4 = 0$ correspondent aux T_2 et T_4 **faire**
11: éliminer les sommets L_{12} et L_{32} ;
12: relier le sommet L_{11} à L_2 et L_{31} à L_{11} en gardant les gardes $m_7 = 500$ et $m_8 = 600$
13: créer pour la transition de sortie $T_1 \in T^{D1}$ un sommet destination L_4 ;
14: relier L_4 au sommet L_{11}, par une transition et lui associer la garde $d_1 = [0 \ \infty[$.
15: **fin tant que.**
16: « configuration_RdPCC », configurer la partie continue du RdPCC pour chaque marquage discret ;
17: mettre $m_0^* = P^C$;
18: créer $L_{f40}, L_{f10}, L_{f20}$ et L_{f30} pour chaque sommet et lui associer les activités $\dot{M}_{40} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\dot{M}_{10} = \begin{bmatrix} 5 \\ -5 \end{bmatrix}$, $\dot{M}_{20} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ et $\dot{M}_{30} = \begin{bmatrix} -8 \\ 8 \end{bmatrix}$;
19: « vérification_marquage » ;
20: les marquages des C-places P_7 et P_8 **sont décroissants Alors**
21: créer depuis L_{f10} et L_{f30} une transition pour chaque place P_i et lui affecter les gardes $m_8 = 0$ et $m_7 = 0$;
22: pour chaque transition construite, créer L_{f11} et L_{f31} sommets destination de L_{f10} et L_{f30} et lui affecter les activités $\dot{M}_{11} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ et $\dot{M}_{31} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$;

23: fusionner les mêmes sommets ayant la même activité ;

24: $\dot{m}_8 = 0$ dans L_{f11} et $\dot{m}_7 = 0$ dans L_{f31} pour chaque P_i dont le marquage est initialement décroissant **Alors**

25: aller « comparaison_vecteurs » ;

26: « comparaison_vecteurs », remplacer les transitions en comparant les vecteurs booléens ;

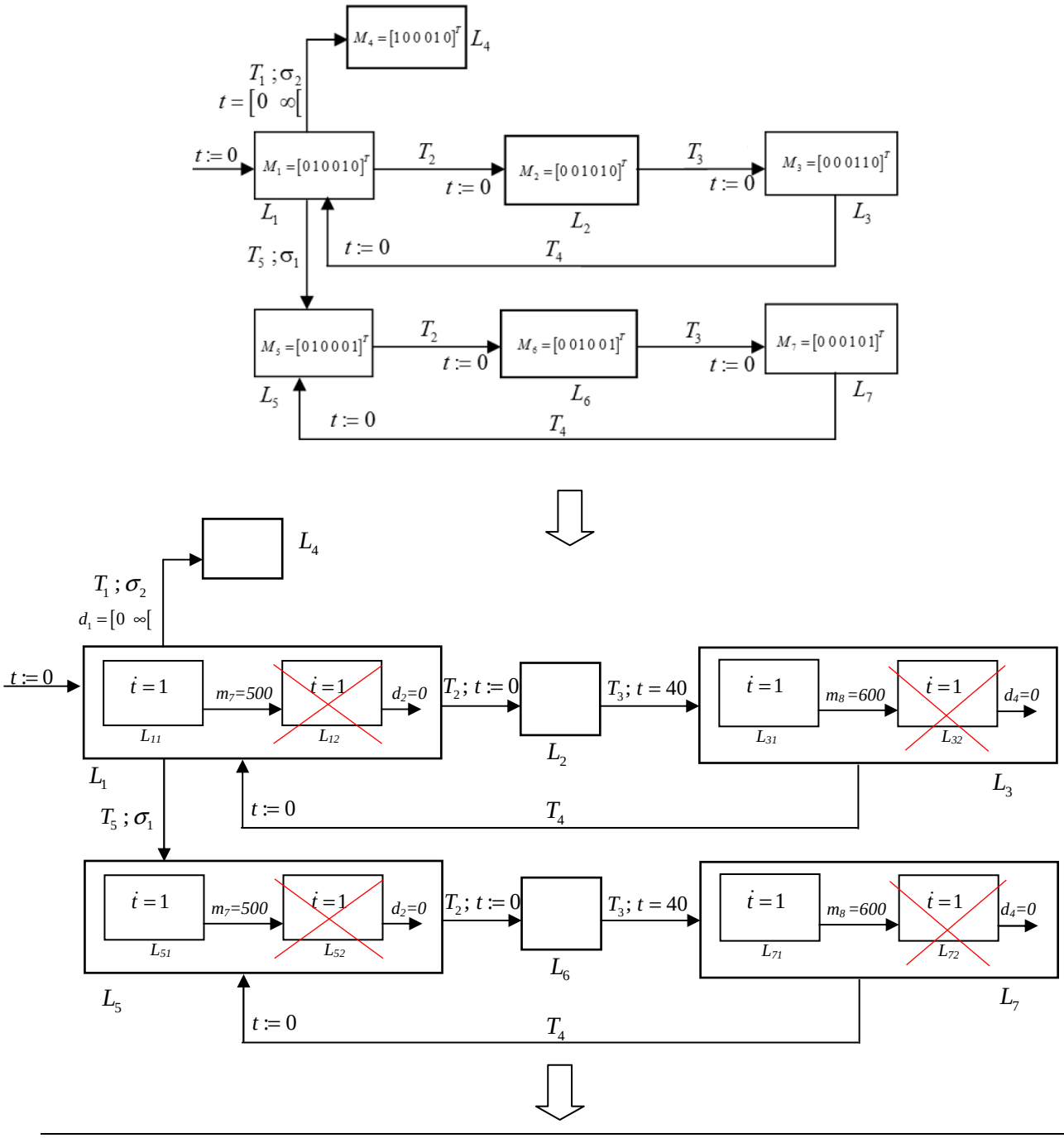
27: calculer le vecteur booléen $\bar{M}$ pour chaque sommet interne ;

28: $(\bar{M}_{10} \geq \bar{M}_{20}) \wedge (\bar{M}_{20} \geq \bar{M}_{30})$ **Alors**

29: remplacer la transition entre L_{11} et L_2 par une transition entre L_{f10} et L_{f20} ;

30: remplacer la transition entre L_{31} et L_{11} par une transition entre L_{f30} et L_{f10}.

31: **Fin**



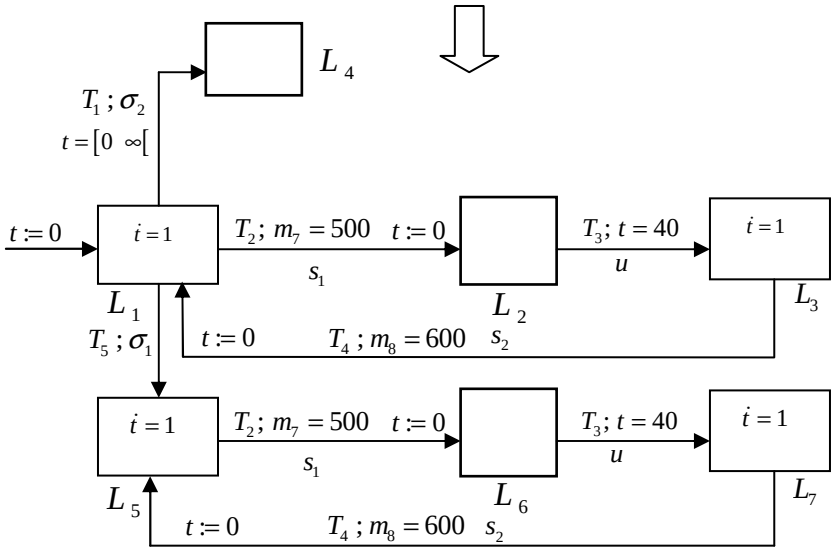


Figure 4.15. Étapes de construction de l'automate initial.

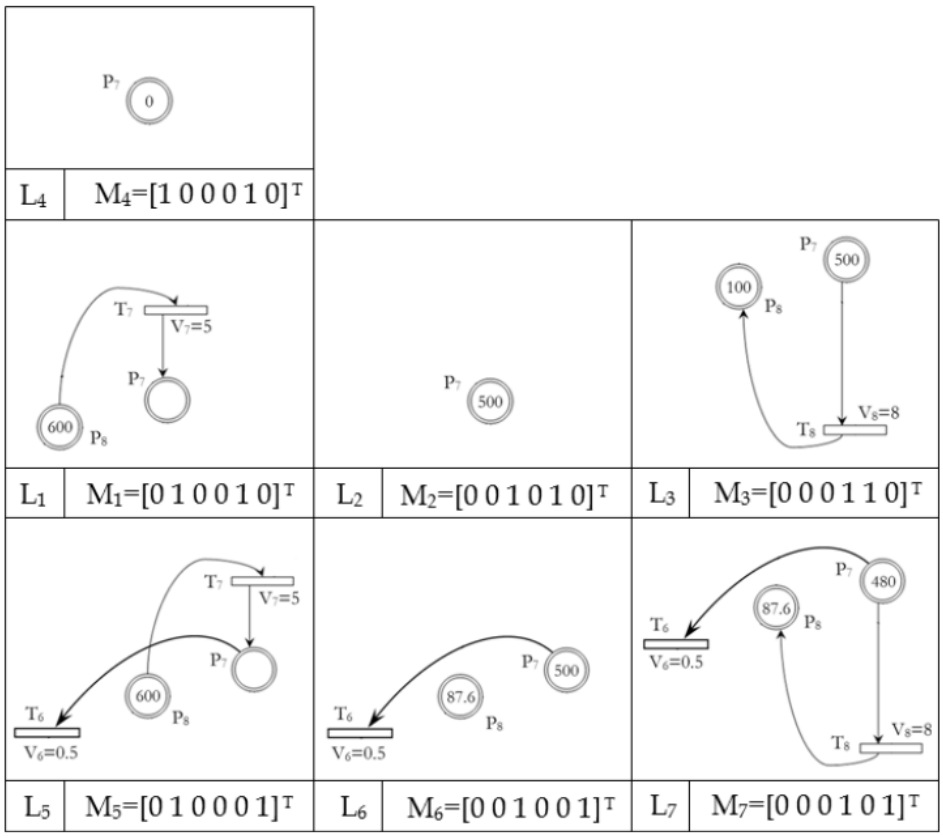


Figure 4.16. Configuration du RdPCC de chaque marquage discret.

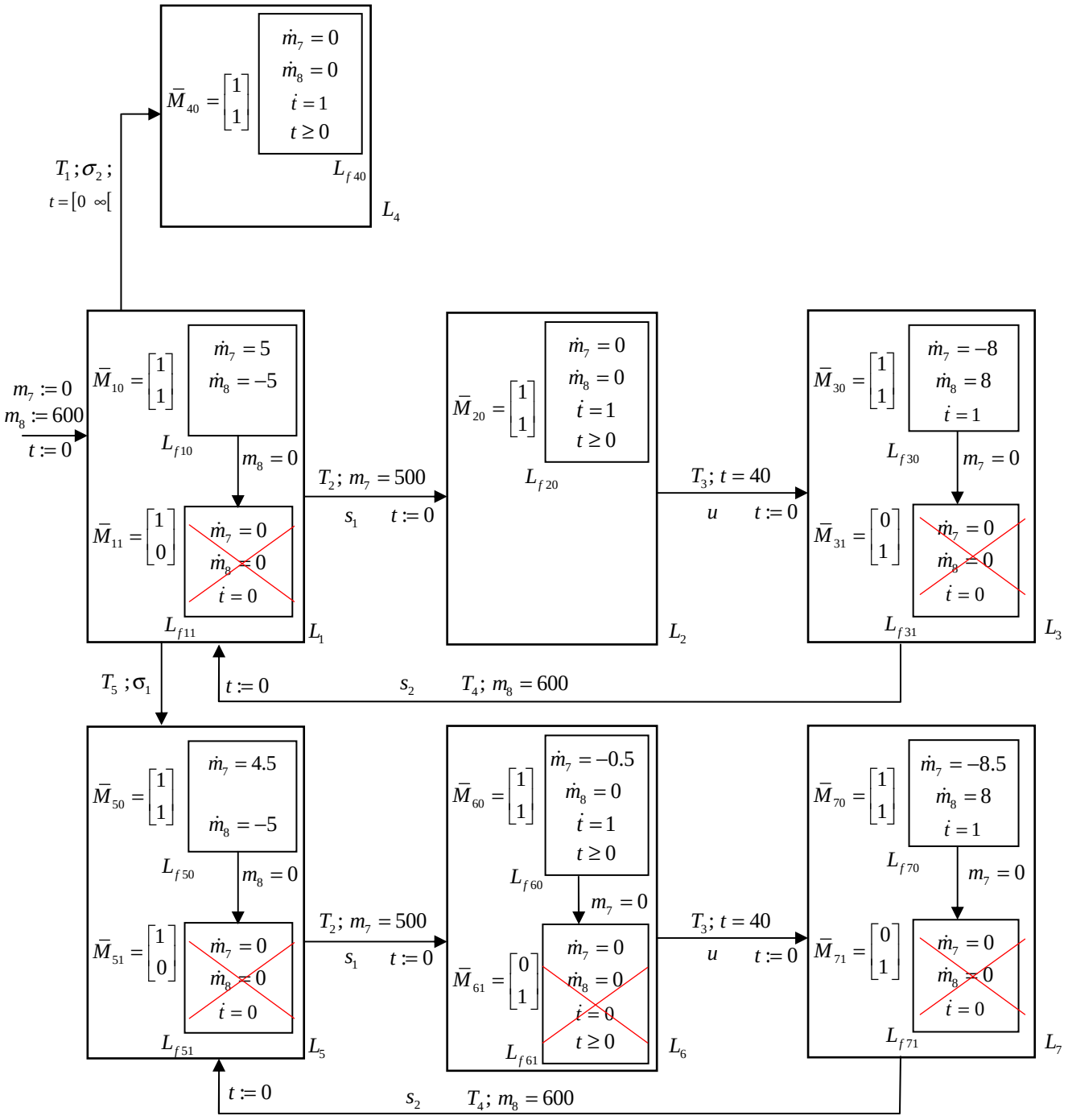


Figure 4.17. Forme hiérarchique de l'automate hybride final.

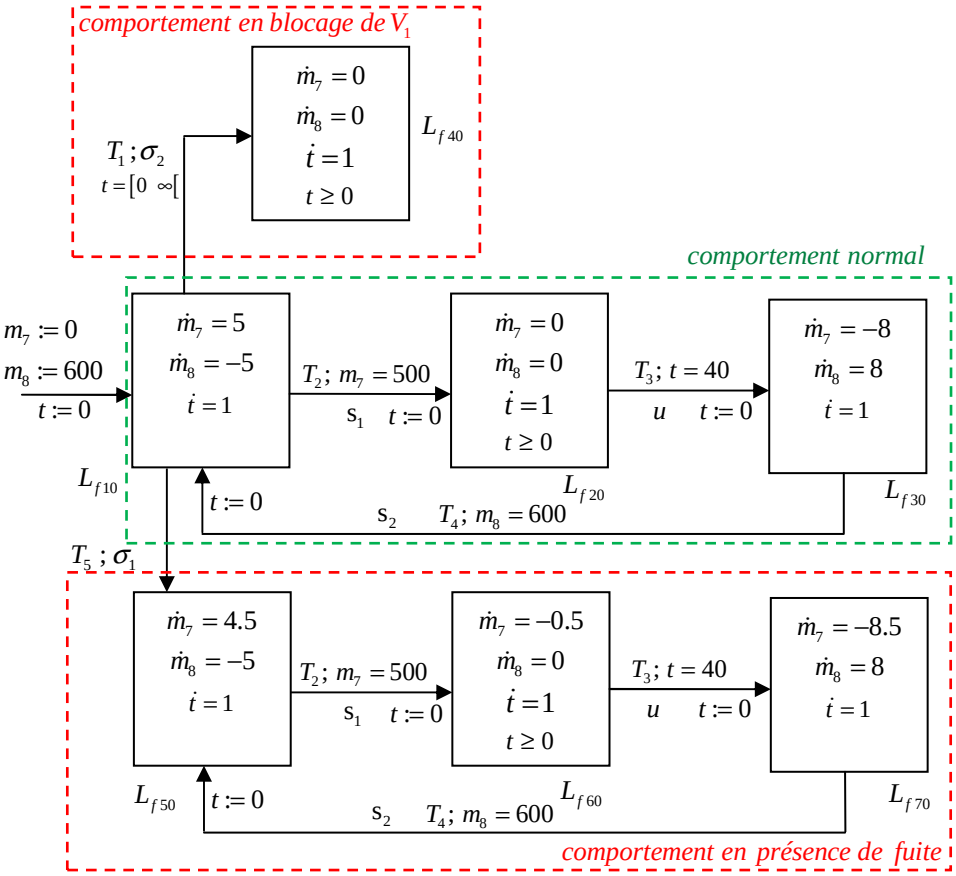


Figure 4.18. Automate hybride final

La seule différence entre les sommets du comportement normal et les sommets du comportement défaillant réside dans la valeur de la dynamique de m_7 m_8. Par définition, les deux comportements sont semblables, seules leurs dynamiques changent à cause du défaut.

Le système peut s'exécuter plus rapidement ou plus lentement selon les événements observables (l'ouverture ou la fermeture des vannes V_1 et V_2), ou selon des événements non observables (blocage des vannes soit en fermeture, ou en ouverture). Ce changement de fonctionnement impliquera un changement d'état du système de l'état normal vers l'état défaillant.

La transition T_5 et T_1 ne seront jamais franchies si les événements non observables σ_1 et σ_2 respectivement ne se produisent pas auparavant. Si une alarme se déclenche, c'est que le comportement du système est passé forcément au comportement défaillant.

Si la fuite se produit, donc la transition T_5 est franchie et le système passe à l'état défaillant, ce qui provoque réellement un ralentissement dans la tâche de remplissage et une accélération dans la tâche de vidange en comparant les délais de séjour de chaque l'IB-état du système en comportement défaillant et le comportement normal (Figure 4.13) et (Figure 4.14)

respectivement. Par contre, lorsqu'un blocage de la vanne V_1 en fermeture se produit, aucun autre événement ne sera observé.

4.6. Analyse de l'automate résultant par l'outil PHAVer

Nous avons vu précédemment que les trajectoires possibles dans un automate hybride sont constituées chacune par une succession de transitions continues pendant lesquelles le système évolue selon la dynamique continue du sommet courant, et de transitions discrètes correspondant aux franchissements. L'état d'un automate hybride peut avoir deux types de successeurs, discrets et continus.

Pour savoir si la région (q', x') est atteignable depuis la région (q, x), deux méthodes peuvent être utilisées. La première est dite méthode d'analyse en avant, elle est basée sur le calcul de l'espace de tous les états qui peuvent être atteints depuis des états appartenant à la région (q, x). Cet espace est dit successeur de la région (q, x). La deuxième est basée sur le calcul de l'ensemble de tous les états à partir desquels on peut atteindre des états de la région (q', x'). Cette méthode, duale à la méthode d'analyse en avant est appelée méthode d'analyse en arrière.

Ces deux procédures de calculs sont complémentaires et il n'est pas possible de prévoir laquelle sera la plus efficace dans un cas particulier.

L'analyse d'atteignabilité constitue un problème central dans la vérification des propriétés des systèmes hybrides modélisés par des automates hybrides. L'existence d'un algorithme effectuant le calcul de l'espace atteignable est très important pour l'analyse. Il est prouvé que, en dehors de quelques cas particuliers, ce problème n'est pas décidable [Henzznger, 95].

Les cas particuliers où la convergence de l'algorithme peut être garantie appartiennent principalement à la classe des automates hybrides linéaires, et les différentes régions sont des polyèdres, et sont donc relativement faciles à manipuler. Cela peut être confirmé en utilisant un outil informatique basé sur des approximations, on peut analyser l'automate hybride résultant.

A cet effet, nous utiliserons l'outil PHAVer[7] pour la vérification exacte des propriétés de sûreté du système afin de calculer l'espace d'états atteignables pour l'automate résultant. L'automate hybride linéaire construit après l'analyse, ne contient pas des sommets qui ne seront jamais visités. Nous présenterons nos résultats après le calcul de l'espace atteignable en appliquant l'analyse en avant.

[7] Nous invitons le lecteur à consulter l'ANNEXE B pour explorer cet outil informatique.

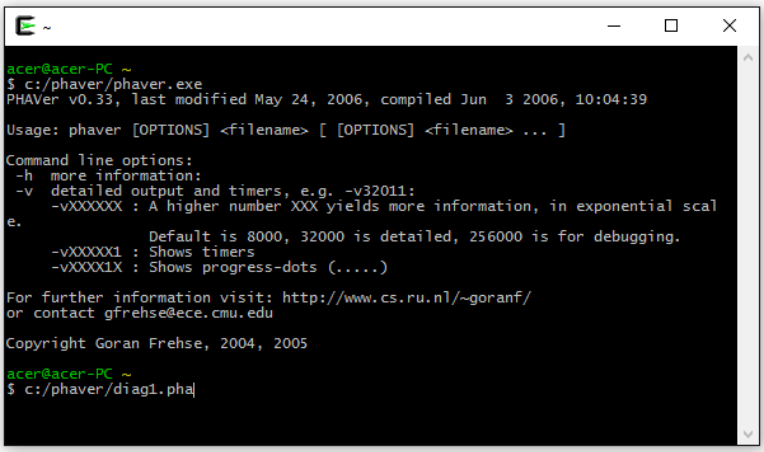


Figure 4.19. Exécution de l'outil PHAVer sur le compilateur Cygwin[8].

Programme du fichier « diag1.pha »

```
automaton diag1
      state_var: m7, t;
      synclabs: T1, T2, T3, T4, T5;

loc S1: while true wait {m7'==5 & t'==1}
              when m7==500 sync T2  do {m7'==m7 & t'==0} goto S2;
              when t<=1000 sync T1  do {m7'==m7 & t'==0} goto S4;
              when t<=1000 sync T5  do {m7'==m7 & t'==0} goto S5;

loc S5: while true wait {m7'==4.5 & t'==1}
              when m7==500 sync T2  do {m7'==m7 & t'==0} goto S6;

loc S6: while true wait {m7'==-0.5 & t'==1}
              when t==40  sync T3 do {m7'==m7 & t'==0} goto S7;

loc S7: while true wait {m7'==-8.5 & t'==1}
              when m7==0  sync T4 do {m7'==m7 & t'==0} goto S5;

loc S2: while true wait {m7'==0 & t'==1}
              when t==40  sync T3 do {m7'==m7 & t'==0} goto S3;

loc S3: while true wait {m7'==-8 & t'==1}

              when m7==0  sync T4 do {m7'==m7 & t'==0} goto S1;

           initially S1 & m7==5 & t==0;
end
echo "analyse en avant";
analyse_avant = diag1.reachable;
analyse_avant.print;
analyse_avant.print("D:/phaver/diag1.txt", 0);
```

[8] Voir annexe B

Résultats après la compilation du fichier :

```
c:/phaver/diag1.pha

  Parsing file c:/phaver/diag1.pha.
  ---------------------------------
analyse en avant
     .....
analyse_avant = diag1.{S1 & (m7 - 5*t == 0 & t >= 0 | m7 - 5*t == 5 &
t >= 0 | t == 0 & m7 == 5),
S2 & m7 == 500 & t >= 0,
S4 & true,
S5 & (-2*m7 + 9*t >= -10000 & t >= 0 & 2*m7 - 9*t >= 0 | -2*m7 + 9*t
>= -10010 & t >= 0 & 2*m7 - 9*t >= 10),
S6 & 2*m7 + t == 1000 & t >= 0,
S7 & 2*m7 + 17*t == 960 & t >= 0,
S3 & m7 + 8*t == 500 & t >= 0};
Finished. Exiting.
```

4.7. Conclusion

La modélisation par le RdPH élémentaire présente plusieurs avantages par la simplicité graphique de la représentation des concepts fréquents dans les SDs. Cependant, l'analyse et la vérification des propriétés du système sur la base du RdPH élémentaire n'est pas une tâche aisée. Dans ce cadre, les automates sont plus éligibles quant à l'analyse et la vérification.

Le problème de tel formalisme est que la modélisation est délicate, vue l'importance de nombre de sommets/variables. Pour tirer profit de chacun des deux modèles, nous avons introduit, dans ce chapitre, un modèle bisimilaire au modèle RdPH élémentaire : l'AHL. Nous avons présenté, pour ce faire, une méthode structurelle permettant de translater le RdP en automate. D'abord, nous avons présenté quelques techniques similaires utilisées dans la littérature. Nous avons distingué par la suite, le principe de la méthode de translation proposée. Un nouvel algorithme permettant de garantir la translation de manière systématique a été établi. Une preuve mathématique pour cette translation a également été présentée. Les idées fournies dans ce chapitre ont été illustrées à travers un exemple simple, l'automate résultant a été analysé en utilisant le logiciel PHAVer et a permis de construire l'automate dynamiquement atteignable. L'intérêt de notre travail est que si nous changeons de marquage initial, il suffit de partir de l'automate obtenu par translation et d'utiliser à nouveau PHAVer.

En se basant sur le modèle d'AHL obtenu, nous pouvons aborder le problème de diagnostic. Le chapitre suivant est consacré au traitement de ce problème.

CHAPITRE 5

CONCLUSIONS ET PERSPECTIVES

Résumé : Dans ce chapitre conclusif, nous survolons quelques contributions développées dans le cadre du diagnostic des SEDs et SDHs les plus intéressantes pour développer nos idées en vue de proposer une nouvelle méthode de diagnostic complémentaire pour notre approche de modélisation des SDHs tolérants aux défauts. Ensuite, le cadre de ces idées sera présenté à la fin de ce chapitre en présentant quelques perspectives de continuation de nos travaux de recherche.

1.1. Introduction

Le diagnostic des défauts est très important pour assurer le bon fonctionnement des systèmes et éviter les pertes de biens et de vies humaines. Dans notre travail de recherche, on s'intéresse au diagnostic à base de modèle. Il existe plusieurs méthodes de diagnostic dans la littérature, ces méthodes sont développées dans deux cadres, à savoir : le cadre des Systèmes à Evénements Discrets (SEDs) et le cadre des Systèmes Dynamiques Hybrides (SDHs).

L'efficacité d'une méthode de diagnostic de SDH peut se mesurer par sa capacité à exploiter, d'une manière optimale, les deux aspects présentés par ces systèmes: l'aspect continu à travers les variables d'état continues et l'aspect discret à travers les événements discrets. Habituellement, ces deux aspects sont abordés par les approches différentes, issues des systèmes continus ou des systèmes à événements discrets (SED).

Nous présentons dans ce chapitre conclusif une brève présentation des méthodes de diagnostic des SDHs. Dans cette présentation, nous respectons une classification de ces méthodes selon: les méthodes issues des approches continues et les méthodes issues des approches SED. Par la suite, et comme perspectives de nos travaux de recherche, nous distinguons une nouvelle approche de diagnostic des SDHs, dont le cadre sera présenté à la fin de ce chapitre.

5.2. Vers le diagnostic des systèmes dynamiques hybrides

L'utilisation d'une approche de diagnostic reposant sur une abstraction des dynamiques continues d'un SDH, à travers l'utilisation de modèles SED, entraîne une perte considérable d'informations, parfois indispensables pour l'identification des défauts. Ainsi, dans certains cas, les comportements défaillants se manifestent par une déviation des trajectoires des dynamiques continues du système. Ceci rend l'utilisation d'une démarche de diagnostic fondée sur une abstraction purement discrète de l'évolution du système, inadéquate pour l'identification des défauts.

Afin de pallier à ce problème, nous pouvons distinguer, d'après l'approche proposée dans [Derbel, 09], une démarche de diagnostic qui prend en considération l'aspect temporel dans l'évolution événementielle du SDH à diagnostiquer. Cette approche repose sur l'utilisation d'un modèle global du système. Ce modèle décrit l'évolution temporelle des événements du système à travers des nouvelles gardes temporelles. Ces gardes permettent d'abstraire les interactions entre les dynamiques continues (les variables) et les dynamiques discrètes (les événements) du système, en utilisant le logiciel PHAVer et une technique d'optimisation pour les calculer.

Les modèles des systèmes hybrides se distinguent par leur grande expressivité, leur capacité à représenter intrinsèquement les interactions entre les dynamiques continues et discrètes d'un système. Toutefois, le développement d'applications à partir des modèles hybrides, telles que la synthèse de superviseurs, de diagnostiqueur ou le model-checking peut se confronter aux résultats d'indécidabilités liés à la plupart des problèmes standards: vérification du vide, universalité, analyse d'accessibilité, déterminisation . . ., [Alur *et al*., 95]. Ces indécidabilités

nous conduisent à considérer des sous-classes de modèles hybrides, ayant des dynamiques plus simples.

En effet, dans le cadre de notre travail, nous avons choisi les systèmes à flux continu et pour cela nous pouvons utiliser une sous-classe des automates hybrides pour modéliser le système à diagnostiquer. La simplicité de la dynamique continue de ce modèle et sa capacité à représenter la dynamique d'un SDH facilitera la synthèse d'un diagnostiqueur en utilisant le modèle d'automate hybride linéaire résultant de la translation présenté dans le chapitre précédent.

Toutefois, le problème de synthèse hors-ligne d'un diagnostiqueur à partir d'un modèle automate hybride rectangulaire et le modèle d'automate temporisé (sans restrictions) est indécidable [Derbel, 09], sachant que l'automate temporisé représente un cas particulier de AHR.

C'est à partir de ces constatations que nous proposons une solution de diagnostic hors-ligne d'un système à flux continu tolérant aux défauts, modélisé par des automates hybrides linéaires. Dans notre solution, nous nous sommes inspirés d'un ensemble de techniques développées dans les travaux de Derbel [Derbel, 09] et Sampath [Sampath *et al.*, 95, 96] qui seront présentés par la suite. Elle repose sur la construction, hors-ligne, d'un AHL, appelé *diagnostiqueur*, à partir du modèle "global" d'AHL résultant de la translation du modèle RdPH élémentaire, décrivant son comportement normal et ses possibles comportements anormaux (défaillants) et en vérifiant certaines hypothèses. Le diagnostiqueur est peut être déployé, en-ligne, pour permettre l'identification des défauts affectant le système.

En effet, l'automate du diagnostiqueur reçoit les événements observables générés par le système et évolue d'une manière déterministe d'un sommet à un autre. Chaque sommet du diagnostiqueur fournit une estimation de l'état courant du système ainsi que les possibles défaillances pouvant affecter son fonctionnement. En se basant sur une analyse des données présentes dans le sommet courant du diagnostiqueur, une fonction de décision émet une alarme lorsqu'un défaut est identifié.

Dans la Figure 3.13, nous avons illustré le schéma global d'une nouvelle approche de diagnostic. Avant de présenter le cadre de cette approche, nous allons présenter quelques méthodes de diagnostic des SEDs et des SDHs les plus intéressantes pour une solution de diagnostic.

5.3. Méthodes de diagnostic des SEDs

Dans cette partie, nous présentons les méthodes de diagnostic de SED les plus intéressantes. Nous considérons une classification des méthodes de diagnostic de SED basée sur la nature du modèle. En effet, nous pouvons distinguer deux catégories de méthodes :

- les méthodes basées sur des modèles logiques ;
- les méthodes basées sur des modèles temporisés.

5.3.1. Les méthodes basées sur des modèles logiques

- élaborer le diagnostic du système à partir d'un modèle **logique** ;
- le temps sera considéré uniquement d'une manière **qualitative** à travers l'ordre d'occurrence des événements.
- Le diagnostic résultant considérera le temps, de même, d'un point de vue **qualitatif**.

Citons à titre d'exemple : l'approche de Sampath [Sampath, 95] et l'approche de Zad [Zad, 03].

5.3.2. Les méthodes basées sur des modèles temporisés

- élaborer le diagnostic du système à partir d'un modèle **Temporisé** ;
- le temps sera considéré uniquement d'une manière **explicite et quantitative** à travers l'utilisation d'horloges internes [Alur et Dill, 94] ou la discrétisation du temps [Zad *et al*., 05].

Deux types de méthodes qui se distinguent par l'élaboration ou non d'un pré-calcul de diagnostic: Les méthodes **en-ligne** ou les méthodes **hors-ligne.** Dans notre solution, on s'intéresse seulement aux méthodes hors-ligne. Ces méthodes sont caractérisées par l'élaboration d'un pré-calcul hors-ligne, sur la base d'observations connues à l'avance, afin de minimiser le calcul à effectuer pendant la phase en-ligne.

Citons à titre d'exemple : méthode à base de modèle à temps discret [Zad, 05], méthode à base de RdP temporel [Ghazel et al, 05] et méthode à base de modèle à temps continu [Derbal, 09].

5.4. Méthodes de diagnostic des Systèmes dynamiques hybrides

Nous présentons dans cette section un survol des méthodes de diagnostic des SDHs. Dans cette présentation, nous respectons une classification de ces méthodes selon : les méthodes issues des approches continues et les méthodes issues des approches SED.

5.4.1. Contributions fondées sur des approches continues

Généralement, ces méthodes se basent sur des approches de diagnostic issues de la communauté des systèmes continus.

5.4.1.1. Génération des résidus

Par exemple l'approche de [Koutsoukos *et al*., 01],

- l'estimation en ligne du mode de fonctionnement des SDH est réalisée grâce à l'emploi de RdP.
- L'occurrence d'un défaut est détectée en comparant les grandeurs mesurées à celles attendues.

L'approche de [Cocquempot *et al*., 04].

- Elle repose sur les méthodes de diagnostic à base de redondance analytique.

- Le système à diagnostiquer est modélisé par un automate hybride.
- La détection de défauts est réalisée grâce à la génération de résidus entre les variables d'entrée et de sortie mesurées
- Les relations de redondance analytique déterminées à partir des entrées et des sorties ainsi que de leurs dérivées,

L'approche de [Balluchi *et al*., 02]

- Elle utilise un observateur hybride constitué d'un observateur continu et un observateur discret ;
- L'observateur discret permet d'identifier l'état discret courant du système tandis que l'observateur continu estime l'évolution des variables continues.

5.4.1.2. Raisonnement causal.

Par exemple l'approche de [Gomaa et Gentil, 96 ; Gomaa, 97]

- Elle est basée sur les réseaux de Pétri continus causaux hybrides (RdPC^{2}H).
- RdPC^{2}H modélise de façon causale la partie continue du SDH.
- RdP classique modélisant le système de contrôle.
- RdP classique modélisant l'interaction entre la partie continue et le système de contrôle.

L'approche de [Karsai *et al*., 03].

Elle repose sur la modélisation du système par un modèle bond graph hybride [Mosterman, 97] puis la génération d'un graphe de propagation des défauts, qui permet de décrire les relations causales et temporelles entre les différents modes de défauts d'un coté, et les observations associées d'un autre.

5.4.2. Contributions fondées sur des approches SED

L'approche de [Lunze, 00, 06]

- Le problème de diagnostic est associé à un problème d'observation d'état qualitatif.
- L'abstraction qualitative des variables continues du système est effectuée à travers l'utilisation de quantificateurs.
- La localisation est une conséquence de l'identification qui associe à chaque modèle f_i un composant ou sous-système.

L'approche à base de modèle hybride à temps discret [Bhowal *et al*., 07].

- Le modèle évolue comme un automate temporisé à *n* différents flux [Alur *et al*., 93].
- modéliser les composants du système en utilisant les graphes de transition d'activités,
- obtenir le modèle global par la composition de ces modèles,
- Le diagnostiqueur est un graphe de transition d'activité, compilé hors-ligne ;
- Chaque sommet du diagnostiqueur contient une estimation de l'état du système et des étiquettes de l'ensemble $\{N, F_1,..., F_m\}$ pour indiquer la présence de défauts.

5.5. Le cadre d'une nouvelle approche de diagnostic

Dans cette partie, nous nous basons sur les travaux de Sampath [Sampath et al., 95, 96] et les travaux de Derbel [Derbel ,09], qui propose une approche de diagnostic des modèles temporisés à temps continu à base d'automates temporisés. Nous pouvons disposer d'une part, du modèle global du système à diagnostiquer décrivant les comportements événementiel, temporel et continu (changement de dynamique) et d'une autre part, d'une observation partielle des événements de ce système. Le modèle du système est "complet" dans le sens où il présente tous les comportements normaux et anormaux (défaillants) du système. Nous pouvons construire le diagnostiqueur de ce système, sous la forme d'un modèle hybride.

Dans cette approche, le diagnostiqueur consiste à inférer les occurrences des défauts non observables en se basant sur les événements observables, les délais écoulés entre ces événements ainsi que l'écart dans les dynamiques (résultant du changement des vitesses de franchissement instantanés dans RdPH élémentaire). Nous souhaitons exploiter les dynamiques continues ainsi que les gardes des transitions non observables. Nous supposons que le système à diagnostiquer est modélisé par un AHL, HA= (*Loc, x, E, δ, F, inv*). Par ailleurs, nous supposons que ce modèle respecte un ensemble de spécifications, caractérisant essentiellement la modélisation des défauts. Ces spécifications sont énumérées ci-dessous :

♦ le modèle du système est "complet", ainsi, il permet de décrire les comportements normaux et anormaux (défaillants) ;
♦ l'ensemble des événements E est réparti en deux sous-ensembles : l'ensemble des événements observables E_o et l'ensemble des événements non observables E_u ;
♦ l'ensemble E_f désigne l'ensemble des événements de défauts. Nous supposons que tous les défauts pouvant affecter le système sont non observables ; i.e., $E_f \subseteq E_u$;

♦ nous supposons que l'ensemble des défauts E_f forme une partition de m sous-ensembles disjoints et non-vides de défauts (ou modes de défaillance) $E_f = Def1 \cup .... \cup Defm$;

♦ les défauts sont supposés être permanents. En effet, le système ne peut pas retourner au comportement normal après l'occurrence d'un défaut ;
♦ deux défauts, appartenant à deux sous-ensembles distincts, ne peuvent pas se produire dans une même exécution du système. Par conséquent, le scénario de défauts multiples, qui correspond à l'occurrence multiple et successive de défauts, provenant de différents modes, n'est pas considérée dans cette proposition. Une extension de cette proposition de diagnostic supportant les défauts multiples peut être envisagée comme une perspective de notre travail ;
♦ à son état initial, nous supposons que le système est dépourvu de défauts. Ainsi, cet état fait partie de l'ensemble des états du comportement normal. Nous supposons que le modèle HA satisfait les hypothèses suivantes:

H1) HA est fortement non-zénon ;
H2) les gardes temporelles $\{x_1, x_2, ..., x_n\} \in x_D$ sont bornées par une constante entière positive K, dans chaque sommet q non-final ; i.e., $inv(q) = x_1 \leq K \wedge ... \wedge x_n \leq K$, et sont non bornées dans chaque sommet final ; i.e., $inv(q) = vrai$

L'hypothèse *H1* garantit l'existence d'une borne inférieure pour l'exécution de chaque cycle de l'AHL et par conséquent, la divergence du temps de chaque exécution. Cette hypothèse est intrinsèquement vérifiée dans les systèmes réels où la progression temps diverge.

L'hypothèse *H2* restreint la durée de séjour dans chaque sommet non final de l'AHL. Aucune restriction n'est imposée pour le séjour dans les sommets finaux. Ainsi, l'automate peut séjourner indéfiniment dans un sommet final, sans pouvoir progresser vers un autre sommet. C'est pourquoi, on peut qualifier les sommets finaux par *sommets puits*. Cette hypothèse est très importante puisqu'elle nous permet de délimiter l'espace d'état du système. Ainsi les valuations des états correspondants à des sommets non-finaux sont incluses dans un hypercube de dimension *K*. En considérant cette hypothèse, l'analyse d'atteignabilité appliquée au modèle considéré, en utilisant le PHAVer, se termine au bout d'un temps fini, en considérant que l'espace d'état définit un ensemble fini de zones de gardes temporelles.

Afin de présenter nos idées pour le diagnostic des SDHs, nous considérons, jusqu'à la fin de ce chapitre, l'exemple du système de chauffage de liquides introduit dans le chapitre 4.

5.6. La structure proposée du diagnostiqueur

Nous présentons dans cette partie la structure du diagnostiqueur que nous désirons construire à partir d'un AHL vérifiant les hypothèses et les spécifications énumérées ci-dessus. Une évolution d'un sommet à un autre dans cet automate se fait par le biais de transitions sur tout l'ensemble des événements *E*. En effet, après l'occurrence de chaque événement, le diagnostiqueur évolue vers le prochain sommet qui fournit une estimation de l'état du système, à l'instant du franchissement de cet événement et si la garde continue est satisfaite. Par ailleurs, à chacun de ces états est associée une étiquette de diagnostic permettant d'indiquer si un défaut s'est produit dans l'exécution menant vers cet état.

Nous pouvons noter par $\Theta = \{N, Def_1, Def_2, ..., Def_m\}$, un ensemble d'étiquettes, dites *étiquettes de diagnostic*. Ces étiquettes permettent de déterminer le mode de défauts affectant un ensemble d'états du système. En effet, un ensemble d'états du système est qualifié de *Def$_i$-défaillant*, s'il est associé à une étiquette *Def$_i$*. De même, il est qualifié de normal, s'il est associé à l'étiquette *N*. Sinon, aucun défaut ne s'est produit avant d'atteindre un état de cet ensemble. Un sommet du diagnostiqueur $q^d = \left(\{(q_1, \phi_1), ..., (q_k, \phi_k)\}, z\right)$ est dit :

- *Defi-certain*, si $\phi_p = Def_i$, $\forall p \in \{1, ..., k\}$.
- *Defi-incertain*, s'il existe $(q, \phi), (\tilde{q}, \tilde{\phi}) \in Dis(q^d)$, où $\phi = F_i$ et $\tilde{\phi} \neq F_i$.
- *Defi-incertain*, s'il existe $(q, \phi), (\tilde{q}, \tilde{\phi}) \in CTE(q^d)$, où $\phi = F_i$ et $\tilde{\phi} \neq F_i$.

Intuitivement, un sommet du diagnostiqueur est *Defi-certain* lorsque tous les états estimés dans ce sommet sont *Defi-défaillants*. Lorsque la fonction de diagnostic détecte que le sommet courant du diagnostiqueur est *Defi-certain*, elle génère une alarme annonçant l'occurrence d'un

défaut de l'ensemble *Defi*. Par contre, si le sommet courant est *Defi-incertain*, aucune décision ne pourra être prise par cette fonction. En effet, les états estimés comportent à la fois des états *Defi-défaillants* et d'autres états associés à un autre mode de fonctionnement (le mode de fonctionnement normal ou un mode de défaillance *Defj*, $j \neq i$) ce qui implique une ambiguïté dans la prise de décision.

Dans la suite, nous pouvons présenter le diagnostiqueur du système de chauffage de liquides, dans la Figure 5.3. Ce diagnostiqueur est obtenu à partir du modèle dynamique de la Figure 5.1. Ce dernier est construit en utilisant les résultats fournis par PHAVer et en calculant les nouvelles gardes temporelles en utilisant une technique d'optimisation.

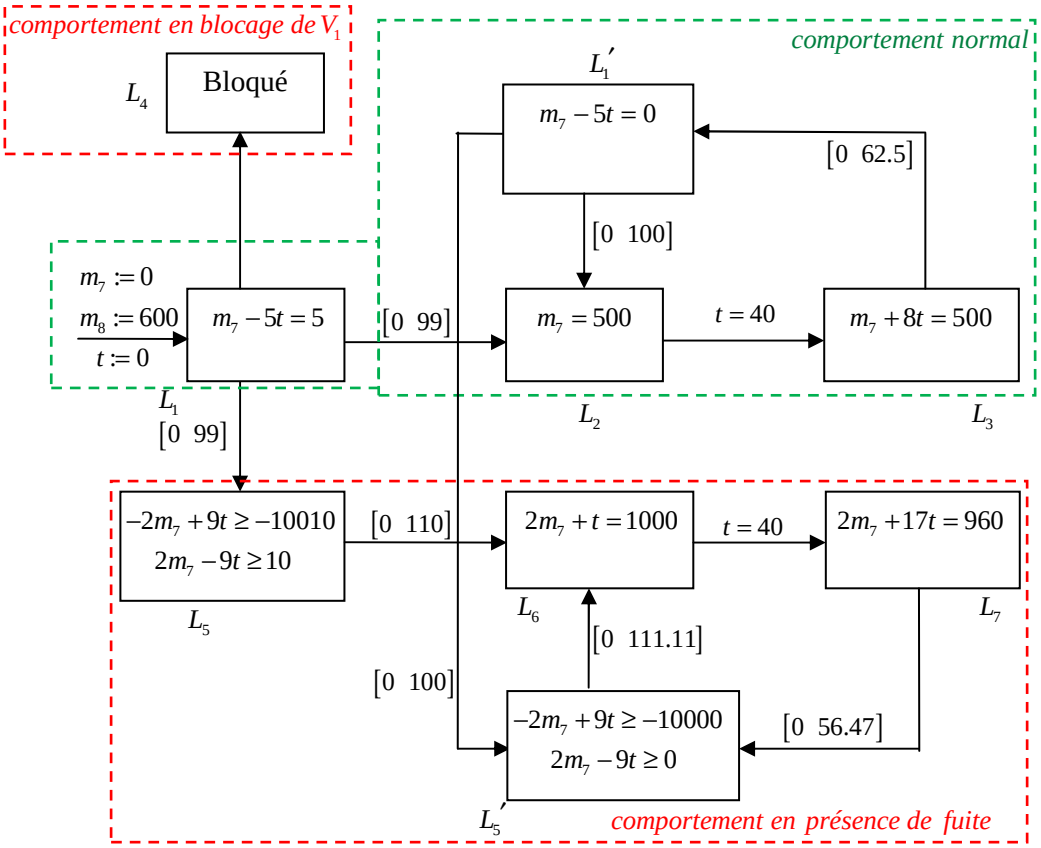


Figure 5.1. Modèle dynamique de l'AHL de la Figure 4.18

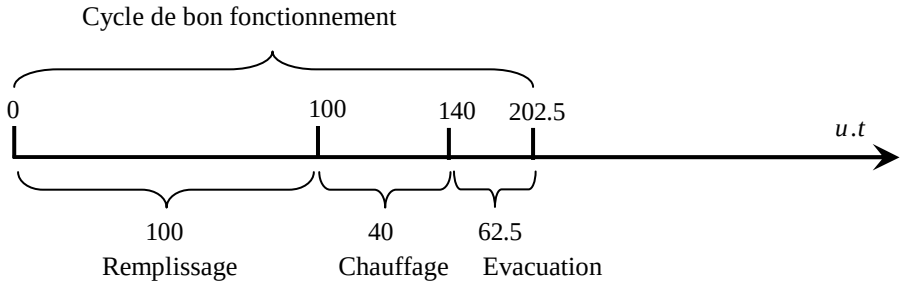


Figure 5.2. Cycle de bon fonctionnement du système de chauffage de liquides

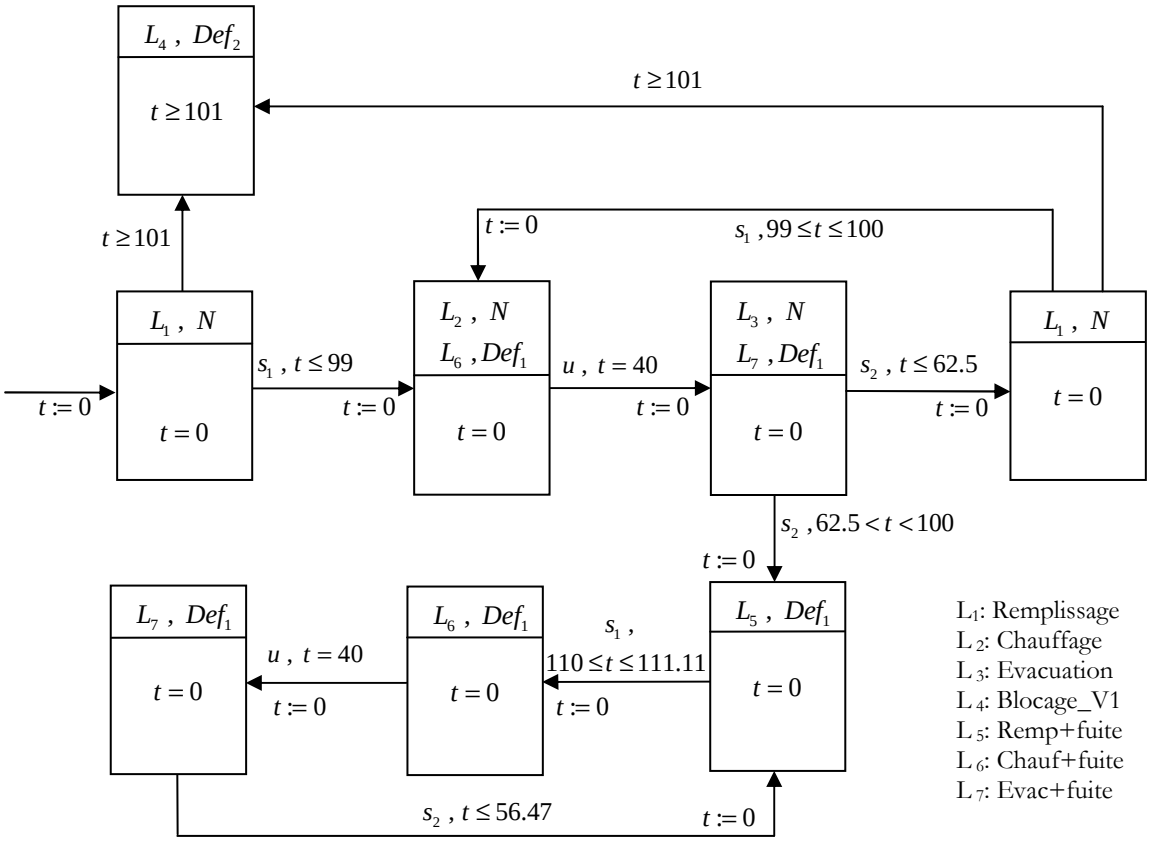


Figure 5.3. Exemple de diagnostiqueur d'un système de chauffage de liquides.

Chaque sommet de ce diagnostiqueur comporte un ensemble de paires (sommet, étiquette), associé à une contrainte sur l'horloge $\boldsymbol{x_D}^{\boldsymbol{d}}$ qui définie la valuation de cette horloge après le franchissement de la transition d'entrée du sommet.

Afin de construire ce diagnostiqueur, nous pouvons considérer la partition de défauts suivante: $E_f = Def_1 \cup Def_2$, où $Def_1 = \{\sigma_1\}$ correspond à l'occurrence d'une fuite et $Def_2 = \{\sigma_2\}$ correspond au blocage de la vanne V_1.

Nous pouvons considérer trois scénarios d'application de ce diagnostiqueur.

Scénario 1 : Nous supposons que le système effectue une exécution sur la trace: (σ_1, 5)(s_1, 95)(u, 40)(s_2, 57.5). En observant la projection observable de cette trace : (s_1, 100)(u, 40)(s_2, 57.5), le diagnostiqueur évolue vers le sommet ({(L_5 , Def_1)}, t = 0), car 57.5 est strictement inférieure que 62.5. La fonction de décision constate que le sommet est Def_1-certain et génère, par conséquence, une alarme indiquant l'occurrence d'un défaut de l'ensemble Def_1. Nous remarquons pour n'importe quelle suite d'événements observés, le diagnostiqueur va évoluer dans des sommets Def_1-certain. Ceci est attendu puisque les défauts considérés dans le modèle du système sont permanents.

Scénario 2 : Nous supposons que le système effectue une exécution sur la trace : (s_1, 100)(u, 40)(s_2, 62.5)(σ_2, 5). Le diagnostiqueur accepte la trace (s_1, 100)(u, 40)(s_2, 62.5), puis il ne reçoit plus d'événements. Lorsque 101u.t. s'écoulent après l'occurrence de l'événement s_2, une transition urgente est franchie vers le sommet $(\{(L_4, Def_2)\}, t \geq 101)$ dans le diagnostiqueur. Ensuite, la fonction de décision constate que le sommet courant est Def_2-certain puis elle génère une alarme indiquant l'occurrence d'un défaut de l'ensemble Def_2.

Scénario 3 : Nous supposons que le système effectue une exécution sur la trace: (σ_2, 5) (s_1, 100)(u, 40)(s_2, 62.5). Le diagnostiqueur **n'accepte pas** la trace (s_1, 100)(u, 40)(s_2, 62.5), car il ne reçoit plus l'événement s_1 après 101u.t. écoulée. Le diagnostiqueur évolue rapidement vers le sommet $(\{(L_4, Def_2)\}, t \geq 101)$, alors la fonction de décision constate que le sommet courant est Def_2-certain puis elle génère une alarme indiquant l'occurrence d'un défaut de l'ensemble Def_2.

Plusieurs perspectives peuvent être envisagées, à court terme ou à long terme, comme suite au travail présenté dans cette thèse. Nous les avons regroupées selon trois axes :

Dans un premier axe, nous envisageons introduire un algorithme de synthèse de diagnostiqueur suivant la structure que nous avons proposé dans ce chapitre. Pour se faire, nous utiliserons les différentes bibliothèques proposées dans la littérature, permettant la détection, la localisation et l'identification des défauts sur le système. Une étude de la performance des algorithmes développés sera ensuite établie afin de permettre de choisir l'implémentation en ligne du diagnostiqueur, la plus optimale, en termes d'évolution continue.

Dans un second axe, nous envisageons de prendre en considération la mesurabilité de certaines variables continues au cours de notre démarche de diagnostic des SDHs. L'approche de diagnostic pour les SDHs présentée dans [Derbel, 09] repose uniquement sur l'observation d'une trajectoire d'événements discrets temporisés générée par le système. Cependant, certaines variables telles que la température ou le débit peuvent être directement mesurées en utilisant des capteurs continus. L'abstraction de ces mesures par l'utilisation de quelques événements discrets peut conduire à des inférences erronées. En effet, une perspective de notre travail consiste à combiner notre solution avec l'approche de Lunze (Lunze, 2006) qui repose sur une fine discrétisation des trajectoires continues. Ainsi, la solution à développer doit répondre à ce compromis: minimiser la perte d'informations engendrée par la discrétisation des trajectoires continues d'un coté, et éviter le problème d'explosion combinatoire dû à cette opération d'un autre.

Dans un dernier axe, nous proposons de combiner l'approche de diagnostic à base d'AHL avec la théorie du contrôle supervisé développée par Ramadge et Wonham [Ramadge et Wonham, 1987]. Ce travail peut être envisagé dans le cadre d'une extension de l'approche proposée dans [Sampath et al., 1998]. Il s'agit de proposer une solution intégrée qui permet de coupler le module du diagnostic avec le module du contrôle. Cette solution repose sur la

synthèse d'un superviseur qui permet d'autoriser ou d'inhiber certains événements contrôlables du système afin de déterminer le plus grand sous-langage temporisé diagnosticable.

5.7. Conclusion

Nous avons présenté au cours de ce chapitre conclusif le principe de la partie complémentaire pour notre approche de modélisation des SDHs à flux continu tolérants aux défauts. Les idées ainsi que les perspectives de cette partie introduisent l'élaboration d'une nouvelle approche de diagnostic des SDHs à base d'AHL.

Dans un premier volet, nous avons présenté quelques contributions dans le cadre de diagnostic des SEDs et SDHs les plus pertinentes pour notre proposition dans ce chapitre. Par la suite, nous avons introduit les premières démarches d'une nouvelle approche de diagnostic. Le système à diagnostiquer est modélisé par un modèle global d'AHL décrivant à la fois son comportement normal et ses comportements défaillants. Ce modèle doit respecter certaines spécifications et hypothèses indispensables pour la construction du diagnostiqueur.

Dans un deuxième volet, nous avons présenté, intuitivement, la structure proposée du diagnostiqueur en basant sur l'analyse d'atteignabilité sous le logiciel PHAVer pour avoir le modèle dynamique. Ce diagnostiqueur est un AHL déterministe qui peut évoluer en fonction des événements observables générés par le système et estime l'état courant du système et les occurrences de défauts. Une fonction de décision, peut analyser plus tard le sommet courant du diagnostiqueur et annonce l'occurrence d'un défaut du mode *Defi* lorsque ce sommet est *Defi-certain*. Enfin, nous avons terminé ce chapitre par nos perspectives de recherche.

CONCLUSION GÉNÉRALE

L'objectif de cette thèse était la surveillance et le diagnostic d'une sous-classe des systèmes dynamiques hybrides (SDH) tolérants aux défauts en proposant une approche de diagnostic permettant la détection, la localisation et l'identification des défauts qui affectent un système. Nous nous sommes basés sur un ensemble de travaux dans la littérature tel que : [Sampath et al., 95, 96], [Ghazel, 05] et [Derbel, 09] pour l'élaboration de cette approche.

Dans un premier chapitre, nous avons commencé par une présentation détaillée du cadre général des SDH. Nous avons présenté la notion des SDH allant de la caractérisation de ces derniers à la description et la classification des aspects hybrides. Par la suite, nous avons entamé les approches et les outils de modélisation et nous avons discuté des outils de simulation et analyse de ce type de systèmes. Notons que cette étape, d'une part, a présenté les outils facilitant la lecture des chapitres qui la suivent et d'autre part elle a justifié le choix des Automates Hybrides Linéaires et les Réseaux de Pétri Hybrides élémentaires comme outils de modélisation. Les SDHs englobent une large classe de systèmes dynamiques, mais dans ce travail, nous avons limité le contenu de ce point à une sous-classe importante fréquemment rencontrée dans la réalité. Ainsi, nous avons ciblé la sous-classe des systèmes à commutations contrôlées, à savoir les systèmes à flux continus.

Les systèmes réels sont devenus de plus en plus complexes, cette croissance de la complexité rend ces systèmes vulnérables aux défaillances, justifiant ainsi l'introduction de modules de surveillance. Cela nous a conduit, dans le deuxième chapitre, de présenter le concept général de la surveillance des systèmes dynamiques allant de quelques définitions et terminologies de base à la fonction de surveillance, son principe et ces sous-fonctions. Par la suite, deux grandes classes des méthodes de surveillance ont été présentées, à savoir les méthodes sans modèles, et les méthodes à base de modèles et nous avons présenté quelques approches de surveillance des SDH afin de situer le chemin de surveillance que nous avons pris.

Le troisième chapitre représente notre première contribution dans cette thèse, nous avons élaboré une méthode de modélisation structurelle des SDH à flux continu tolérants aux défauts, permettant ainsi la représentation et la détection des défauts dans ce type de systèmes. Nous avons présenté les notions utilisées, à savoir les approches de modélisation des systèmes dynamiques hybrides à flux continu tolérant aux défauts et les techniques de modélisation des défauts. En nous basant sur les avantages des réseaux de Pétri hybrides (RdPH) et les automates hybrides (AH), nous avons proposé de coupler la puissance d'analyse des AH à la puissance de modélisation des RdPH, cela permet d'avoir une approche combinant les avantages des deux modèles. Leur association a été réalisée en effectuant une translation structurelle du RdPH élémentaire en présence des défauts vers l'AH linéaire correspondant. Cette translation a été détaillée dans le quatrième chapitre de cette thèse et représente notre deuxième contribution. Pour tirer profit de chacun des deux modèles, nous avons introduit, dans ce chapitre, un modèle bisimilaire au modèle RdPH élémentaire : l'AH linéaire. Nous avons présenté, pour ce faire, une méthode structurelle permettant de translater le RdP en automate. Un algorithme permettant de garantir la translation de manière systématique a été établi. Une preuve mathématique pour cette translation a également été présentée. Les idées fournies dans ce chapitre ont été illustrées à

travers un exemple simple, l'automate résultant a été analysé en utilisant le logiciel PHAVer et a permis de construire l'automate dynamiquement atteignable.

Dans le dernier chapitre, nous avons introduit le cadre d'une approche de diagnostic. Le système à diagnostiquer est modélisé par un modèle global d'AH linéaire décrivant à la fois son comportement normal et ses comportements défaillants. Ce modèle doit respecter certaines spécifications et hypothèses indispensables pour la construction du diagnostiqueur. Ainsi, l'approche proposée permet l'identification des défauts qui s'expriment par un changement des dates d'occurrence des événements observables. La solution présentée dans notre travail repose sur la construction hors-ligne d'un diagnostiqueur.

Les perspectives de notre recherche que nous l'avons présenté dans le chapitre conclusif restent juste des idées préliminaires pour nos prochain travaux.

ANNEXE A

Décidabilité et Non Décidabilité des Problèmes Classiques Avec les Automates Hybrides

D'une manière générale, le problème d'accessibilité est un problème non décidable [Alur *et al.*, 95], [Alur *et al.*, 93] pour les automates hybrides. Ceci peut être expliqué par la puissance de modélisation du formalisme qui dépasse celle de la machine à deux compteurs[1] dont le problème d'arrêt est déjà indécidable. Á l'aide de deux variables et une horloge, nous pouvons facilement simuler les trois instructions de base de la machine à deux compteurs. Ainsi, il est important de déterminer des sous classes du modèle de base, dont le problème d'accessibilité est décidable. Cependant, il faut s'assurer que la sous classe décidable ne soit pas trop restrictif pour avoir une puissance de modélisation suffisante pour les systèmes à modéliser.

Une sous-classe décidable intéressante est donnée par les automates temporisés [Alur et Dill, 94]. Les problèmes classiques de vérification avec les automates temporisés peuvent être également exprimés à travers les langages temporisés. A titre d'exemple, le problème d'accessibilité d'un état peut être ramené au test du vide d'un langage temporisé d'un automate temporisé[2].

Ainsi, pour vérifier une propriété d'exclusion mutuelle entre deux processus, il faut vérifier que l'état *l* où les deux processus sont à leurs sections critiques respectives soit inaccessible. De la même façon, nous pouvons définir le langage temporisé *L* correspondant à l'automate temporisé où *l* soit final. Le test d'accessibilité de *l* revient à effectuer le test de vacuité sur le langage *L*. Ainsi, le problème du vide d'un langage[3] et le problème d'accessibilité d'un état sont équivalent; la décidabilité de l'un induit la décidabilité de l'autre.

Le problème d'accessibilité est décidable pour la classes des automates temporisés à contraintes simples[4] [Alur et Dill, 94] ou diagonales[5] [Bérard *et al.*, 98] en PSPACE. Le calcul de l'espace atteignable se fait par l'automate des régions [Alur et Dill, 94] qui illustre les différents espaces atteignables dans les sommets de l'automate.

Le calcul de l'espace atteignable peut être effectué à travers les automates des régions proposés dans [Alur et Dill, 94], [Alur, 99]. Ce formalisme regroupe les différentes valeurs possibles des variables dans les sommets sous forme de polyèdres. Cette construction peut être infinie si les variables divergent.

L'automate temporisé constitue l'une des classes les plus restrictives des automates hybrides. Les problèmes qui sont non décidables pour cette classe le seront pour des classes moins restrictives. Il est ainsi intéressant de dégager les problèmes indécidables pour cette classe d'automate.

---

[1] La machine à deux compteurs (machine de Minsky) est définie dans [El Touati, 13]

[2] Pour un automate temporisé *A* avec un ensemble de sommets *L* et un sous ensemble de sommets finaux $L_F \subset F$, un langage temporisé de *L(A)* est l'ensemble des mots temporisés qu'on peut générer à partir du sommet initial dans l'un des sommets finaux.

[3] C'est à dire, le problème de tester le vide d'un langage temporisé accepté par un automate temporisé.

[4] De la forme $x \sim c$ avec $\sim \in \{<, \leq, \geq, >\}$.

[5] De la forme $x \sim c$ ou bien $x - x' \sim c$ avec $\sim \in \{<, \leq, \geq, >\}$.

La classe des automates à multi-vitesse initialisés (multirate initialized automata) forme également une classe décidable [Alur *et al.,* 93], [Nicollin *et al.*, 93], [Henzinger et Rusu, 98]. Cette classe autorise des dérivées constantes différentes et **non nulles** et avec des initialisations lors du changement de la dynamique d'une variable suite à l'exécution d'une transition.

Les automates hybrides rectangulaires initialisés [Henzinger et Rusu, 98] constituent aussi une classe décidable pour le problème d'atteignabilité. Cependant, le problème d'atteignabilité est non décidable pour les automates rectangulaires d'une manière générale [Puri et Varaiya, 94].

Il est prouvé dans [Alur *et al.*, 95] qu'il est possible de simuler une machine à deux compteurs avec un automate hybrides ayant cinq horloges et deux variables ayant des dérivées (pentes) distinctes. Ainsi, le problème d'accessibilité est non décidable pour un automate admettant deux dérivées distinctes et des contraintes diagonales. Les auteurs ont également simulé la machine de Minsky par un automate hybride à cinq horloges et une variable de pente non nulle et différente de un (skewed variable, en anglais).

Dans [Cerans, 92], l'auteur prouve l'indécidabilité du problème d'atteignabilité pour un automate temporisé admettant au moins trois chronomètres (stopwatches, en anglais). La non décidabilité est également prouvée pour un automate temporisé ayant une variable de dérivée nulle et des contraintes diagonales. Un autre résultat de non décidabilité est donné dans [Alur *et al.,* 93] pour un automate temporisé avec six variables de pentes nulles sans contraintes diagonales.

Dans [Kesten *et al.,* 92], [Kesten *et al.,* 99], les auteurs étudient une classe d'automates hybride appelé CSHA (Constant Slope Hybrid Automata, en anglais). Elle se caractérise par des dérivées constantes entières et des gardes sous forme de combinaisons booléennes d'inéquations à coefficients entiers. Les auteurs ont proposé une sous classe intéressante dont le problème d'accessibilité est décidable: les graphes d'intégration (Inegration Graphs, en anglais). Une version étendu et décidable du graphe d'intégration est proposé dans [Bouajjani et Robbana, 95]. Dans [Bouajjani *et al.*, 94], les auteurs montrent la décidabilité d'un graphe temporisé avec un seul chronomètre.

Dans [Asarin *et al.,* 12], les auteurs considèrent une classe particulière des AHL ayant des trajectoires continues (c'est dire l'affectation correspond à l'identité) qu'ils appellent PCD (Piecewise Constant Derivative, en anglais). Les auteurs prouvent que le problème d'atteignabilité est décidable pour les PCD de dimension 2. Cependant, le problème devient non décidable pour les PCD de dimension supérieur ou égale à trois. Les auteurs définissent une classe intermédiaire entre les PCD de dimension deux et trois appelé HPCD (Hierarchical PCD) tolérant des grades comparatives. Le problème d'atteignabilité reste ouvert pour cette classe.

ANNEXE B

L’outil PHAVer

Cette annexe est consacrée à logiciel PHAVer (Polyhedral Hybrid Automaton Verifier). PHAVer est un outil de vérification des systèmes dynamiques hybride. Il est développé par Goran Frehse, du laboratoire Verimag de Grenoble–France. Il présente beaucoup de similitudes avec l'outil HyTech [Henzinger *et al.*, 97] développé à l'université Berkeley aux Etats-Unis. Nous allons par la suite présenter La syntaxe de PHAVer et ses points de similitudes avec HyTech.

I. Présentation de PHAVer

PHAVer est un outil pour la vérification de propriétés de sureté pour les systèmes dynamiques hybrides linéaires par morceaux. PHAVer utilise une arithmétique exacte dont la robustesse est garantie par l'utilisation de la bibliothèque *Parma Polyhedral Library* [Bagnara *et al.*, 02]. La vérification des propriétés de sureté pour les systèmes dynamiques hybrides est ramenée à un problème de calcul d'atteignabilité, qui n'est décidable que pour une sous-classe des automates hybrides dite automates hybrides rectangulaires initialisés. PHAVer utilise un algorithme à la volée qui donne une sur-approximation des dynamiques affines par des automates hybrides linéaires. Un ensemble d'algorithmes a été développée pour réduire le nombre de bits et le nombre de contraintes qui sont nécessaires pour représenter les régions polyédrique et améliorer l'efficacité globale de l'algorithme de vérification. PHAVer a aussi la capacité de calculer les relations de simulation et de décider de l'équivalence et de raffinement entre automates hybrides

II. Syntaxe de PHAVer

PHAVer utilise les automates hybrides à entrées/sorties [Frehse, 05] (hybrid Input/Output automata). Dans un automate hybride à entrées/sorties l'ensemble des variables continues est scindé en trois sous-ensembles. $X = X_I \cup X_O \cup X_L$, tel que X_I est l'ensemble des variables d'entrées, X_O est l'ensemble des variables de sortie et X_L est l'ensemble des variables locales.

Ces sous-ensembles sont deux à deux disjoints. $X_I \cap X_O = X_I \cap X_L = X_O \cap X_L = \phi$. Les variables appartenant à X_L où à X_O sont les variables contrôlées par l'automate, tandis que les variables appartenant à X_I sont des variables non-contrôlées.

La caractéristique principale de PHAVer est qu'il différencie entre les variables d'entrée (**input_var**) et les variables de contrôle (**contr_var**). Ce qui est très important pour la vérification d'équivalence et de relation de simulation.

La syntaxe qu'utilise PHAVer pour la description textuelle des automates hybrides est similaire à celle de HyTech. La structure générale de la description d'un automate hybride est comme suit :

```
automaton automate
   contr_var: var 1, var 2,... ;
   input_var: var 3, var 4,... ;
   parameter: var 5, var ident6,... ;
   synclabs: lab_ident1, lab_ident2,... ;
   loc Sommet_1: while invariant wait {dynamique};
      when guard_1 sync label do {initialisation} goto Sommet_2;
      when ...
   loc Sommet_3: while ...
end
```

III. Structure de données

Il y a quatre types de structures de données qui peuvent être affectées aux identificateurs : formules linéaires, ensembles d'états symboliques, relations symboliques et automates.

♦ *Formules linéaires :* Elles sont spécifiées sur une collection de variables, nombre et constantes qui peuvent être combinés en utilisant +, -, /, *, (et). tant que la combinaison est linéaire sur les variables.

♦ *Ensemble d'états symboliques :* Un état symbolique est une combinaison d'un sommet et d'une formule linéaire, unie par &. *e.g.* **S1 & m1+m2 == 20 & t1 >0**. Un ensemble d'états symboliques d'un automate **aut** est affceté à une variable par la formule

identificateur = aut.{ensemble d'etat symboliques}.

♦ *Relation symbolique :* Elles sont obtenues par les algorithmes de simulation.

♦ *Automates :* L'affectation d'un automate à l'identificateur **aut** se fait par description de tous ces paramètres qui sont :

Variables : Toutes les variables doivent être déclarées à l'aide des instructions **state_var** (pour les variables contrôlées) et **Input_var** (pour les variables d'entrées). Notons que l'instruction **contr_var** n'est utilisée que depuis la version 0.35 et que le mot **state_var** était utilisé dans les versions antérieures.

Sommets : La déclaration des sommets est effectuée par l'instruction **loc**. L'invariant est une expression linéaire qui combine les variables d'entrées, les variables de sortie et les constantes. La définition des pentes de variables dépend des dynamiques :

♦ Pour une dynamique linéaire, c'est une formule linéaire entre les variables d'état.

e.g., **0 ≤ m1' & m1' ≤ 10** pour $\dot{m}1 \in [0\ 10]$.

♦ Pour les dynamiques affines, c'est une formule linéaire de la forme entre les variables d'état et leurs dérivés. *e.g.*, m1' == 2*m1 pour $\dot{m}1=3m1$,

Transitions : Une transition est spécifiée par les instructions **when** …**goto**. On doit avoir toujours une étiquette de synchronisation associée à la transition. Une formule linéaire **Initialisation** spécifiées relation de saut après le mot **do**. Les variables d'état qui ne changent pas de valeur lors du franchissement de la transition doivent être spécifiées explicitement par la relation de type **x' == x**, et les variables changeant de valeur pas la relation **x' == x0**, avec x0 constant.

PHAVer dispose d'un ensemble de commandes pour l'analyse des automates hybrides.

IV. Commande de PHAVer

Les principales commandes de PHAVer sont présentées ci-après :

- **& :** L'ampersand est utilisé pour la composition d'automates. *e.g.*, **Aut** = **Aut1** & **Aut2**
- **.reachable :** calcule l'ensemble des états atteignable par l'automate depuis ses états initiaux, et en faisant une analyse avant. Ainsi l'instruction **Atteignable** = **Aut.reachable** affecte l'espace d'état atteignable par l'automate **Aut** à la variable **Atteignable**.
- **.print('fichier', arg) :** écrit une description de l'automate dans le fichier **'fichier'**. le mot clé **arg** dénote le format du fichier. Elle peut prendre les valeurs 0, 1 et 2.
- **.reverse :** Inverse la causalité d'un automate. Cette commande peut être utilisée pour l'analyse en arrière d'un automate, en l'inversant puis en effectuant son analyse avant.
- **.intial_states :** remplace les états initiaux de l'automate.
- **.difference_assign :** fait la différence entre deux espaces d'états.

PHAVer donne la possibilité de représenter l'espace d'état atteignable sur des graphes bidimensionnels. Sous Linux on utilise la commande **graph** du logiciel **plotutils**, disponible sur http://www.gun.org/software/plotutils/. Sous windows/Cygwin il existe un script matlab, disponible à http://www.es.ru.nl/~goranf/.

RÉFÉRENCES BIBLIOGRAPHIQUES

A

[Alla *et al.*, 92]: Alla H., Cavaille J.B., Le Bail J., Bel G. Les systèmes de production par lot: une approche discretcontinuutilisant les réseaux de Petri hybrides, Automation of Mixed Processes (ADPM 92) Janvier, Paris, France, 1992.

[Alla et David, 97]: Alla. H, David. R.'' Du Grafcet Aux réseaux de Petri''. Hermès. (Ed). Paris. 1997

[Alla et David, 98] : Alla H. and David R., A modeling and analysis tool for discrete event systems: Continuous Petri net, Perform. Evaluation 33(3) 175, 1998.

[Alla et David, 98]: Alla, H. et R. David. Continuous and hybrid petri nets, Journal of Circuits, Systems and Computer, vol. 8, no 1, p. 159 – 188, 1998.

[Alla, 94]: Alla H., Les réseaux de Petri: un outil particulièrement adapté à la modélisation des systèmeshybrides, Automation of Mixed Processes (ADPM 94), p.17-25, Novembre, Bruxelles, Belgique, 1994.

[Allam, 98] : Allam M., Sur l'analyse quantitative des réseaux de Petri hybrides : Une approche basée sur les automates hybrides. Thèse d'état de l'inpg, Institut Nationale Polytechnique de Grenoble, 1998.

[Alur *et al.*, 03]: Alur R., Ivancic F. and Dang T., Progress on reachability analysis of hybrid systems using predicate abstraction. In O. Maler and A. Pnueli, editors, Hybrid Systems: Computation and Control: 6th International Workshop, HSCC 2003, Prague, Czech Republic, LNCS 2623, pages 4–19. Springer, April 2003.

[Alur *et al.*, 93]: Alur R., Courcoubetis C., Henzinger T.A., and Ho P.H.. Hybrid automata : an algorithmic approach to the specification and verification of hybrid systems. In Hybrid Systems, LNCS, 736 :209–229, 1993.

[Alur *et al.*, 95]: Alur R., Courcoubetis C., Halbwachs N., Henzinger T.A., Ho P.H., Nicollin X., A. Olivero, Sifakis J. and Yovine S., The algorithmic analysis of hybrid systems. Theoretical Computer Science, vol. 138, pp. 3-34, 1995.

[Alur et Dill, 94] : Alur, R. et D. L. Dill. 1994, «A theory of timed automata», Theoretical Computer Science, vol. 126, p. 183−235.

[Alur, 99]: Alur. R. ''Timed Automata''. NATO-ASI 1998 Summer School on Verification of Digital and Hybrid Systems. A revised and shorter version appears in 11th International Conference on Computer-Aided Verification, LNCS 1633, pp. 8-22, Springer-Verlag. 1999.

[Antsaklis *et al.*, 93]: Antsaklis, P. J. Lemmon, M. D. and Stiver, J. A. Hybrid system modeling and event identification. Technical report, Technical Report of the ISIS Group at the University of Notre Dame ISIS-93-002, Notre Dame, January 1993.

[Antsaklis, 00]: Antsaklis. P. J., ''Special issue on hybrid systems: Theory and applications, a brief Introduction to the Theory and Applications of Hybrid Systems''. Proceeding of the IEEE. Vol. 88. N°. 7. pp. 879-889. July 2000.

[Asarin *et al.*, 00]: Asarin E., Bournez O., Dang T., and Maler O. Approximate reachability analysis of piecewise linear dynamical systems. In Hybrid Systems: Computation and Control, LNCS, 1790:20–31, 2000.

[Asarin *et al.*, 07]: Asarin. E, Gordon.P, Schnieder. G. ''Algorithmic Analysis of Polygonal Hybrid Systems, Part II: Phase Portrait and Tools''. Elsevier Science.2007.

[Asarin *et al.*, 12]: Eugene Asarin, Venkatesh P. Mysore, Amir Pnueli, and Gerardo Schneider. Low dimensional hybrid systems ? decidable, undecidable, don't know. Information and Computation, 211(0) :138– 59, 2012.

[Asarin et Dang, 04]: Asarin. E, Dang. T. '' Abstraction by Projection and Application to Multi-affin Systems''. HSCC'04. Hybrid systems: Control and Computation. 2004.

[Aswani et Tomlin, 07]: Aswani. A, Tomlin. C. ''Reachability Algorithm for Biological Piecewise Affine Hybrid Systems''. In Hybrid systems: Computer and Control. 2007.

B

[Bagnara *et al.*, 02]: Bagnara. R., E. Ricci, E. Zaffanella, et P. M. Hill, "Possibly not closed convex polyhedra and the Parma Polyhedra Library," 2002.

[Balaguer et Chatain, 12]: Sandie Balaguer, Thomas Chatain, and Stefan Haar. A concurrency-preserving translation from time Petri nets to networks of timed automata. Formal Methods in System Design, 40(3) :330–355, june 2012.

[Balluchi *et al.*, 02] : Balluchi, A., L. Benvenuti, M. D. D. Benedetto et A. L. Sangiovanni-vincentelli, «Design of observers for hybrid systems», dans Proceedings of Hybrid Systems : Computation and Control, volume 2289 of LNCS, Springer-Verlag, p. 76−89, 2002.

[Basseville, 88]: Basseville, M. «Detecting changes in signals and systems - a survey», Automatica, vol. 24, no 3, p. 309–326, 1988.

[Bemporad et Morari, 99]: Bemporad, A. and Morari, M. Control of systems integrating logic, dynamics, and constraints. Automatica, vol. 35(3): p. 407–428, 1999.

[Bérard *et al.*, 08]: Bérard B., Cassez F., Haddad S., Lime D., and Roux O.H. Comparison of expressiveness for timed automata and time Petri nets. In Vangelis Th. Paschos, editor, Combinatorial Optimization and Theoretical Computer Science, pages 93–144. ISTE Publishing / John Wiley, JAN 2008.

[Bérard *et al.*, 98]: B. Bérard, A. Petit, V. Diekert, and P. Gastin. Characterization of the expressive power of silent transitions in timed automata. Fundamenta Informaticae, 36(2,3) :145–182, August 1998.

[Bergman et Larsson, 98] : Bergman N. and Larsson M., Fault detection and isolation in the water tank world. In Proc. First Conference on Computer Science and Systems Enginering, Linköping, Sweden, Mar 1998. ECSEL.

[Berthomieu *et al.*, 06] : Berthomieu, F. Peres, and F. Vernadat. Bridging the gap between timed automata and bounded time petri nets. Lecture Notes in Computer Science: Formal Modeling and Analysis of Timed Systems, Volume 4202, pp. 82–97, 2006.

[Bhowal *et al.*, 07] : Bhowal, P., D. Sarkar, S. Mukhopadhyay et A. Basu. «Fault diagnosis in discrete time hybrid systems - a case study», Information Sciences, vol. 177, N°5, p. 1290−1308, 2007.

[Biswas *et al.*, 03] : Biswas G., Simon G., Mahadevan N., Narasimhan S., Ramirez J., and Karsai G., A robust method for hybrid diagnosis of complex systems. In Frits W. Vaandrager and Jan H. van Schuppen, editors, In Proc. of the 5th Symposium on Fault Detection,Supervision and Safety for Technical Processes, pages 1125–1131, June 2003.

[Bonhomme, 01]: Bonhomme P., Réseaux de Petri P-Temporels : Contribution a la Commande Robuste. Thèse pour obtenir le grade de Docteur préparé à l'université de Savoie 12 juillet 2001.

[Bornot *et al.*, 98]: Bornot S., Sifakis J., and Tripakis S., Modeling urgency in timed systems. In Willem-Paul de Roever, Hans Langmaack, and Amir Pnueli, editors, Revised Lectures of the 1st International Symposium on Compositionality: The Significant Difference (COMPOS'97), volume 1536 of Lecture Notes in Computer Science, pages 103–129. Springer-Verlag, 1998.

[Bouajjani *et al.*, 94]: A. Bouajjani, R. Echahed, and R. Robbana. Verifying invariance properties of timed systems with duration variables. In Hans Langmaack,Willem-Paul Roever, and Jan Vytopil, editors, Formal Techniques in Real-Time and Fault-Tolerant Systems, volume 863 of Lecture Notes in Computer Science, pages 193–210. Springer Berlin Heidelberg, 1994.

[Bouajjani et Robbana, 95]: A. Bouajjani and R. Robbana. Verifying ?-regular properties for a subclass of linear hybrid systems. In Pierre Wolper, editor, Computer Aided Verification, volume 939 of Lecture Notes in Computer Science, pages 437–450. Springer Berlin Heidelberg, 1995.

[Boufaied, 03]: Boufaied, A. Contribution à la surveillance distribuée des systèmes à événements discrets complexes, thèse de doctorat, L'université Paul Sabatier de Toulouse, 2003.

[Boyer, 01]: Boyer M., "Contribution à la modélisation des systèmes à temps contraint et application au multimédia. Thèse de doctorat, Université Toulouse 3, 2001.

[Brams, 82]: Brams G.W., "Réseaux de Petri : Théorie et Pratique", Tome 1, Théorie et analyse, Masson, 1982.

[Brams, 83]: Brams G.W., "Réseaux de Petri : Théorie et Pratique", Tome 2, modélisation et applications, Masson, 1983.

[Branicky, 93]: Branicky. M. S. ‘’Asymptotic stability of m-switched systems using Lyapunov-like functions’’. *LIDS Tech. Report, 2214. 1993.*

[Branicky, 94]: Branicky. M.S. ‘’Stability of Switched and Hybrid Systems’’. *Proceeding of the 33rd Conference On decision and Control. 19941994 American Control Conf.,* pp. 3110-3114, Baltimore, June 1994.

[Branicky, 95]: Branicky M.S. Studies in hybrid systems: Modeling, Analysis and Control. PhD thesis, MIT, Massachusetts, USA, 1995.

[Branicky, 96]: Branicky. M. S., ‘’Studies in Hybrid Systems: Modeling, Analysis, and Control’’. PhD Thesis, Massachusetts Institute of Technology. 1996.

[Branicky, 97]: Branicky. M. S., ‘’Stability of Hybrid Systems: State Of the Art’’. *Proceeding of the 36th Conference On Decision and Control. 1997.*

[Branicky, 98]: Branicky. M. S., Multiple Lyapunov Functions and Other Analysis Tools for Switched and Hybrid Systems’’. *IEEE Trans. Automatic Control*, 43(4):475-482. *April, 1998.*

[Buisson et Lu, 94]: Buisson. J. et LU. Y, ‘’Analyse des Systèmes Hybrides avec Les Bond-Graphs.Modes Impulsionnels et Formulation Implicite’’. ADPM’94. pp. 77_82. 1994.

C

[Cassez et Roux, 03] : Cassez F. and Roux O.(H.), Traduction structurelle des réseaux de Petri temporels vers les automates temporisés. In 4ème Colloque sur la Modélisation des Systèmes Réactifs (MSR 03), Metz, France, octobre 2003.

[Cassez et Roux, 04] : Cassez F. and Roux O.(H.), Structural translation from time Petri nets to timed automata. In Electronic Notes in Theoretical Computer Science: Fourth International Workshop on Automated Verification of Critical Systems (AVoCS’04), London (UK), Elsevier, September 2004.

[Cerans, 92]: K. Cerans. Algorithmic problems in analysis of real-time systems specifications. PhD thesis, University of Latvia, 1992.

[Cerone et Maggiolo-Schettini, 99]: Cerone A. et Maggiolo-Schettini A., Time-based expressivity of time Petri nets for system specification". Theoretical Computer Science, 216. Elsevier, 1999.

[Champagnat, 97]: Champagnat. R. ‘’Modeling Hybrid Systems by Meand of High_level Petri Nets: Benefits and limitations’’. Commande des Systèmes Industriels (CIS). Vol. 1. pp. 469_475. 1997.

[Chombart, 96]: Chombart A., Flaus J.M., Valentin-Roubinet C. Hybrid Systems Modelling: a comparison of three methods applied to an example. Congrès IFAC’96, 30 Juin-5 Juillet, San Francisco, USA, 1996.

[Chouikha et Schnieder, 98]: Chouikha. M, Schnieder. E. ‘’Modelling of Continuous-discrete Systems with Hybrid Petri Nets’’. Proceeding of CESA Computational Engineering in Systems. 1998.

[Chow et Wilsky, 84]: Chow, E. et A. Wilsky. «Analytical redundancy and the designe of robust failure detection system», IEEE transactions on Automatic and Control, vol. 29, n° 7, p. 603–614, 1984.

[Cocquempot *et al*., 04]: Cocquempot, V., T. E. Mezyani et M. Staroswieckiy, «Fault detection and isolation for hybrid systems using structured parity residuals», dans Asian Control Conference, ASCC’04, vol. 2, New Mexico, p. 1204−1212, 2004.

[Cocquempot, 04] : Cocquempot Vincent, "Contribution à la surveillance des systèmes industriels omplexes", Habilitation à Dériger des Recherches, Laboratoire d’Automatique et de

Génie Informatique et Signal de Lille—LAGIS UMR 8146, France, 10 novembre 2004.

[Coholahan et Roussopoulos, 83]: Coholahan J.E. et Roussopoulos N., "Timed requirements for timed driven systems using augmented Petri nets", IEEE Transactions in Software Engineering, 9. IEEE Computer Society, 1983.

[Combacau *et al*., 00] : Combacau, M., P. Berrut, F. Charbonnaud et A. Khatab. «Refexions sur la terminologie : Surveillance - supervision», Groupement pour la recherche en Produtique, Systèmes de Production Sûrs de Fonctionnement, 2000.

[Combacau, 91] : Combacau, M. Commande et surveillance des systèmes à événements discrets complexes : applications aux ateliers flexibles, thèse de doctorat, L'université Paul Sabatier de Toulouse. 1991.

D

[Daigle *et al.*, 06]: Daigle M., Roychoudhury I., Biswas G.and Koutsoukos X., Efficient simulation of component-based hybrid models represented as hybrid bond graphs. Technical Report ISIS- 06-712, Institute for Software Integrated Systems Vanderbilt University, Nashville, TN, USA, 2006.

[David et Alla, 01]: David. R, Alla. H. ''On Hybrid Petri Nets''. Discrete Event Dynamic Systems: Theory and Applications. 2001.

[David et Alla, 04]: David, R. et H. Alla., Discrete, Continuous, and Hybrid Petri Nets, Berlin, Heidelberg Springer, 2004.

[David et Alla, 05]: David R. et Alla H., " Discrete, Continuous, and Hybrid Petri Nets", Springer, 2005.

[David et Alla, 10]: David, R. et H. Alla. Discrete, Continuous and Hybrid Petri Nets, Springer, Berlin Heidelberg, 2010.

[David et Alla, 87]: David R. et Alla H., "Continuous Petri Nets" Dans les proceedings of the eight European workshop on application and theory of Petri nets, Pages 275-294, Zaragoza (Espagne), Juin 1987.

[David et Alla, 92]: David R. et Alla H., "Du Grafcet aux Réseaux de Petri", Hermès, 1992.

[David et. Alla, 92] : R. David et H. Alla, "Du Grafcet aux Réseaux de Petri", Hermès, 1992.

[David, 91]: David R., Modeling of Dynamic Systems by Petri Nets, European Control Conference, 91, p.136147,Grenoble, France, 1991.

[Daws et Yovine, 96]: Daws C. and Yovine S., Reducing the number of clock variables of timed automata. In Proc. 1996 IEEE Real-Time Systems Symposium, IEEE International, RTSS'96, pages 73–81,Washington, DC, USA, december 1996. IEEE Computer Society Press.

[De schutter *et al.*, 03]: De schutter. B, Heemels. W. P.M. H, Bemporad. A. ''Modeling and Control of Hybrid Systems''. Lecture notes of the DISC. 2003.

[Decarlo *et al.*, 00]: Decarlo.R, Branicky. MS, Pettersson. S, Lennartson.B., "Perspective and Results on The Stability And Stabilizability of Hybrid Systems". *Proceeding of the IEEE, vol. 8. N° 7. July. 2000.*

[Demongodin et Rouibia, 03] : Demongodin I. et Rouibia S., "Modélisation par Réseaux de Petri Lots et Analyse de L'état Stable Par Automates Hybrides", 4e conférence francophone de modélisation et simulation MOSIM'03, (Toulouse) France, 2003.

[Derbel *et al.*, 09]: Derbel, H., Alla, H., Ben Hadj-Alouane, N., & Yeddes, M., Online Diagnosis of Systems with Rectangular Hybrid Automata Models. In 13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM_09) (p. 123-129). Moscou, Russia, 2009.

[Derbel, 09] : Derbel, H. Diagnostic à base de modèles des systèmes temporises et d'une sous-classe de systèmes dynamiques hybrides. Thèse de doctorat de l'université joseph fourrier, Grenoble, 2009.

[Deschamps, 07] : Deschamps, E. Diagnostic de services pour la reconfiguration dynamique de systèmes à événements discrets complexes, thèse de doctorat, Laboratoire d'Automatique de Grenoble, 2007.

[Domlan *et al.*, 04]: Domlan E. A., Maquin D., and Ragot J., Diagnostic des systèmes à commutation, approche par la méthode de l'espace de parité. In Conférence Internationale Francophone d'Automatique, CIFA'2004, Douz, Tunisie, novembre 2004.

[Dubuisson, 01]: Dubuisson, B. *Diagnostic, intelligence artificielle et reconnaissance des formes*. Traité IC2 : Information - Commande - Communication. Hermes, 2001.

[Dubuisson, 90]: Dubuisson, B. Diagnostic et reconnaissance des formes, Hermès, 1990.

E

[EL Mezyani, 05] : EL Mezyani T..méthodologie de surveillance des systèmes dynamiques hybrides. Thèse de doctorat, spécialité : automatique et informatique industrielle, préparé au laboratoire d'automatique Génie informatique et Signal UMR CNRS 8146 de l'université des sciences et technologies de Lille, 2005.

[El Touati *et al.*, 09]: El Touati Y., Yeddes M., Ben Hadj Alouane N. and Alla H. du réseaux de Petri Temporel Etendu vers les automates hybrides linéaires pour l'analyse des systèmes. Author manuscript, published in IEEE conférence Internationale Francophone d'Automatique, Bucarest, Romania, CIFA 2009.

[EL Touati, 13] : EL Touati Yamen. Synthèse de contrôleurs par réseaux de petri temporels étendus. Thèse de doctorat, spécialité: informatique, préparée au sein de l'u.r. OASIS en coopération avec GIPSA-Lab, Université de Manouba, École Nationale des Sciences de l'Informatique, 2013.

[Engell, 97]: Engell S. Modelling and analysis of hybrid systems. 2nd IMACS MATHMOD Conference, pp. 17-31, Vienne, Autriche, 1997.

F

[Farreny, 89]: Farreny, H. Les systèmes experts - Principes et exemples. Cépadues, 1989.

[Favela, 99] : Favela A. Modélisation et analyse du comportement dynamiques des systèmes hybrides : une approche basée sur le modèle de l'automate hybride. PhD thesis, Laboratoire d'Automatique de Grenoble-Institut National Polytechnique de Grenoble, 1999.

[Flaus et Thévenon, 00]: Flaus J.M. and Thévenon L., Modular representation of complex hybrid systems: Application to the simulation of batch processes. Simulation Practice and Theory (SIMPRA), 2000.

[Flaus, 98]: Flaus J.M. Modeling and Analysis of Hybrid Dynamical Systems: an Introduction. Journal Européen des Systèmes Automatisés (JESA), vol. 32, n°7-8, pp. 797-830, 1998.

[Frank, 90]: Frank, P. M. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy – a survey and some new results. *Automatica*, 1990, vol. 26(3), p. 459 – 474.

[Frank, 96]: Frank, P. M. Analytical and qualitative model-based fault diagnosis - a survey and some new results. *European Journal of Control*, 1996, vol. 2(1), p. 6 – 28.

[Frehse, 05]: Frehse G. Phaver: algorithmic verification of hybrid systems past hytech. In M. Morari and L. Thiele, editors, editors, Hybrid Systems: Computation and Control: 8th International Workshop, HSCC2005, Zurich, Switzerland, LNCS 3414, pages 258–273. Springer, march 2005.

G

[Gertler, 97]: Gertler J., "Fault Detection and Isolation using Parity Relations". Control Engineering Practice, 1997.

[Ghazel *et al.*, 05] : Ghazel, M., Toguéni et M. Bigang. «A monitoring approach for discrete events systems based on a timed perti net model», dans 16th IFAC World Congress, 2005.

[Ghomri, 12] : Ghomri Latéfa. Synthèse de contrôleur de systèmes hybrides à flux continu par réseaux de Petri hybrides. Thèse de doctorat, université of Abou-Bekr Belkaïd–Tlemcen, Algérie, Mars 2012.

[Gomaa et Gentil, 96]:Gomaa, M. et S. Gentil, «Hybrid industrial dynamical system supervision via hybrid continuous causal petri nets», dans CESA'96 IEEESMC IMACS Symposium on Discrete Events and Manufacturing Systems, Lille, France, Springer-Verlag, p. 380-384, 1996.

[Gomaa, 97] :Gomaa, M. 1997, Représentation et Supervision des Systèmes Hybrides par Réseaux de Petri, thèse de doctorat, Institut national polytechnique de Grenoble, Grenoble, 1997.

[Guéguen *et al.*, 08]: Guéguen H., Lefebvre M.A., Nasri O. and Zaytoon J., Safety verification and reachability analysis for hybrid systems. Proceedings of the 17th World Congress the International Federation of Automatic Control Seoul, Korea, July 6-11, 2008.

[Guéguen et Lefebvre, 01]: Guéguen, H. and Lefebvre, M. A., A comparison of mixed specification formalisms. Journal Europeen des Systemes Automatises (APII JESA)., vol. 35, n°4, pp. 381-394, 2001.

[Guillaume *et al.*, 03]: Guillaume G. Gardey, O.H. Roux, and O.F. Roux. A zone-based method for computing the state space of a time Petri net. In In Formal Modeling and Analysis of Timed Systems, (FORMATS'03), volume 2791 of Lecture Notes in Computer Science, pages 246 259, Marseille, France, Springer, September 2003.

[Guillaume *et al.*, 06]: Guillaume G. Gardey, O.H. Roux, and O.F. Roux. State space computation and analysis of time Petri nets. Theory and Practice of Logic Programming (TPLP). Special Issue on Specification Analysis and Verification of Reactive Systems, 6(3) :301–320, 2006. Copyright Cambridge Press.

H

[Haar *et al*., 00]: Haar S., Kaiser L., Simonot-Lion F., and Toussaint J. On equivalence between timed state machines and time petri nets. Rapport de recherche RR 4049, INRIA, 2000.

[Haar *et al*., 02]: Haar S., Kaiser L., Simonot-Lion F., and Toussaint J. Equivalence of timed state machines and safe tpn. In Proceedings of the Sixth International Workshop on Discrete Event Systems (WODES'02), WODES'02, pages 119–124, Washington, DC, USA, 2002. IEEE Computer Society.

[Hamscher *et al.*, 92]: Hamscher, W. Console, L. and Kleer. J. Readings in model-based diagnosis. *Morgan Kaufmann, San Meteo, CA, Etats-Unis*, 1992.

[Harel *et al.*, 87]: Harel, D., A. Pnueli, J. Schmidt et R. Sherman « On the Formal Semantics of Statecharts », dans Proc. 2nd IEEE Symp. on Logic in Computer Science, Ithaca, NY, p. 54- 64, 1987.

[Harel, 87]: Harel, D. « Statecharts : A visual formalism for complex systems », Sci. Comput. Programming 8, p. 231–274, 1987.

[Henzinger *et al.*, 95]: T.A. Henzinger, P.W. Kopke, A.Puri et P. Varaiya, "What's decidable about hybrid automata? The algorithmic analysis of hybrid systems", Proceedings of 27th annual ACM Symposium on theory of computing, pp. 373-382, 1995.

[Henzinger *et al.*, 97]: Henzinger, T. A., P. H. Ho et H. W. Toi., «Hytech : A model checker for hybrid systems», International Journal on Software Tools for Technology Transfer, vol. 1, no 1-2, p. 110–122. 1997.

[Henzinger *et al.*, 98]: Henzinger, T., P. Kopke, A. Puri et P. Varaiya. « The what's decidable about hybrid automata? », Journal of Computer and System Sciences, vol. 57, p. 94-124, 1998.

[Henzinger et Majumdar, 00]: Henzinger, T. A. et Majumdar R., «Symbolic model checking for rectangular hybrid systems», dans TACAS 2000 : Tools and algorithms for the construction and analysis of systems, Lecture Notes in Computer Science, New-York, Springer-Verlag, p. 142–156. 2000.

[Henzinger et Rusu, 98]: T. A. Henzinger and V. Rusu. Reachability verification for hybrid automata. in HSCC'98 : hybrid systems computation and control, lecture notes in computer science 1386, pages 190–204. Springer-Verlag, 1998.

[Henzinger, 96]: Henzinger T.A.. The theory of hybrid automata. Proceedings of the 11th Annual Symposium on Logic in Computer Science, LNCS, pages 278–292, 1996.

[Hoblos, 01]: Hoblos G. "Contribution à l'analyse de la tolérance aux fautes des systèmes d'instrumentation", thèse à l'université de Lille (France), soutenue le 20 mars 2001.

[Holliday et Vernon, 87]: Holliday M.A. et Vernon M.K., "A generalized timed Petri net model for performance analysis". IEEE Transactions in Software Engineering, 13. IEEE Computer Society, 1987.

I

[Isermann et Balle, 00]: Isermann. R and P. Balle, "Applied terminology of fault detection, supervision and safety for technical processes", Site internet de IFAC Symposium on Fault Detection Supervision and Safety for Technical Process, 2000.

[Isermann et Balle, 97]: Isermann, R., and P. Balle. "Trends in the application of model-based fault detection and diagnosis of technical processes". Control Engineering Practice, 5(5), pp. 709-719, 1997.

[Isermann, 84]: Isermann, R. «Process fault detection based on modeling and estimation methods – a survey», Automatica, vol. 20, n° 4, p. 387–404, 1984.

J

[Jéron *et al.*, 06]: Jéron, T., Marchand, H., Pinchinat, S., & Cordier, M. Supervision patterns in discrete event systems diagnosis. In Proc. 8th international workshop on discrete event systems (WODES 2006), 2006.

K

[Karsai *et al.*, 03] : Karsai, G., S. Abdelwahed et G. Biswas. 2003, «Integrated diagnosis and control for hybrid dynamic systems»,

[Kesten *et al.*, 92]: Y. Kesten, A. Pnueli, J. Sifakis, and S. Yovine. Integration graphs : A class of decidable hybrid systems. In Hybrid Systems, volume 736 of Lecture Notes in Computer Science, pages 179–208. Springer-Verlag, 1993. (appeared in In Proceedings of Workshop on Theory of Hybrid Systems , Lyngby, Denmark, June 1992).

[Kesten *et al.*, 99]: Y. Kesten, A. Pnueli, J. Sifakis, and S. Yovine. Decidable integration graphs. Information and Computation, 150(2) :209–243, May 1999.

[Kesten et Pnueli, 92]: Kesten Y. and Pnueli A. Timed and hybrid statecharts and their textual representation. Formal techniques in real-time and fault-tolerant systems, LNCS, 571 :591–620, 1992.

[Khansa, 97]: Khansa W., "Réseaux de Petri p-temporels : contribution à l'étude des systèmes à événements discrets". Thèse de doctorat, Université de Savoie, 1997.

[Kloetzer et Belta, 06]: Kloetzer M. and Belta C., Reachability analysis of multiaffine systems. In J Hespanha and A Tiwari, editors, Hybrid Systems: Computation and Control: 9th International Workshop, HSCC2006, Santa Barbara, CA, USA, LNCS 3927, pages 348–362. Springer, march 2006.

[Kopke, 96]: Kopke, P. W. The Theory of Rectangular Hybrid Automata, thèse de doctorat, Cornell University, NY, USA. 1996.

[Kosaraju, 82] : Kosaraju S. R., "Decidability of Reachability in Vector Addition System", in *Proc. 14th Annual Symposium Theory Computing*, San Francisco (US), pp. 267-281, 1982.

[Koutsoukos *et al*., 01]: Koutsoukos, X., F. Zhao, H. Haussecker, J. Reich et P. Cheung, «Fault modeling for monitoring and diagnosis of sensor-rich hybrid systems», dans Proceedings of the IEEE Conference on Decision and Control, p. 793−801, 2001.

[Koutsoukos *et al.*, 02]: Koutsoukos X. , Kurien J. , and Zhao F., Monitoring and diagnosis of hybrid systems using particle filtering methods. In Proceedings of the 15th International Symposium on Mathematical Theory of Networks and Systems MTNS 2002, August 2002.

[Kurovszky, 02] : Kurovszky M. Etude des Systèmes Dynamiques Hybrides par représentation d'état discrète et automate hybride. Thèse de doctorat, l'INPG, France, 2002.

L

[Lafortune et Cassandra, 98]: Lafortune S., Cassandra G. : Introduction to discrete event systems : *Kuwer Academic Publishers* , :848 pages. Hardbound (.), 1998.

[Larsen *et al.*, 95]: K.G. Larsen, P. Pettersson, and W. Yi. Model-checking for real-time systems. In FCT'95 : Proceedings of the 10th International Symposium on Fundamentals of Computation Theory, pages 62–88, London, UK, 1995. Springer-Verlag.

[Le bail *et al*., 91]: Le Bail J., Alla H., David R. Hybrid Petri Nets, European Control Conference, p.1472 1477, Grenoble, France, 1991.

[Le bail *et al*., 92]: Le bail. J, Alla. H, David. R., ''Asymptotic continuous Petri Nets : An affecient approximation of discrete event systems''. Proceeding of the IEEE conference robotics and automation, Nice, France May, 1992.

[Lee *et al.*, 03]: Lee E.A., Neuendorffer S., and Wirthlin M. J. Actor-oriented design of embedded hardware and software systems. Journal of Circuits, Systems, and Computers 231–260, 2003.

[Lefebvre, 00] : Lefebvre, D. «Contribution à la modélisation des systèmes dynamiques à événements discrets pour la commande et la surveillance», Habilitation à Diriger des Recherches, Université de Franche Comté/ IUT Belfort – Montbéliard, 2000.

[Lesecq et al., 01]: Lesecq S., Petropol S., Barreau A. "Asyunchronous motor parametric faults diagnosis using wavelet analysis", Conférence IEEE Sdemped 2001, Goriza , Italie.

[Liberzon, 03]: Liberzon D., Switching in Systems and Control. ISBN 0-8176-4297-8, Jun 2003.

[Liberzon, 99]: Liberzon. D, Morse. S., ''Basic Problems in Stability and Design of Switched Systems''. *IEEE Control Systems Magazine. October, 1999.*

[Lime et Roux, 06]: Lime D. and Roux O.H., Model checking of time petri nets using the state class timed automaton. Discrete Event Dynamic Systems, (vol.16, pp.179–205), 2006.

[Lin et Wonham, 94]: Lin, F. et W. M. Wonham «Diagnosability of discrete event systems and its applications», Discrete Event Dynamic Systems, vol. 4, no 2, p. 197–212, 1994.

[Liu *et al.*, 13]: Hu-Chen Liu, Qing-Lian Lin, Ming-Lun Ren. Fault diagnosis and cause analysis using fuzzy evidential reasoning approach and dynamic adaptive fuzzy Petri nets, Computers & Industrial Engineering 66, 899–908, 2013.

[Lunze, 00]: J. Lunze. Diagnosis of quantized systems based on a timed discrete-event model. In IEEE Transactions on Systems, Man, and Cybernetics, Part A, 30(3):322 335, May 2000.

[Lunze, 06]: Lunze, J. «Diagnosis of discretely controlled continuous systems», Automatisierungstechnik, vol. 54, N°8, p. 385−395, 2006.

[Lygeros *et al.*, 03]: Lygeros. J, Johansson. K, Simic. S, Zhang. J, Sastry. S. ''Dynamical Properties of Hybrid Automata''. IEEE Transaction On Automatic control. Vol. 48. N°. 1. January, 2003.

[Lygeros, 04]: Lygeros, J., editor (2004). Lecture Notes on Hybrid Systems, volume 2034. Department of Electrical and Computer Engineering University of Patras.

M

[Mayr, 81] : Mayr E. W., "An Algorithm for the General Petri Net Reachability Problem", in *Proc. 13th Annual Symposium Theory Computing*, pp. 238-246, 1981.

[Mendler et Lüttgen, 01]: Mendler, M. et G. Lüttgen« Statecharts: from visual syntax to model-theoretic semantics », dans Workshop on Integrating Diagrammatic and Formal Specification Techniques (IDFST 2001), p. 615–621, 2001.

[Merlin, 74]: Merlin P.M., "A study of recoverability of communication protocols". Ph.D. Thesis, Department of Computer Science, University of California, 1974.

[Milne, 87]: Milne R., Strategies for diagnosis. IEEE Transactions on Systems, Man and Cybernetics, Vol.17, N°3, p 333-339, 1987.

[Mokhtari, 07] : Mokhtari, A. 2007, Diagnostic des systèmes hybrides : développement d'une méthode associant la détection par classification et la simulation dynamique, thèse de doctorat, Institut National des Sciences Appliquées de Toulouse.

[Montmain, 92]: Montmain, J. Interprétation qualitative de simulation pour le diagnostic en ligne de procédés continus. Thèse de doctorat, Institut National Polytechnique de Grenoble, 1992.

[Mosterman et Biswas, 00]: Mosterman. P, Biswas. G. '' A comprehensive methodology for building hybrid models of physical systems''. Artificial Intelligence 00 (2000) 1–39. 2000.

[Mosterman, 02]: Mosterman P., A modelling and simulation environment for hybrid bond graphs. Proceedings of the IMECHE Part I, Journal of Systems and Control Engineering, Part I, 216(1) pp:35–46, February 2002.

[Mosterman, 97] : Mosterman, P. J, Hybrid Dynamic Systems : a Hybrid Bond Graph Modeling Paradigm and its Application in Diagnosis, thèse de doctorat, Vanderbilt University, 1997.

[Mosterman, 97]: Mosterman P. ''Hybrid Dynamic Systems: A Hybrid Bond Graph Modeling Paradigm And its Application In Diagnosis''. PhD thesis. Nashville, Tennessee. 1997.

[Müller et Stauner, 00]: Müller, O. et T. Stauner. «Modelling and verification using linear hybrid automata–a case study», Mathematical and Computer Modelling of Dynamical Systems, vol. 6, N°1, p. 71–89, 2000.

[Murata, 89]: Murata T., "Petri-nets: Properties, Analysis and Applications", Dans les proceedings de IEEE, 77(4) : 541-580, 1989.

N

[Narasimhan *et al.*, 00]: Narasimhan, V.S., G. Biswas, G. Karsai, T. Pasternak and F. Zhao, "Building observers to handle fault isolation and control problems in hybrid systems", Proc. IEEE Intl, Conference on Systems, Man, and Cybernetics, Nashville, TN. pp. 2393–2398, 2000.

[Nicollin *et al.*, 93]: X. Nicollin, J. Sifakis, and S. Yovine. From atp to timed graphs and hybrid systems. Acta Informatica, 30(2) :181–202, 1993.

[Nicollin, 93]: Nicollin, A. Olivero, J. Sifakis, and S. Yovine. An approch to the description and analysis of hybrid systems. Lecture Notes in Computer Science, 736, 1993.

[Nourelfath, 97]: Nourelfath, M. Extension de la théorie de la supervision à la surveillance et à la commande des systèmes à événements discrets : application à la sécurité opérationnelle des systèmes de production. Thèse de doctorat, L'Institut National des Sciences Appliquées de Lyon, INSA. 1997.

O

[Ondel, 06]: Ondel, O. Diagnostic par reconnaissance des formes : Application à un ensemble convertisseur-machine asynchrone. Thèse de doctorat, École Centrale de Lyon, 2006.

P

[Pandalai et Holloway, 00]: Pandalai, D. N., & Holloway, L. E. Template languages for fault monitoring of timed discrete event processes. IEEE Transactions Automatic Control, 45(5), 868–882, 2000.

[Patton et Chen, 91]: Patton, R. J. and Chen, J. A review of parity space approaches to fault diagnosis. In *SAFEPROCESS'91*, p. 239–255, Baden-Baden (Allemagne), 1991.

[Patton, 94] : Patton, R. Robust model based fault diagnosis: the state of the art. In *SAFEPROCESS'94*, volume 1, p. 1–23, Helsinki, Finland, 1994.

[Patton, 97]: Patton, R., Fault-tolerant control systems: The 1997 situation. In Proc. of the IFAC symposium on fault detection, supervision and safety for technical processes, Hull, UK, pages 1033–1054, 1997.

[Peleties, 91]: Peleties, '' Stability of Switched and Hybrid Systems''. *American control conference. 1991.*

[Peterson, 81]: Peterson J.L., "Petri nets theory and the modeling of systems", Prentice-Hall, 1981.

[Petri, 62]: Petri C.A., "Kommunication mit Automaten", Thèse de PhD, Université de Bonn, Allemagne, 1962.

[Pettersson et Lennartson, 02]: Pettersson S. and Lennartson B., Hybrid system stability and robustness verification using linear matrix inequalities. International Journal of Control, 75:1335–1355, 2002.

[Philippot, 06] : Philippot, A. Contribution au diagnostic décentralisé des systèmes à événements discrets : Application aux systèmes manufacturiers, thèse de doctorat, L'Université de Reims Champagne Ardenne, 2006.

[Potocnik *et al.*, 03]: Potocnik B., Bemporad A., Torrisi F.D., Music G., and Zupancic B., Hysdel modeling and simulation of hybrid dynamical systems. In Proceedings of MATHMOD Conference, pages 5–7, February 2003.

[Puri *et al.*, 96]: Puri, A. Borkar, V. and Varaiya, P. e-approximation of differential inclusions. In In Proceedings of Hybrid Systems III Workshop : Verification and Control, vol. 1066 of Lecture Notes in Computer Science, pages 362–376, 1996.

[Puri et Varaiya, 94]: A. Puri and P. Varaiya. Decidability of hybrid systems with rectangular differential inclusions. In David Dill, editor, Computer Aided Verification, volume 818 of Lecture Notes in Computer Science, pages 95–104. Springer Berlin / Heidelberg, 1994. 10.1007/3-540-58179-046.

R

[Ramadge et Wonham, 1987]: Ramadge, P. et W. Wonham «Supervisory control of a class of discrete event systems», SIAM J.Contr. Optim, vol. 25, p. 57−96, 1987.

[Ramamoorhy et Ho, 80]: Ramamoorhy C.V. et Ho G.S., "Performance evaluation of asynchronous concurrent systems using Petri nets". IEEE Transactions in Software Engineering, 6. IEEE Computer Society, 1980.

[Ramchandani, 74]: Ramchandani C., "Analysis of asynchronous concurrent systems using Petri nets", PhD Thesis, Project MAC, MAC-TR 120, MIT, 1974.

[Rayhane, 04]: Rayhane, H. Surveillance des systèmes de production automatisés : Détection et Diagnostic. Thèse de doctorat, Institut National Polytechnique de Grenoble, INPG, 2004.

[Renganathan et Bhaskar, 11]: Renganathan, K., & Bhaskar, V., An observer based approach for achieving fault diagnosis and fault tolerant control of systems modeled as hybrid petri nets. fISAg Transactions, 50 (3), 443 – 453, 2011.

[Renganathan et Bhaskar, 13]: Renganathan K., & Bhaskar, V., Modeling, analysis and performance evaluation for fault diagnosis and Fault Tolerant Control in bottle-filling plant modeled using Hybrid Petri nets. Applied Mathematical Modelling 37, 4842–4859, 2013.

[Reutenauer, 89] : Reutenauer C., *Aspects mathematiques des reseaux de Petri,* Masson, Paris, 1989.
[Roth *et al.*, 11]: Roth, M., Lesage, J.-J., & Litz, L. The concept of residuals for fault localization in discrete event systems. Control Engineering Practice, 19, 978–988, 2011.

S

[Sampath et al., 1998]: Sampath, M., S. Lafortune et D. Teneketzis. 1998, «Active diagnosis of discrete event systems», IEEE Transactions on Automatic Control, vol. 43, N°7, p. 908−929.
[Sampath *et al.*, 95]: Sampath, M., R. Sengupta, S. Lafortune, K. Sinnamohideen et D. Teneketzis. «Diagnosability of discrete event systems», IEEE Transactions on Automatic Control, vol. 40, no9, p. 1555–1575, 1995.
[Sampath *et al.*, 96] Sampath, M., R. Sengupta, S. Lafortune et K. Sinnamohideen. 1996, «Failure diagnosis using discrete event models», IEEE Transactions on Control Systems Technology, vol. 4, N°2, p. 105-124.
[Sava et Alla, 06]: Sava A.T. and Alla H., A control synthesis approach for time discrete event systems. Math. Comput. Simul., 70(5) :250–265, 2006.
[Sava, 01] : Sava A.T., Sur la Synthèse de la Commande Des Systèmes à Evenements Discrets Temporisés. thèse de doctorat, Institut Nationale Polytechnique de Grenoble, 2001.
[Sayed-Mouchaweh *et al.*, 08]: Sayed-Mouchaweh, M., Philippot, A., & Carré-Ménétrier, V. Decentralized diagnosis by Boolean discrete event system model: Application on manufacturing systems. International Journal of Production Research 46(19), 5469–5490, 2008.
[Sayed-Mouchaweh, 12]: Sayed-Mouchaweh, M. Decentralized fault free model approach for fault detection and isolation of discrete event systems. European Journal of Control, 18(1), 82–93, 2012.
[Sifakis et Yovine, 96]: Sifakis J. and Yovine S., Compositional specification of timed systems (extended abstract). In: STACS'96, Proceedings of the 13th Annual Symposium on Theoretical Aspects of Computer Science. LNCS 1046, Springer-Verlag, pages 347– 359. Springer, 1996.
[Sifakis, 79]: Sifakis J., "Performance evaluation of systems using nets. Net theory and applications", Advanced course on general net theory of processes and systems, LNCS 84. Springer, 1979.
[Spathopoulos, 00]: Spathopoulos M. «Supervisory control for rectangular hybrid automata», dans 39th IEEE Conference on Decision and Control, p. 35–41. 2000.
[Srba, 08]: Srba J. Comparing the expressiveness of timed automata and timed extensions of petri nets. In Franck Cassez and Claude Jard, editors, Formal Modeling and Analysis of Timed Systems, volume 5215 of Lecture Notes in Computer Science, pages 15–32. Springer Berlin Heidelberg, 2008.
[Starke, 78]: Starke P. H.. Free Petri nets languages. Mathematical Foundations of Computer Science, LNCS 64. Springer, 1978.
[Stiver *et al.*, 92]: Stiver et al,''Modeling and analysis of Hybrid Control Systems''. CDC Arizona USA. 1992.
[Stiver *et al.*, 96]: Stiver, J. Antsaklis, P. and Lemmon, M. A logical des approach to the design of hybrid control systems. Math. and Computer Modeling, Special Issue on Discrete Event Systems, vol. 23(11/12), p. 55–76, 1996.

T

[Tavernini, 87]: Tavernini. M, ''Differential automata and their discrete simulators''. Nonlinear Analysis Theory, Method and Applications, 11.6, pp 665-683. 1987
[Taylor, 94]: Taylor H., A Modeling Language for Hybrid Systems, Odyssey Research Associates (ORA) 301 Dates Drive, Ithaca, NY 14850, From Proc_ CACSD-94. (IEEE/IFAC Symp. on Computer –Aided Control System Design),Tucson, AZ, March 94.

[Tittus, 95]: Tittus. M.''Control synthesis for batch process''. PhD thesis, Chalmers University of Technology. 1995

[Toguyéni *et al*., 06] : Toguyéni A.K.A. , Craye E. and Sekhri L. Study of the Diagnosability of Automated Production Systems Based on Functional Graphs. Mathematics and Computers in

[Toguyeni, 92]: Toguyeni, A., Surveillance et diagnostic en ligne dans les ateliers flexibles de l'industrie manufacturière, thèse de doctorat, Université de Lille, 1992.

[Tolbi *et al*., 16]: B. Tolbi, H. Tebbikh & H. Alla (2016): Fault-tolerant continuous flow systems modelling, International Journal of Systems Science, DOI: 10.1080/00207721.2016.1160454

[Touaf, 05]: Touaf Samir. "diagnostic logique des systemes complexes dynamiques dans un contexte multi-agent", thèse de doctorat, université Joseph Fourier – Grenoble 1, France, 02 mars 2005.

V

[Valette *et al*., 89]: Valette, R., J. Cardoso et D. Dubois «Monitoring manufacturing systems by means of petri nets with imprecise markings», dans IEEE Conference Intelligent Control, NY, p. 233–238, 1989.

[Van Der Schaft et Schumacher, 00]: Van Der Schaft A. and Schumacher H. An Introduction to hybrid Dynamical Systems, lecture Notes in control and Information Sciences, Springer-Verlag, Berlin (Allemagne), Londres(Angleterre),251:175Pp, 2000.

[Villani *et al.*, 05]: Villani E, Pascal. J. C, Miyagi. P, Valette. R. ''A Petri Net-Based Object-Oriented Approach for the Modelling of Hybrid Productive Systems''. Nonlinear Analysis 62, 1394–1418, 2005.

[Villani *et al.*, 07]: Villani E, Miyagi. P, Valette. R. '' Modelling and Analysis of Hybrid Supervisory Systems: A Petri Net Approach''. Advances in Industrial Control series ISSN 1430-9491 ISBN 978-1-84628-650-6 e-ISBN 978-1-84628-651-3 Printed on acid-free paper © Springer-Verlag London Limited. 2007.

[Villemeur, 88] : Villemeur, A. Sûreté de fonctionnement des systèmes industriels, vol. 67, Edition EYROLLES, Collection DER-EDF, 1988.

W

[Walter, 83]: Walter B., "Timed Petri-nets for modelling and analyzing protocols with real-time characteristics". Third IFIP workshop on protocols specification, testing and verification. North-Holland, 1983.

[Willsky, 76]: Willsky A.S., "A survey of design methods for failure detection in dynamic systems". Automatica, Vol. 12, pp.601-611, 1976.

Z

[Zad *et al*., 03]: Zad, S. H., R. H. Kwong et W. M. Wonham, «Fault diagnosis in discrete-event systems: Framework and model reduction», IEEE Transactions On Automatic Control, vol. 48, N°7, p. 1199−1212. 2003.

[Zad *et al*., 05] : Zad, S., R. Kwong et W. M. Wonham, «Fault diagnosis in discrete-events systems: Incorporating timing information», IEEE Transactions On Automatic Control, vol. 50, N°7, p. 1010−1015, 2005.

[Zad *et al.*, 98]: Zad, S. H., R. H. Kwong et W. M. Wonham., «Fault diagnosis in discrete-event systems », dans Proceedings of the IEEE Conference on Decision and Control (CDC'98), p. 3769–3774, 1998.

[Zad *et al*., 98]: Zad, S. H., R. H. Kwong et W. M. Wonham. «Fault diagnosis in discrete-event systems», dans Proceedings of the IEEE Conference on Decision and Control (CDC'98), p. 3769−3774, 1998.

[Zad *et al*., 99]: Zad, S. H., R. H. Kwong et W. M. Wonham. «Fault diagnosis in finite-state automata and timed discrete-event systems», dans Proceedings of the 38th IEEE Conference on Decision and Control, 1999.

[Zadeh, 65]: Zadeh, L., "fuzzy sets", Information and Control, vol.8, p. 338 –353,1965.

[Zaytoon, 01] : Zaytoon J. Systèmes dynamiques hybrides. Hermes Sciences publications, 2001.

[Zemouri, 03] : Zemouri, M. R., Contribution à la surveillance des systèmes de production à l'aide des réseaux de neurones dynamiques : Application à la e-maintenance., thèse de doctorat, l'Université de Franche-Comté, 2003.

[Zhang et Jiang, 03a]: Zhang, Y. et Jiang, J., Bibliographical review on reconfigurable fault-tolerant control systems. *In Proc. of the 5th IFAC Symposium on Fault Detection, Supervision and safety for Technical Processes*, June 9-11, Washington, D.C., USA, pages 265 276, 2003a.

[Zhang et Jiang, 08]: Zhang, Y. et Jiang, J., Bibliographical review on reconfigurable fault-tolerant control systems. Annual Reviews in Control, 32:229–252, 2008.

[Zwingelstein, 95] : Zwingelstein, G. Diagnostic des défaillances. Traité des nouvelles Technologies, série Diagnostic et Maintenance, Hérmes, 1995. Simulation, Vol. 70, issues 5-6, 24, pp. 377-393, Elsevier, February 2006.

Les travaux de cette thèse ont été présentés via plusieurs articles et communications internationales et nationales qui sont comme suit:

Publications:

Article : Fault-tolerant continuous flow systems modelling, *a été publié dans la revue "**Taylor & Francis online**", **International Journal of Systems Science**, DOI: 10.1080/00207721.2016.1160454. Mars 2016. Auteurs:* ***Bilal TOLBI,*** *Hicham TEBBIKH & Hassane ALLA*
http://www.tandfonline.com/doi/full/10.1080/00207721.2016.1160454

Communications Internationales:

1- B. Tolbi and S. Kechida ' *Modeling and Control of Hybrid Dynamical Systems; Application On a System With Two Tanks* ' ***third international conference on systems and information processing ICSIP'13*** organisée à l'université de 08 Mai 1945-Guelma le 12-14 Mai 2013.

2- B. Tolbi et H. Tebbikh ' *Surveillance des systèmes dynamiques par automates hybrides à chronomètre'* ***la 8ème conférence international conception & production intégrées*** organisée à l'université Abou Bekr Belkaid-Tlemcen le 21-23 Octobre 2013.

Communications Nationales :

1- B. Tolbi, F. Bouriachi et S. Kechida '*Introduction à la commande des systèmes dynamiques hybrides : Application à un système à deux réservoirs'* la 2ème journée sur les signaux et systèmes JSS'11 organisée à l'université de 08 Mai 1945-Guelma le 12 Novembre 2011.

2- B. Tolbi et S. Kechida ' *Modélisation et commande des systèmes dynamiques hybrides : application a un système à deux réservoirs* ' la 8eme conférence de génie électrique (CGE08) organisée le 16-17 Avril 2013 à l'école militaire polytechnique de Bordj El Bahri, Alger.

3- S. Kechida, F. Bouriachi et **B. Tolbi** *'Modélisation et simulation des systèmes dynamiques hybrides'* la rencontre des femmes scientifiques Méditerranéennes REFSCIME2013 organisée à l'université de 20 Aout 1955-Skikda les 15 et 16 Avril 2013.

4- B. Tolbi, H. Alla et H. Tebbikh *'Surveillance des systèmes dynamiques hybrides'* la 1er journée sur les signaux et systèmes JSS'13 organisée à l'université de 08 Mai 1945-Guelma le 26 juin 2013.

5- B. Tolbi, H. Alla et H. Tebbikh *'Translation du RdP hybride élémentaire en automate hybride'* la 2ème journée sur les signaux et systèmes JSS'13 organisée à l'université de 08 Mai 1945-Guelma le 23 Novembre 2013.

6- B. Tolbi, H. Alla, H. Tebbikh, F. Bouriachi and A. Zeroual *'modeling of faults in fault-tolerant systems using hybrid automata and hybrid petri nets'* la 1er Journée Doctorale JD'14 organisée à l'université de 08 Mai 1945-Guelma le 01 Mars 2014.

7- A/H. Zeroual, N. Messai, S. Kechida, F. Hamdi and **B. Tolbi** *'data traffic estimation using a hybrid cell transmission model HCTM'* la 1er journée Doctorale JD'14 organisée à l'université de 08 Mai 1945-Guelma le 01 Mars 2014.

8- B.Tolbi *'Modeling of fault in fault-tolerent systems using hybrid automata and hybrid Petri Nets'* Journées Scientifiques du Laboratoire d'électronique Avancée JSLEA'2014 organisée à l'université Hadj Lakhdar de Batna, 23-24 Avril 2014.

9- B. Tolbi, H. Tebbikh et H. Alla, *'Une similarité temporelle entre les RDPH élémentaires et les AH linéaires en vue d'une modélisation des systèmes à flux continu tolérants aux fautes'* Journée Doctorale JD'15 organisée à l'université de 08 Mai 1945-Guelma le 12 Novembre 2015.