

50376
45514
99

No ordre: 1974

THESE

présentée à

L'université des Sciences et Technologies de Lille

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE

Spécialité: Productique: Automatique et Informatique Industrielle

par

Etienne LOIEZ

Ingénieur I.S.E.N.

CONTRIBUTION AU DIAGNOSTIC DE SYSTEMES ANALOGIQUES Modélisation par des bandes temporelles

Soutenue le 3 mars 1997 devant le jury composé de:

J.P. RICHARD	Président
M.O. CORDIER	Rapporteur
P. DAGUE	Rapporteur
P. BORNE	Examineur
J.P. CASSAR	Examineur
V. COCQUEMPOT	Examineur
P. TAILLIBERT	Examineur
P. VANHEEGHE	Examineur

Thèse dirigée par M. P. VANHEEGHE et M. P. BORNE, préparée au Laboratoire d'Automatique et d'Informatique Industrielle de Lille
(L.A.I.L.U.R.A. - C.N.R.S. D1440), Ecole Centrale de Lille, I.S.E.N.



Remerciements

Je remercie Monsieur Jean Pierre Richard, Professeur à l'Ecole Centrale de Lille, d'avoir bien voulu présider le jury de cette thèse.

Que Monsieur Philippe Vanheeghe, Responsable du département Signaux et Systèmes de l'Institut Supérieur d'Electronique du Nord, soit remercié pour le soutien moral et les conseils prodigués pendant ce travail.

Je remercie Monsieur Pierre Borne, Directeur Scientifique de l'Ecole Centrale de Lille, pour sa participation à la direction de ces travaux.

Je suis très reconnaissant envers Madame Marie-Odile Cordier, Professeur à l'université de Rennes 1, et Monsieur Philippe Dague, Professeur à l'université de Paris 13, d'avoir accepté d'être les rapporteurs de cette thèse.

Mes plus vifs remerciements vont à Monsieur Vincent Cocquempot, Maître de Conférence à l'université d'Artois, pour les longues heures passées à lire ce manuscrit.

Je remercie la société Dassault Electronique de m'avoir confié les travaux de recherche qui sont à l'origine de cette thèse et tout particulièrement Monsieur Patrick Taillibert. Son aide constante et son optimisme tout au long de ces trois années m'ont été extrêmement précieux.

Je remercie Monsieur Jean-Philippe Cassar, Professeur à l'université de Lille 1, pour avoir accepté de faire partie de ce jury.

Je tiens aussi à remercier Madame Corinne Bos-Plachez et Mademoiselle Magda Wozniak pour l'ambiance qu'elles ont fait régner dans notre bureau et pour les commentaires sur mon travail.

Je remercie le Conseil régional du Nord-Pas de Calais et l'Institut Supérieur d'Electronique du Nord pour le financement de mes travaux.

J'ai été touché par la confiance que m'a accordée Monsieur Jean-Nöel Decarpigny, Directeur de l'I.S.E.N., en m'acceptant dans son établissement.

Que Monsieur Michel Lannoo, Directeur de la Recherche à l'I.S.E.N., soit associé à ces remerciements.

Table des matières

1 Introduction	11
1.1 Les systèmes analogiques	11
1.2 Les techniques de diagnostic	12
1.2.1 Les systèmes experts	13
1.2.2 Les arbres de décision	13
1.2.3 Les dictionnaires de fautes	13
1.3 Le diagnostic à base de modèles	14
1.4 Etude de systèmes existants	16
1.5 Propositions	19

Partie I: Modéliser avec des Bandes Temporelles **21**

2 Modélisation des observations	25
2.1 Modélisation à l'aide de bandes temporelles	25
2.2 Modélisation à l'aide de suites de bandes temporelles	28
3 Relations entre les suites de bandes temporelles	31
3.1 Egalité de deux bandes temporelles	31
3.2 Egalité et équivalence de deux suites de bandes temporelles	32
3.3 Synchronisation de deux suites de bandes temporelles	39
3.4 Inclusion de suites de bandes temporelles	43
3.5 Intersection de deux suites de bandes temporelles	44
3.6 Signe d'une suite de bandes temporelles	49
4 Opérations primitives	51
4.1 Somme de deux suites de bandes temporelles	51
4.1.1 Définitions	51
4.1.2 Propriétés	53
4.1.3 Somme d'un intervalle et d'une suite de bandes temporelles	56
4.2 Multiplication de deux suites de bandes temporelles	59
4.2.1 Définitions	60
4.2.2 Propriétés	64
4.2.3 Multiplication d'une suite de bandes temporelles par un intervalle	66
4.3 Soustraction de deux suites de bandes temporelles	68

4.4	Intégrale d'une suite de bandes temporelles	68
4.4.1	Définitions	69
4.4.2	Propriétés	71
4.5	Conclusion	76
5	Généralisation	77
5.1	Analogie entre les opérations sur les intervalles et les opérations sur les bandes temporelles	77
5.2	Extension aux suites de bandes temporelles des opérations réelles	79
5.3	Opérations polynomiales sur les bandes temporelles	83
5.4	Extension primitive des opérations non polynomiales	85
Partie II: Consistance des modèles		89
6	Description du système	93
6.1	Les observations	93
6.2	Description du comportement théorique	94
6.3	Exemple de modèles	96
6.4	Description du système	98
7	Inconsistance d'un modèle	99
7.1	Inconsistance d'un modèle	99
7.2	C-inconsistance d'un modèle	102
7.3	Liens entre la consistance et la C-consistance	103
7.4	Conclusion	106
8	Famille d'algorithmes de détection d'inconsistance	107
8.1	Préliminaires	107
8.2	Calcul de l'extension temporelle d'un modèle	109
8.3	Inconsistance des modèles d'ordre 0	111
8.3.1	Algorithme de détection d'inconsistance des modèles d'ordre 0	111
8.3.2	Exemple	113
8.3.3	Extension à la C-consistance	114
8.3.3.1	Non continuité des suites de bandes temporelles	115
8.3.3.2	Expression des fonctions des modèles	116
8.3.3.3	Imprécision introduite par les constantes	118
8.3.3.4	Conclusion	121
8.4	Algorithme de consistance des modèles d'ordre 1	121
8.4.1	Exemple	121
8.4.2	Explication	125
8.4.3	Choix d'un algorithme	127
8.4.4	Algorithme de détection d'inconsistance d'un modèle d'ordre 1	128
8.4.4.1	Algorithme initial	128
8.4.4.2	Optimisation de l'algorithme	132
8.4.4.3	Complétude de l'algorithme	139
8.5	Extension aux modèles d'ordre n	142
8.5.1	Consistance des modèles d'ordre 2	142
8.5.1.1	Formalisation	142

8.5.1.2	Exemple : le circuit LC	143
8.5.1.3	Complétude de l'algorithme	145
8.5.1.4	Extension à l'ordre n	147
8.6	Conclusion	147

Partie III: Systèmes de diagnostic **151**

9 G.D.E et les SBT **155**

9.1	Rappel de G.D.E.	155
9.1.1	Les 4 étapes	155
9.1.2	La procédure d'inférence	156
9.2	Prédiction de SBT	158
9.2.1	Calcul des prédictions	159
9.2.1.1	Règle 1 : principe fondamental	159
9.2.1.2	Règles 2 & 3 : règles internes	163
9.2.1.3	Algorithme de calcul des prédictions	166
9.3	Détection des conflits	166
9.3.1	Confrontation de 2 prédictions	167
9.3.1.1	Confrontation de deux intervalles encadrant une grandeur constante	166
9.3.1.2	Confrontation de deux SBT représentant une grandeur	168
9.3.1.3	Confrontation de deux SBT représentant une grandeur et sa dérivée	168
9.3.1.4	Généralisation	169
9.3.2	Confrontation de n valeurs	170
9.3.2.1	Confrontation de k intervalles encadrant une grandeur constante	170
9.3.2.2	Confrontation de k SBT représentant une grandeur	171
9.3.2.3	Confrontation de k prédictions d'une grandeur et d'une prédiction de la dérivée de cette grandeur	171
9.3.2.4	Confrontation de la prédiction d'une grandeur et de k prédictions de la dérivée de cette grandeur	174
9.3.2.5	Confrontation d'une grandeur et de ses dérivées première et seconde	176
9.3.2.6	Conclusion	177
9.3.3	règles de gestion de la base de données	177
9.4	Limites du propagateur	178
9.5	Conclusion	181

10 DOGS **183**

10.1	DOGS et le Full adder	183
10.2	DOGS : le compilateur	185
10.2.1	Modèles d'ordre 0	185
10.2.2	Extension aux modèles d'ordre supérieur	190
10.3	Avantages et inconvénients d'un compilateur	191
10.4	Approche théorique	195
10.5	Conclusion	197

11 Implémentation et résultats	199
11.1 Création des Suites de Bandes Temporelles	199
11.1.1 Algorithme de création des SBT	199
11.1.2 Extension de l'algorithme	205
11.2 DOGS et le monostable	206
11.2.1 Le monostable	206
11.2.2 Compilation des modèles	207
11.2.3 Résultats	211
CONCLUSION	215
Références bibliographiques	219

Avant propos

Lorsqu'un système est en panne, il existe toujours deux solutions : soit le remplacer, soit le réparer, c'est à dire remplacer le ou les sous-systèmes défectueux. Dans le cadre des systèmes électroniques analogiques, l'avancée technologique de ces dernières années a eu comme conséquence l'utilisation de plus en plus fréquente de la première solution. Ceci est non seulement dû à la décroissance du coût de fabrication mais surtout à la difficulté croissante à diagnostiquer ces systèmes, c'est à dire à localiser les sous-systèmes défectueux. Néanmoins, le diagnostic reste un challenge pour les ingénieurs dans de nombreux cas : lorsque les composants sont chers, dans le cadre de la mise au point de prototypes et de l'amélioration de systèmes existant.

Cette thèse propose des solutions pour améliorer les systèmes automatiques de diagnostic de tels systèmes.

Chapitre 1

Introduction

Le but de ce chapitre est de présenter les difficultés spécifiques du diagnostic de systèmes analogiques et quelques solutions proposées dans la littérature pour surmonter ces difficultés. Cette étude nous amène à la présentation de nouvelles propositions.

1.1 Les systèmes analogiques

Nous proposons d'abord de caractériser les systèmes analogiques. Les différentes quantités physiques des systèmes seront appelées les *grandeurs*. Par exemple, en électronique, les grandeurs sont les tensions, les courants, etc...

Les systèmes analogiques peuvent être caractérisés de la manière suivante :

- Les grandeurs évoluent continûment dans le temps. Un exemple d'évolution de "grandeur" est illustré figure 1.1.

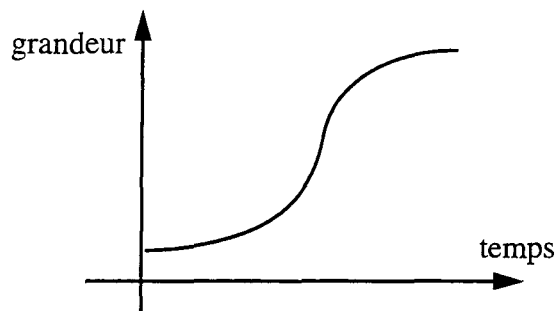


Figure 1.1 : Exemple de grandeur

- Le comportement des systèmes analogiques est caractérisé par la valeur des grandeurs et des variations de celles-ci. Par exemple, le courant traversant une capacité est proportionnel à la variation de la tension aux bornes de ce composant (i.e. le modèle d'une capacité est $i(t) = c v'(t)$ où $i(t)$ est le courant et $v'(t)$ est la dérivée de la tension). D'une manière générale, le comportement d'un système analogique est décrit par un ensemble d'équations différentielles.

Le diagnostic de systèmes analogiques soulève de nombreuses difficultés, caractérisées de la manière suivante [Dague et al, 91] :

- (1) Seul un faible nombre de grandeurs peuvent être mesurées. Par exemple, les courants des circuits électroniques ne sont en règle générale pas mesurables.
- (2) Les mesures des grandeurs sont imprécises. De plus, les processus de mesure fonctionnent par échantillonnage : les valeurs des grandeurs sont connues en un nombre fini d'instant.
- (3) Les variations des grandeurs ne sont pas mesurables. Ces variations peuvent uniquement

être déduites des mesures des grandeurs. Le problème est que la validité de cette déduction dépend de la procédure d'échantillonnage : toute grandeur doit être échantillonnée à au moins deux fois sa fréquence propre (i.e. théorème de Shannon). Or, il est irréaliste de supposer une procédure de mesure respectant le théorème de Shannon car les grandeurs ont des fréquences propres très différentes. Un exemple caractéristique est la mesure d'une grandeur variant très rapidement (par exemple, quelques nanosecondes) dans un système dont la période est importante (par exemple, quelques millisecondes). Un échantillonnage correct de ce signal aboutirait à l'obtention de 100 000 échantillons (un échantillon toutes les nanosecondes pendant une milliseconde) ! Or, aucun appareil de mesure n'est capable de réaliser une telle mesure.

- (4) Les paramètres des composants prennent leurs valeurs dans un domaine continu. Ceci engendre une infinité de fautes possibles car les paramètres d'un composant défectueux peuvent avoir une infinité de valeurs possibles. Par exemple, la valeur d'une résistance qui fuit est comprise entre 0 et sa valeur nominale.
- (5) Les valeurs des paramètres sont imprécises. Par exemple, la valeur d'une résistance peut être connue à 10% près.
- (6) Les systèmes sont en règle générale fortement bouclés. Ceci engendre une forte interaction entre les composants. La panne d'un composant provoque souvent une déviation du comportement normal de la plupart des composants et par conséquent une déformation de bon nombre de grandeurs. Il n'est donc pas possible de raisonner à partir des déviations du comportement normal d'un système.
- (7) Les pannes multiples ne sont pas des occurrences rares. En effet, la panne d'un composant peut provoquer la destruction d'autres composants. Ce phénomène est appelé une cascade de fautes. Par exemple, le court-circuitage de la résistance R du circuit illustré figure 1.2 engendre une augmentation du courant traversant la lampe L, qui provoque sa destruction.

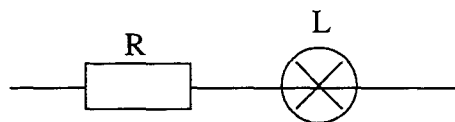


Figure 1.2 : Exemple de pannes en cascade

Le diagnostic de systèmes analogiques soulève donc de nombreux problèmes. Quelques uns sont naturellement surmontés par certaines des techniques de diagnostic que nous présentons maintenant.

1.2 Les techniques de diagnostic

Le but de ce paragraphe n'est pas de comparer les techniques de diagnostic, mais plutôt de les énumérer. Des comparaisons entre ces techniques sont présentées dans la littérature, notamment dans [Hamscher, 88] et [Abu-Hanna, 96].

1.2.1 Les systèmes experts

Les systèmes experts ne sont pas uniquement dévolus au diagnostic. Toutefois, le diagnostic est un de leur domaine d'application privilégié.

Le principe est de coder les connaissances des experts sous forme de règles, ces règles forment la *base de connaissances*, puis de simuler le raisonnement de l'expert, ce qui est réalisé par un mécanisme de déduction appelé *moteur d'inférence*.

La qualité du diagnostic fourni par les systèmes experts dépend principalement de la qualité des déductions et de la valeur des informations traitées. La puissance déductive est localisée dans le "moteur d'inférence" dont les techniques sont assez bien maîtrisées. En revanche, la difficulté majeure est la construction de la base de connaissances qui, traditionnellement, est construite avec des relations cause / effet, connaissances aussi appelées connaissances empiriques. Par nature, ces connaissances sont intimement liées au processus diagnostiqué. Ceci soulève deux problèmes :

- Le moindre changement dans le processus nécessite une reconstruction de la base de connaissances dans sa totalité.
- La création de la base de connaissances nécessite une certaine accumulation de connaissances par des experts. Or, ceci est une limitation car la durée de vie de processus industriels est parfois plus courte que celle nécessaire pour accumuler les connaissances (ce qui augmente d'autant plus la difficulté de création de la base de connaissances).

Par contre, il existe des domaines où le lien entre les connaissances et le système est naturel. C'est le cas entre autre du diagnostic médical. Un système expert bien connu dans ce domaine est MYCIN [Shortliffe, 76].

1.2.2 Les arbres de décision

Le principe est de construire un arbre décrivant la séquence des observations et les "déductions" pouvant être réalisées à partir de ces observations. Comme dans le cas des systèmes experts, les arbres sont créés à partir de connaissances empiriques. Ceci aboutit aux mêmes inconvénients : les arbres sont difficiles à écrire et ils sont fortement liés au processus diagnostiqué.

1.2.3 Les dictionnaires de fautes

Le principe est de construire un dictionnaire où à chaque symptôme, c'est à dire la différence entre les comportements normal et réel du système ou la forme d'une observation, est associée une liste de pannes possibles. La construction de ce dictionnaire est réalisée par simulation du comportement du système pour un ensemble de pannes prédéfinies. Chaque simulation permet d'obtenir un couple (panne, symptômes). Après avoir réalisé cette opération pour toutes les pannes prédéfinies, les couples sont ensuite inversés afin d'obtenir des couples (symptôme, pannes). L'inconvénient de ce système est qu'il nécessite une définition au préalable des différentes pannes possibles. Or, il en existe un nombre important dans le cas des systèmes analogiques. De plus, ces systèmes n'opèrent que pour les valeurs exactes des paramètres et non pas pour des intervalles. La dispersion des valeurs des paramètres ne peut donc pas être prise en considération. Enfin, en dehors des pannes prédéfinies, aucun diagnostic n'est possible. Ceci limite le nombre de pannes détectables et notamment les pannes multiples.

Les principaux défauts de ces systèmes pour le diagnostic de processus analogiques sont leur dépendance vis-à-vis du processus diagnostiqué et leur limitation à un nombre fini de pannes. Le diagnostic à base de modèles permet de surmonter ces problèmes.

1.3 Le diagnostic à base de modèles

Le diagnostic à base de modèles, introduit en 1976 par de Kleer avec le système INTER pour les circuits électroniques analogiques [de Kleer, 76] et Brown avec le système WATSON pour les récepteurs de radios [Brown, 76], repose uniquement sur la connaissance

- du comportement des composants,
- de la structure du processus diagnostiqué, c'est à dire de la liste des composants du processus et des connections entre les composants.

L'utilisation de telles connaissances a permis une réelle avancée dans le domaine du diagnostic :

- les systèmes de diagnostic sont indépendants des processus diagnostiqués,
- l'effort à fournir pour modéliser les systèmes est limité, notamment dans certains domaines tels que l'électronique où le comportement des composants est assez bien défini¹.

Le principe du diagnostic à base de modèles est de localiser les pannes en vérifiant la cohérence du comportement modélisé du système avec les observations. Schématiquement, ceci peut être vu comme la réalisation de trois étapes successives :

- (1) La génération d'une hypothèse concernant l'état de fonctionnement du système. Typiquement, ceci revient à supposer que certains composants sont défectueux.
- (2) La simulation du comportement du système à partir des modèles des composants, étape aussi appelée prédiction.
- (3) La comparaison du résultat de la simulation, appelé *comportement prédit*, au *comportement réel* du système, décrit par les observations (i.e. les mesures des grandeurs).

Les trois étapes ci dessus sont une vue schématique du système. La génération d'hypothèse est en fait imbriquée dans les deux autres étapes, ce qui permet de limiter le nombre d'hypothèses générées. C'est aussi la raison pour laquelle cette étape n'est pas présentée sur le schéma illustré figure 1.3.

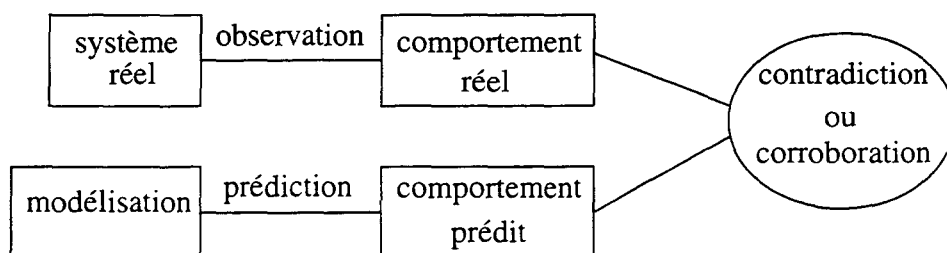


Figure 1.3 : Vue schématique du diagnostic à base de modèles

1. En électronique, la majorité des modèles des composants ont été définis pour la simulation du comportement des systèmes fonctionnant normalement. Les travaux réalisés par [Brown et al, 82] ont pointé sur la limitation de certains de ces modèles dans le cadre du diagnostic, en particulier ceux correspondant à des composants complexes tels que le transistor. L'effort de modélisation de tels composants est plus important car il nécessite l'utilisation de connaissances d'experts en diagnostic [Dague et al, 90a].

Deux approches permettant de prédire les diagnostics sont proposées dans la littérature.

- Le diagnostic peut être déduit à partir des corroborations entre les comportements prédit et réel. C'est le *diagnostic abductif*. Le diagnostic est l'hypothèse réalisée sur l'état de fonctionnement du système, hypothèse qui permet d'expliquer le comportement réel du système. Cette méthode, formalisée par l'intermédiaire de la théorie des ensembles [Reggia et al, 83] et en terme logique [Poole, 90], [Console et al, 90], repose essentiellement sur la connaissance de tous les modèles de fautes (i.e. modélisation du comportement des composants en panne) car elle nécessite la simulation du comportement du système lorsque celui ci est défectueux.
- Le diagnostic peut être déduit à partir des contradictions entre les comportements prédit et réel. C'est le *diagnostic fondé sur la cohérence*. Le diagnostic correspond à une déduction logique : toute contradiction permet de montrer que l'hypothèse réalisée sur l'état de fonctionnement du système est absurde. Cette méthode repose essentiellement sur la connaissance des modèles de bon fonctionnement des composants. Dans ce cadre, l'hypothèse est le bon fonctionnement d'un ensemble de composants C, son absurdité (i.e. toute contradiction) permet de montrer qu'au moins un des composants de C est défectueux. L'ensemble C est alors appelé *ensemble de conflits*. La formalisation de cette approche et la déduction des diagnostics à partir des ensembles de conflits a été proposée en 1987 par Reiter [Reiter, 87]. L'intégration des modèles de fautes [Struss et al, 89], [de Kleer et al, 89], [de Kleer et al, 92b] permet de limiter le nombre de diagnostics (i.e. tout sur-ensemble d'un diagnostic n'est plus nécessairement un diagnostic), mais ne permet pas de réduire la taille des diagnostics minimaux, c'est à dire de mieux localiser les pannes. Le système de diagnostic le plus connu reposant sur cette approche est le General Diagnosis Engine (GDE) de [de Kleer et al, 87].

Ces approches, comparées par [Console et al, 91a] et [Poole, 88], ne sont pas exclusives. Elles ont d'ailleurs été intégrées dans un même paradigme par [Console et al, 91b]. Leur utilisation simultanée autorise un diagnostic avec des connaissances partielles [Ayeb et al, 90].

Le diagnostic à base de modèles permet de surmonter bon nombre de difficultés inhérentes au diagnostic de systèmes analogiques :

- Les pannes peuvent être localisées en connaissant uniquement les modèles de bon fonctionnement.
- L'utilisation de la simulation facilite le diagnostic des systèmes partiellement observés et fortement rebouclés.
- Le diagnostic à base de modèles autorise la détection
 - de fautes multiples. Les liens entre les fautes (i.e. les pannes en cascade) peuvent parfois être pris en considération [Tatar, 96].
 - de fautes dans la structure [Böttcher, 95], [Davis, 84].

Le problème du diagnostic à base de modèles réside dans la modélisation [Dague et al, 87], [Struss, 91]. D'un côté, elle doit être assez précise afin de détecter des incohérences. De l'autre côté, elle doit tenir compte de l'imprécision des mesures et des modèles des composants afin de respecter l'hypothèse fondamentale : les observations et les modèles sont corrects.

Un axe de recherche consiste à relâcher cette hypothèse en considérant par exemple des modèles de précision différente [Struss, 89], [ten Teije et al, 96].

Une autre possibilité consiste à améliorer la modélisation (surtout celle des grandeurs) afin de faciliter la détection d'incohérence. Les travaux présentés dans cette thèse s'inscrivent dans cette voie. Nous décrivons maintenant quelques systèmes de diagnostic de processus analogiques et les modélisations attenantes.

1.4 Etude de systèmes existants

Afin de prendre en compte l'imprécision des mesures et des paramètres des composants, de nombreux systèmes reposent sur une modélisation qualitative.

Une première approche, proposée par [Dague et al, 87] avec le système DEDALE consiste à comparer les comportements réel et normal du processus diagnostiqué¹. L'idée est que toute panne du système engendre une déviation significative du comportement normal du système. Malheureusement, cette hypothèse s'est avérée inexacte lors de l'expérimentation de DEDALE ; dans bon nombre de cas (25%), une panne a simplement engendré une distorsion du comportement sans changement qualitatif notable.

Un autre système reposant sur la comparaison des comportements est décrit dans [Malik et al, 96]. Toutefois, la référence n'est plus nécessairement le comportement normal du système et les aspects dynamiques des processus sont pris en compte. Les résultats obtenus s'avèrent meilleurs, notamment grâce à l'utilisation de modèles de fautes. Les auteurs ne situent pas l'avantage de ce système dans sa capacité à détecter des contradictions, mais plutôt dans sa rapidité d'exécution. Ceci s'expliquant par l'utilisation d'une procédure ne reposant pas sur la simulation.

Une deuxième approche consiste à modéliser l'évolution des grandeurs. Les systèmes reposant sur cette modélisation sont en règle générale basés sur l'algorithme de simulation QSIM de [Kuipers, 86]. C'est le cas des travaux de [Pan, 84], MIDAS de [Oyeleye et al, 90], MIMIC de [Dvorak et al, 89], DATMI de [DeCoste, 90], IDS de [Ng, 90]². Une grandeur est représentée par une suite d'états qualitatifs correspondant alternativement à un instant et à un intervalle de temps. Un état qualitatif est un couple (qval, qdir) où qval est un réel ou un intervalle de réels représentant la valeur de la grandeur et qdir est un des symboles {inc, std, dec} signifiant respectivement que la grandeur croît, est stable ou décroît. Un exemple de modélisation quantitative d'une grandeur est illustré figure 1.4

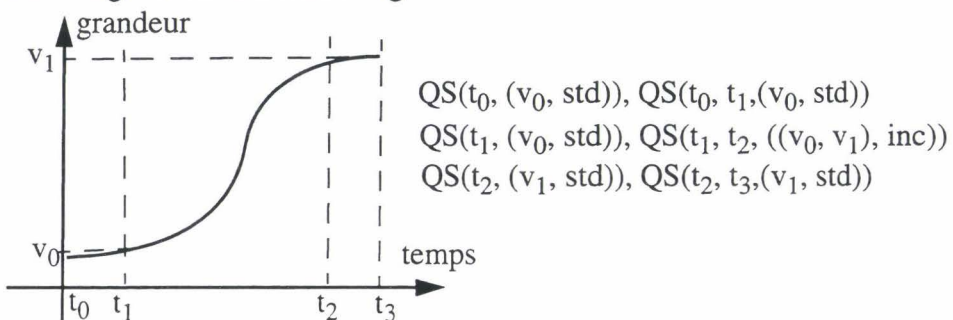


Figure 1.4 : Une modélisation qualitative d'une grandeur

L'avantage de cette modélisation est que l'algorithme de simulation est rapide. Comme dans le cas des ordres de grandeurs, la détection d'une contradiction nécessite un changement

1. Le formalisme utilisé pour comparer ces grandeurs est celui des ordres de grandeur de [Raiman, 86].
2. Ces systèmes ne sont pas tous directement issus de la modélisation de Kuipers, toutefois les modélisations proposées sont très proches.

qualitatif du comportement du processus. Cet inconvénient, ainsi que celui du à l'ambiguïté introduite lors de la simulation, ont conduit à l'intégration de quelques informations quantitatives dans certains systèmes (c'est le cas de MIMIC, MIDAS et DATMI).

Une troisième approche, proposée par [Leyval et al, 91], consiste à modéliser les variations des grandeurs par des segments de droite. Un exemple est illustré figure 1.5

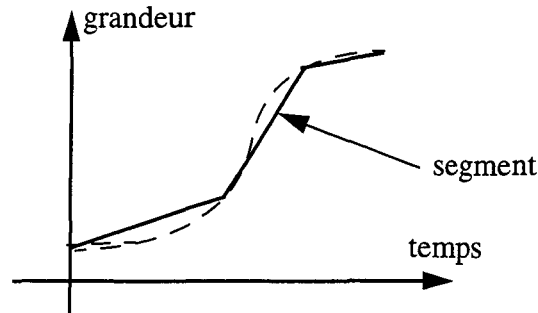


Figure 1.5 : Modélisation qualitative de Leyval

Cette modélisation, plus complexe que celles définies précédemment, permet de mieux caractériser les variations des grandeurs. Toutefois, les modèles utilisés dans le cadre du diagnostic sont les réponses des systèmes à une impulsion, c'est à dire les fonctions de transfert (elles mêmes approximées par des fonctions affines).

D'une manière générale, l'ambiguïté et l'imprécision des modélisations qualitatives rendent difficile la détection de contradictions. C'est pourquoi les approches basées sur une modélisation quantitative nous semble plus pertinentes dans le domaine du diagnostic de systèmes analogiques.

Une première approche a été proposée par [Brown et al, 82]¹ avec SOPHIE III pour le diagnostic de circuits électroniques analogiques. Afin de prendre en compte l'imprécision des mesures, la valeur des grandeurs est encadrée par des intervalles. Les résultats obtenus sont concluants mais pointent sur deux limitations :

- une certaine lenteur du système due à l'utilisation d'intervalles,
- seuls les aspects statiques des circuits sont considérés : les grandeurs sont supposées constantes au cours du temps et les modèles des composants ne font pas intervenir la dérivée.

Depuis, des solutions permettant de surmonter ces limitations ont été proposées.

La rapidité des systèmes basés sur une modélisation par des intervalles a été améliorée, non seulement grâce à l'évolution de la puissance de calcul des ordinateurs, mais aussi grâce à l'élaboration de nouveaux algorithmes de prédiction et de détection de contradiction (CATS de [Dague et al, 90b] et ACP de [Hamscher, 91a]).

Dans CATS/DIANA de [Dague et al, 90c], les aspects dynamiques des processus sont pris en compte

- en modélisant les grandeurs par des tableaux d'intervalles ; les intervalles correspondant aux échantillons du processus de mesure.
- en encadrant la dérivée des grandeurs.

1. Une présentation plus complète de SOPHIE III est aussi proposée dans [de Kleer et al, 92a].

Cette modélisation peut être vue comme une extension du modèle de [Brown et al, 82]. En effet, l'aspect dynamique des processus est diagnostiqué comme une suite d'états statiques.

CATS a été expérimenté sur de nombreux circuits de taille et de complexité variées, tant en électronique [Dague et al, 91], qu'en électricité [Devès et al, 92] ou en génie chimique (projet Esprit 3 HINT #6447) [Alarcon et al, 94].

Les résultats ont été suffisamment satisfaisants pour autoriser le démarrage de l'industrialisation du programme pour le diagnostic de circuits électroniques analogiques. Néanmoins, les différentes expérimentations menées ont mis en évidence deux difficultés, présentées ainsi par un des concepteurs de CATS [Loiez et al, 96b] :

- *La première concerne la représentation des grandeurs par des tableaux d'intervalles qui conduit à un volume d'informations considérable (plusieurs centaines d'échantillons par grandeur) qui n'est pas utilisé de manière optimale. En effet, un même conflit est souvent présent sur un grand nombre d'échantillons alors que le traitement d'un seul de ces échantillons aurait permis de le détecter, réduisant d'autant le coût du traitement. A l'opposé, il est des cas où la situation contraire se produit (conflit présent sur quelques échantillons seulement). L'équi-importance accordée à chacun des échantillons représentant les grandeurs du système est donc une cause d'inefficacité.*
- *La seconde est liée à l'utilisation de la dérivée comme unique moyen de représentation des propriétés temporelles des composants. Cela nécessite de pouvoir effectuer une approximation numérique de la dérivée des signaux observés ou prédits. Or, en général, aucune information n'est disponible sur la fonction à dériver ; c'est en particulier le cas pour un circuit en panne où, du fait de la panne, n'importe quel signal peut apparaître. Il est dans ce cas théoriquement impossible d'encadrer correctement la dérivée et pratiquement très difficile de trouver une procédure acceptable ; cela nous a conduit non seulement à élargir bien plus que nécessaire les intervalles de la dérivée mais également à éliminer les échantillons pour lesquels le calcul est trop hasardeux (variation rapide du signal). Cela a pour conséquence la non-détection de certaines contradictions.*

Pour remédier au problème du manque de concision des informations, une solution aurait été de recourir aux propagateurs temporels tels que TCP [Williams, 86] et TINT [Hamscher, 90a] en représentant un signal par une suite continue d'épisodes, chaque épisode représentant la valeur de la grandeur pendant une période temporelle. Les concepteurs de CATS n'ont pas donné suite à cette idée par crainte de l'inévitable multiplication du nombre d'épisodes lors de la propagation des grandeurs.

Le problème du calcul de la dérivée est plus difficile à résoudre. Ceci est directement lié à la procédure d'échantillonnage des appareils de mesure. Le théorème de Shannon assure que la reconstruction correcte d'un signal nécessite une fréquence d'échantillonnage au moins deux fois supérieure à la plus rapide des composantes de Fourier du signal. Or, ceci est irréaliste, surtout dans le cadre du diagnostic [Dague et al, 96]. Afin de résoudre ce problème, [Bos-Plachez, 96] propose d'utiliser la logique floue pour réduire la largeur des intervalles. Bien que permettant la détection d'un plus grand nombre de pannes, cette proposition apporte uniquement des solutions partielles.

Les travaux proposés dans cette thèse se situent dans la continuité de CATS. L'étude porte sur une nouvelle modélisation permettant d'apprôter des solutions à ces deux problèmes.

1.5 Propositions

Nous partons de deux hypothèses jugées satisfaisantes lors de l'expérimentation de CATS :

- Aucune panne ne se déclare pendant la période d'observation du système¹,
- Le comportement des composants est connu et est décrit par des équations différentielles.

Nous proposons d'encadrer un signal sur une fenêtre temporelle par deux fonctions polynomiales par morceaux. Les polynômes et la fenêtre temporelle sont appelés bande temporelle. Une Suite de Bandes Temporelles (SBT) est une suite de bandes temporelles contiguës. Un exemple est illustré figure 1.6.

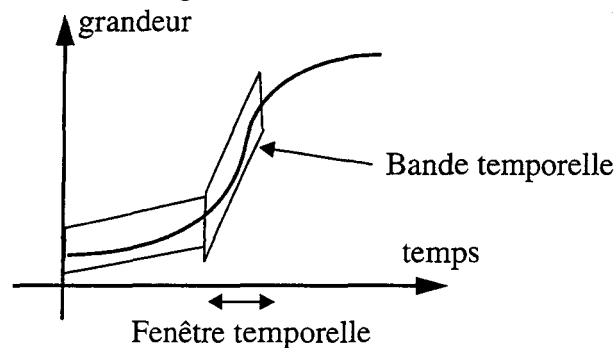


Figure 1.6 : Modélisation par des bandes temporelles

Cette modélisation peut être vue comme une extension de TCP [Williams, 86]. Elle hérite d'ailleurs des propriétés de concision des informations. Toutefois, l'intérêt principal des SBT est qu'elles permettent d'éviter le calcul de la dérivée des grandeurs lors du diagnostic des systèmes.

La démarche est l'inverse de celle de CATS qui pour comparer une grandeur et sa dérivée propose de déduire la dérivée à partir de la grandeur alors que nous proposons d'intégrer la grandeur. Cette démarche est illustrée schématiquement figure 1.7.

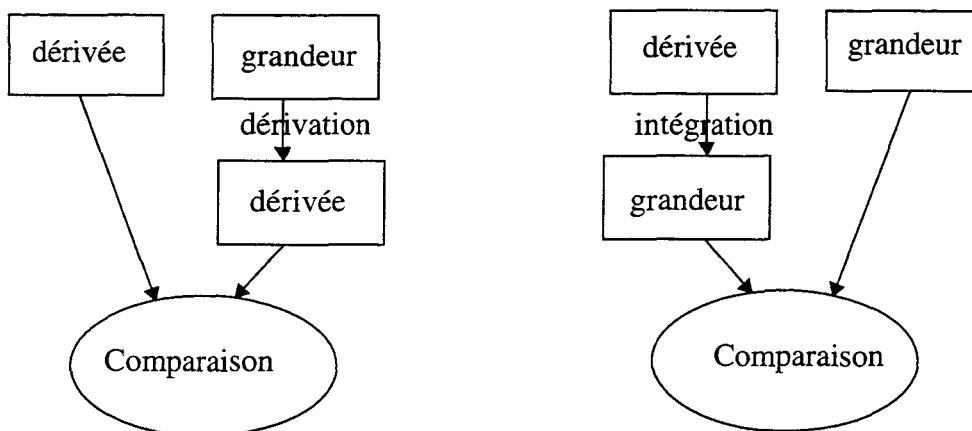


Figure 1.7 : Comparaison des démarches

1. Cette hypothèse, réalisée dans le cadre de DIANA, a ensuite été relâchée par [Alarcon et al, 94].

Cette thèse est consacrée à l'utilisation des suites de bandes temporelles pour le diagnostic de systèmes analogiques.

La deuxième partie est consacrée à la formalisation des Suites de Bandes Temporelles (SBT). Après avoir défini quelques propriétés des SBT, cette partie est essentiellement consacrée à l'étude de l'arithmétique des SBT. Les opérations d'addition, de multiplication et d'intégration, appelées opérations primitives, sont d'abord définies. Par analogie avec l'arithmétique des intervalles [Moore, 66], une généralisation est ensuite proposée.

La troisième partie est consacrée à l'étude des procédures de détection des conflits. Cette étude ne s'inscrit pas dans le cadre de GDE, mais plutôt dans celui de la théorie du diagnostic à partir des principes premiers [Reiter, 87]. Le comportement d'un composant ou d'un ensemble de composants est décrit par une équation différentielle. Les algorithmes de détection d'ensembles de conflits consistent à vérifier la cohérence d'un modèle (i.e. d'une équation différentielle) avec les observations.

La quatrième partie est consacrée à l'intégration des SBT dans des systèmes de diagnostic. Une intégration (théorique) des SBT dans GDE est proposée. Les limites d'un tel système sont alors étudiées : la plus importante étant la limitation aux équations différentielles d'ordre 1. Une autre système, DOGS, est ensuite proposé : il consiste à compiler un ensemble de modèles, puis à vérifier un par un la cohérence des modèles générés avec les observations.

Les résultats obtenus avec une implémentation partielle du deuxième système sont ensuite présentés.

PARTIE I

Modéliser avec des Bandes Temporelles



Introduction

Nous proposons dans cette partie une modélisation permettant de représenter des grandeurs, ou quantités physiques, évoluant continûment dans le temps.

La notion de grandeur est d'abord abstraite et est remplacée par la notion de fonction temporelle continue. Pour modéliser ces fonctions, nous introduisons la notion de bandes temporelles. Représenter une fonction par une bande temporelle consiste à définir deux polynômes encadrant cette fonction sur un intervalle de temps. Afin de limiter le degré des polynômes, la représentation par des bandes temporelles est étendue à la représentation par des suites de bandes temporelles.

Ensuite, certaines relations entre les suites de bandes temporelles, et plus particulièrement l'égalité, sont étudiées. Etant donnée une relation entre des suites de bandes temporelles, le but est d'établir les relations existant entre les fonctions représentées par ces suites.

Nous nous proposons alors de définir quelques opérations sur les suites de bandes temporelles. Notre objectif est de déterminer une suite de bandes temporelles représentant une opération entre des fonctions à partir de la représentation de ces fonctions.

Dans un premier temps, les opérations d'addition, de multiplication et d'intégration sont définies. Deux propriétés de ces opérations sont particulièrement étudiées :

- la conservation de la représentation ; cette propriété garantit que toute fonction résultant d'une opération est correctement représentée ;
- la non introduction d'imprécision ; cette propriété garantit que toute suite de bandes temporelles résultant d'une opération entre des suites de bandes temporelles représente uniquement les fonctions que cette suite est censée représenter.

Enfin, par analogie avec l'arithmétique des intervalles et à partir des trois opérations déjà définies, une extension à l'ensemble des opérations sur les suites de bandes temporelles est proposée. Certaines propriétés des opérations sur les intervalles sont alors utilisées pour démontrer les propriétés des opérations sur les suites de bandes temporelles. Nous proposons alors une meilleure caractérisation des opérations polynomiales sur les suites de bandes temporelles, puis une extension aux opérations non polynomiales est proposée.

Chapitre 2

Modélisation des observations

Nous nous proposons d'aborder dans ce chapitre la modélisation des observations. Les observations que nous prenons en considération sont définies sur une fenêtre temporelle unique et concernent une grandeur évoluant continûment dans le temps telle que celle présentée sur l'exemple de la figure 2.1.

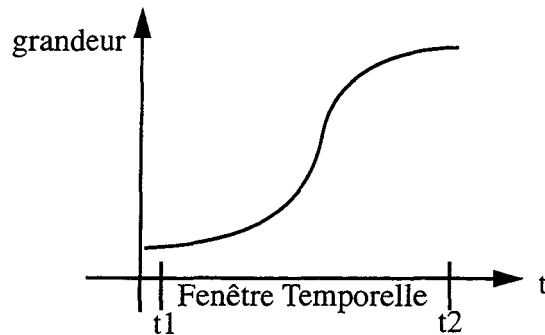


Figure 2.1 : Exemple d'observation

Dans la suite de ce chapitre, toute grandeur sera représentée par une fonction temporelle continue $g(t)$.

Nous notons \mathbf{G} l'ensemble des fonctions continues de \mathbf{R} dans \mathbf{R} , où \mathbf{R} est l'ensemble des nombres réels.

2.1 Modélisation à l'aide de bandes temporelles

Nous définissons tout d'abord la notion de bande temporelle destinée à encadrer une grandeur entre deux instants t_i, t_s .

Définition 2.1 : “Bande temporelle”

Une bande temporelle est un quadruplet (t_i, t_s, p_i, p_s) , où t_i et t_s sont des nombres réels et p_i et p_s sont des polynômes, tels que :

$$t_i < t_s \wedge \forall t \in [t_i, t_s[, p_i(t) \leq p_s(t)$$

Une bande temporelle est aussi un ensemble de couples de réels (x, y) où x et y appartiennent respectivement aux intervalles $[t_i, t_s[$ et $[p_i(x), p_s(x)]$.

D'une manière formelle :

$$(t_i, t_s, p_i, p_s) = \{ (x, y) / t_i \leq x < t_s, p_i(x) \leq y \leq p_s(x) \}$$

Soit $B = (t_i, t_s, p_i, p_s)$ une bande temporelle. On appellera :

- $[t_i, t_s[$ l'extension temporelle de B ,
- $t_i(B) = t_i$ (resp. $t_s(B) = t_s$) l'instant de début (resp. de fin) de l'extension temporelle,
- p_i le polynôme inférieur de B ,
- p_s le polynôme supérieur de B ,
- $d^0 B = \max(d^0(p_i), d^0(p_s))$, le degré de B (où $d^0(p)$ est le degré du polynôme p ,
- $B(t_0) = [p_i(t_0), p_s(t_0)]$, la valeur de B à l'instant t_0 (appartenant à $[t_i, t_s[$). Par extension, on note $B(t_s) = [p_i(t_s), p_s(t_s)]$. (B n'est pas définie à l'instant t_s , par contre les polynômes p_i et p_s le sont.)

Le symbole \in sera utilisé dans son sens habituel de la théorie des ensembles. Ainsi, un couple (x, y) appartient à une bande temporelle (t_i, t_s, p_i, p_s) ssi il existe un couple (x', y') de la bande temporelle égal à (x, y) . De plus, deux couples (x, y) et (x', y') sont égaux ssi x est égal à x' et y est égal à y' .

La définition suivante découle directement de cette remarque.

Définition 2.2 : "Appartenance à une bande temporelle"

Un couple de réels (x, y) appartient à une bande temporelle (t_i, t_s, p_i, p_s) ssi x et y appartiennent respectivement aux intervalles $[t_i, t_s[$ et $[p_i(x), p_s(x)]$.

D'une manière formelle :

$$(x, y) \in (t_i, t_s, p_i, p_s) \Leftrightarrow x \in [t_i, t_s[\wedge y \in [p_i(x), p_s(x)]$$

Notons que, pour tout couple (x, y) appartenant à une bande temporelle, la valeur de y est indépendante de la valeur de x ; par contre, les bornes de l'intervalle encadrant y dépendent de x (et des polynômes de la bande temporelle).

Une bande temporelle délimite une surface du plan $(t, y(t))$ par les courbes d'équation

$$y(t) = p_i(t), \quad y(t) = p_s(t), \quad t = t_i, \quad t = t_s.$$

Un exemple de bande temporelle de degré 1 est présenté figure 2.2.

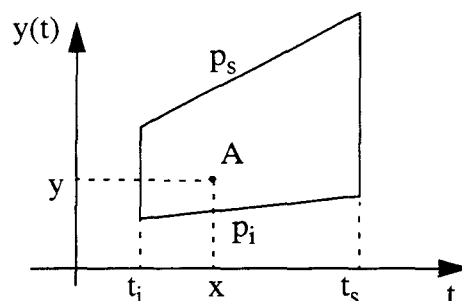


Figure 2.2 : Exemple de bande temporelle

Le point A de coordonnées (x, y) appartient à la bande temporelle (t_i, t_s, p_i, p_s) .

L'existence de cette surface est garantie par les contraintes de la définition des bandes temporelles (Définition 2.1) ; en d'autres termes, pour tout instant x appartenant à l'extension temporelle d'une bande temporelle B , il existe au moins un couple (x, y) appartenant à B . On notera \mathcal{B} l'ensemble des bandes temporelles.

La bande temporelle est l'objet mathématique qui va permettre de modéliser les observations, c'est à dire représenter les grandeurs. Nous introduisons maintenant cette notion :

Définition 2.3 : "Représentation d'une grandeur par une Bande Temporelle"

Une grandeur $g(t)$ est représentée par une bande temporelle $B = (t_i, t_s, p_i, p_s)$, et on notera $g(t) \text{ Rep } B$, ssi, pour tout instant t_0 compris entre t_i et t_s , la valeur de $g(t)$ à l'instant t_0 , $g(t_0)$, est comprise entre les réels $p_i(t_0)$ et $p_s(t_0)$.

D'une manière formelle :

$$g(t) \text{ Rep } (t_i, t_s, p_i, p_s) \Leftrightarrow \forall t_0 \in [t_i, t_s[, p_i(t_0) \leq g(t_0) \leq p_s(t_0)$$

Une bande temporelle représentant une grandeur $g(t)$ correspond à un ensemble de valeurs possibles de $g(t)$. En effet, pour chaque couple (x, y) appartenant à B , le réel y correspond à une valeur possible de $g(t)$ à l'instant x (i.e. $g(x)$).

Remarque :

Avec les notations introduites précédemment, la définition formelle précédente (définition 2.3) peut se réécrire de différentes manières :

$$\begin{aligned} g(t) \text{ Rep } B &\Leftrightarrow \forall t_0 \in [t_i, t_s[, (t_0, g(t_0)) \in B \\ &\Leftrightarrow \forall t_0 \in [t_i, t_s[, g(t_0) \in B(t_0) \end{aligned}$$

La figure 2.3 présente un encadrement de la grandeur présentée à la figure 2.1 par deux bandes temporelles de degrés différents :

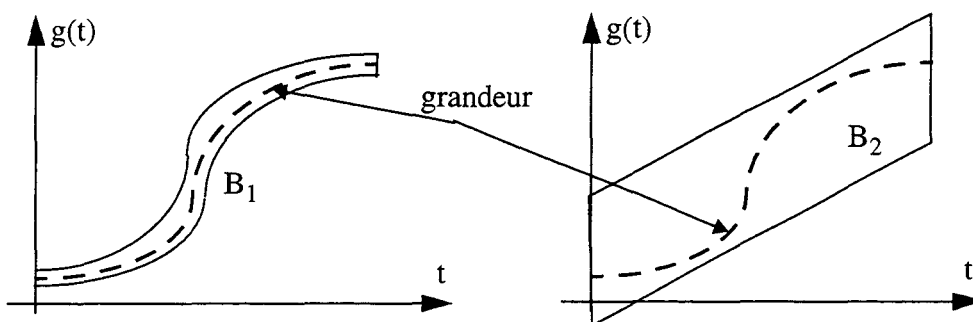


Figure 2.3 : Encadrement d'une grandeur par des bandes temporelles

Encadrer un signal par une bande temporelle consiste donc à borner les valeurs d'une grandeur par deux courbes, dont les équations sont des polynômes. Il existe une infinité de bandes temporelles différentes permettant d'encadrer une grandeur. Chaque bande temporelle constitue un plus ou moins bon encadrement d'une grandeur ; la qualité de l'encadrement d'une grandeur $g(t)$ par une bande temporelle est définie intuitivement de la manière suivante¹

- la qualité d'un encadrement à un instant t_0 est la somme des distances entre $g(t_0)$ et $p_i(t_0)$ et entre $g(t_0)$ et $p_s(t_0)$, c'est à dire par la largeur de la bande (i. e. $p_s(t_0) - p_i(t_0)$).
- la qualité moyenne d'un encadrement sur un intervalle $[t_1, t_2]$ est la somme des qualités instantanées, c'est à dire $\int_{t_1}^{t_2} (p_s(x) - p_i(x))dx$.

Il semble évident qu'une bande temporelle de degré élevé permet de définir un encadrement de meilleure qualité. Par exemple, la bande temporelle B_1 de la figure 2.3 (dont le degré est 3) est un meilleur encadrement de la grandeur $g(t)$ que la bande temporelle B_2 (dont le degré est 1). En règle générale, une bande temporelle de degré élevé est nécessaire pour obtenir une représentation des observations de "bonne qualité". Or, l'augmentation du degré des bandes temporelles va accroître non seulement la difficulté pour créer les bandes temporelles, mais aussi la complexité des opérations sur celles ci (opérations définies plus loin dans cette partie). C'est pourquoi nous introduisons la notion de suite de bandes temporelles.

2.2 Modélisation à l'aide de suites de bandes temporelles

Nous définissons d'abord les suites de bandes temporelles.

Définition 2.4 : "Suite de Bandes Temporelles"

Une suite de bandes temporelles est un ensemble fini ordonné de bandes temporelles $\{B_1, \dots, B_n\}$ tel que :

$$\forall k \in [1, n-1], t_s(B_k) = t_i(B_{k+1}) \wedge B_k(t_s(B_k)) \cap B_{k+1}(t_s(B_k)) \neq \emptyset$$

Pour une suite de bandes temporelles $SB = \{B_1, \dots, B_n\}$, on appellera :

- $[t_i(SB), t_s(SB)[= [t_i(B_1), t_s(B_n)[$, l'extension temporelle de SB,
- $\text{card}(SB) = n$, la taille de SB,
- $SB[k] = B_k$, la $k^{\text{ième}}$ bande de SB,
- $d^0(SB) = \max(d^0(B_1), \dots, d^0(B_n))$, le degré de SB,
- $SB(t_0) = B_k(t_0)$, la valeur de SB à l'instant t_0 (t_0 étant un instant appartenant à l'extension temporelle de B_k). ($SB(t_0)$ est définie de manière unique car, par définition des suites de bandes temporelles, l'instant t_0 appartient à l'extension temporelle d'une et une seule bande temporelle de SB).

Afin de simplifier l'écriture, les notations SBT et Suites de Bandes Temporelles seront utilisées indifféremment.

Une suite de bandes temporelles sera dite continue lorsque les fonctions enveloppes inférieure et supérieure, constituées respectivement par les polynômes inférieurs et supérieurs des bandes temporelles, sont continues (i.e. $B_k(t_s(B_k)) = B_{k+1}(t_s(B_k))$)

1. Ces notions sont définies de manière informelle car elles ne seront plus utilisées par la suite.

Par exemple, la figure 2.4 illustre deux suites de bandes temporelles ; seule SB_2 est continue.

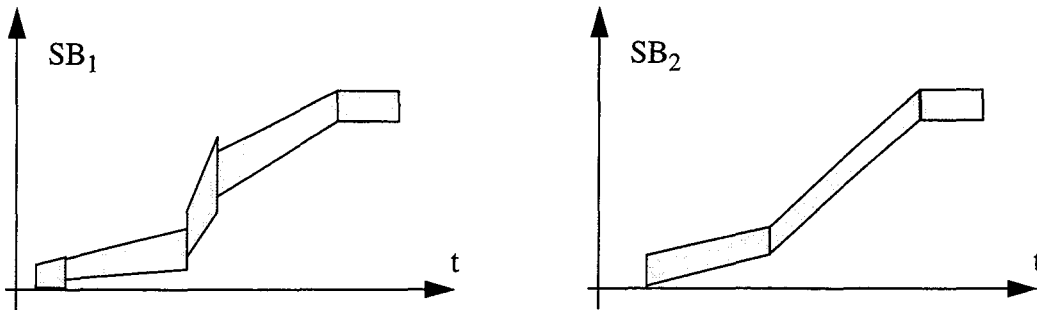


Figure 2.4 : Exemple de suites de bandes temporelles

La définition de l'appartenance à une bande temporelle est étendue aux suites de bandes temporelles.

Définition 2.5 : “Appartenance à une suite de bandes temporelles”

Un couple de réels (x, y) appartient à une suite de bandes temporelles SB ssi (x, y) appartient à une des bandes temporelles de SB .

D'une manière formelle :

$$(x, y) \in SB \Leftrightarrow \exists B \in SB, (x, y) \in B$$

De même, nous étendons la définition de la représentation d'une grandeur par une bande temporelle.

Définition 2.6 : “Représentation d'une grandeur par une suite de bandes temporelles”

Une grandeur $g(t)$ est représentée par une suite de bandes temporelles SB , et nous noterons $g(t) \text{ Rep } SB$, ssi la grandeur $g(t)$ est toujours représentée par une des bandes temporelles de SB .

D'une manière formelle :

$$g(t) \text{ Rep } SB$$

$$\Leftrightarrow \forall t_0 \in [t_i(SB), t_s(SB)[, \exists (t_i, t_s, p_i, p_s) \in SB \text{ tel que } t_0 \in [t_i, t_s[\wedge p_i(t_0) \leq g(t_0) \leq p_s(t_0)$$

La contrainte sur les valeurs des bandes temporelles appartenant à une suite ($B_k(t_s(B_k)) \cap B_{k+1}(t_s(B_k)) \neq \emptyset$) nous permet d'assurer que toute suite de bandes temporelles représente au moins une fonction temporelle continue.

Remarque :

La définition formelle précédente (définition 2.6) peut aussi se réécrire :

$$g(t) \text{ Rep } SB \Leftrightarrow \forall t_0 \in [t_i(SB), t_s(SB)[, \exists B \in SB \text{ tel que } t_0 \in [t_i(B), t_s(B)[\wedge (t_0, g(t_0)) \in SB$$

ou

$$g(t) \text{ Rep } SB \Leftrightarrow \forall t_0 \in [t_i(SB), t_s(SB)[, (t_0, g(t_0)) \in SB(t_0)$$

La grandeur représentée figure 2.3 peut être représentée par la suite de bandes temporelles suivante :

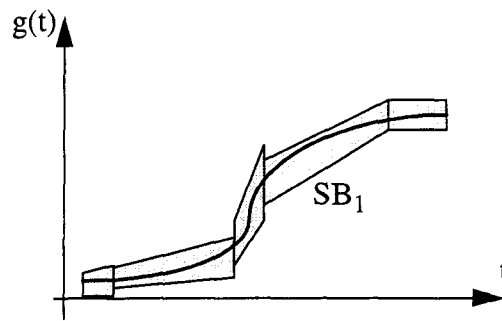


Figure 2.5 : Représentation d'une grandeur par une suite de bandes temporelles

Nous nous limitons dans ce chapitre à définir formellement la représentation d'une grandeur par une suite de bandes temporelles. Il est évident que toute grandeur peut être représentée par une suite de bandes temporelles et que la création de cette suite dépend des données disponibles sur la grandeur (fonction temporelle, valeurs exactes ou imprécises des grandeurs en différents instants, ...). Nous supposons dans la suite de cette partie que les suites de bandes temporelles représentant les grandeurs sont connues.

Chapitre 3

Relations entre les suites de bandes temporelles

Les bandes temporelles sont définies comme des quadruplets, mais aussi comme des ensembles de couples de réels.

La première relation définie est l'égalité de deux bandes temporelles. Cette définition est issue de la définition usuelle de l'égalité de deux quadruplets, mais elle correspond à la définition usuelle de l'égalité de deux ensembles.

La relation d'égalité est ensuite étendue naturellement aux suites de bandes temporelles. Toutefois, cette définition ne correspond plus à la définition usuelle de l'égalité de deux ensembles. C'est pourquoi, nous introduisons la notion de suites de bandes temporelles équivalentes dont la propriété principale est de correspondre à la représentation des mêmes fonctions temporelles.

L'ensemble des suites équivalentes forme une classe d'équivalence. Nous caractérisons un élément, celui de plus petite taille.

Les relations d'inclusion, d'intersection et d'ordre sont ensuite définies.

3.1 Egalité de deux bandes temporelles

Définition 3.1 : *“Egalité de deux bandes temporelles”*

Deux bandes temporelles sont égales ssi

- leurs extensions temporelles sont égales,
- leurs polynômes sont égaux.

D'une manière formelle :

$$(t_{i1}, t_{s1}, p_{i1}, p_{s1}) = (t_{i2}, t_{s2}, p_{i2}, p_{s2}) \Leftrightarrow t_{i1} = t_{i2} \wedge t_{s1} = t_{s2} \wedge p_{i1} = p_{i2} \wedge p_{s1} = p_{s2}$$

La définition de l'égalité de deux bandes correspond à la définition “classique” de l'égalité de deux quadruplets : les éléments de même rang de chaque quadruplet sont égaux. Toutefois, une bande temporelle est aussi définie comme étant un ensemble de couples de réels. La définition de l'égalité de deux bandes temporelles aurait donc pu aussi provenir de la définition “classique” de l'égalité de deux ensembles : deux ensembles sont égaux ssi tout élément de l'un appartient à l'autre et inversement. Nous montrons maintenant que, pour les bandes temporelles, ces deux notions sont équivalentes.

Lemme 3.1 :

Deux bandes temporelles B_1 et B_2 sont égales ssi tout élément (x, y) appartenant à B_1 appartient aussi à B_2 et inversement.

D'une manière formelle :

$$B_1 = B_2 \Leftrightarrow (\forall (x, y) \in \mathbf{R}^2, (x, y) \in B_1 \Leftrightarrow (x, y) \in B_2)$$

Preuve :

Soient deux bandes temporelles $B_1 = (t_{i1}, t_{s1}, p_{i1}, p_{s1})$ et $B_2 = (t_{i2}, t_{s2}, p_{i2}, p_{s2})$.

\Rightarrow

Supposons que $B_1 = B_2$.

$$\begin{aligned} (x, y) \in B_1 &\Leftrightarrow x \in [t_{i1}, t_{s1}[\wedge y \in [p_{i1}(x), p_{s1}(x)] \\ &\Leftrightarrow x \in [t_{i2}, t_{s2}[\wedge y \in [p_{i2}(x), p_{s2}(x)] \text{ (car } B_1 = B_2) \\ &\Leftrightarrow (x, y) \in B_2 \end{aligned}$$

\Leftarrow

Raisonnons par l'absurde. Supposons que tout couple appartenant à B_1 appartienne aussi à B_2 et inversement et que $B_1 \neq B_2$.

D'après la définition de la différence de deux bandes temporelles, on a

$$t_{i1} \neq t_{i2} \vee t_{s1} \neq t_{s2} \vee p_{i1} \neq p_{i2} \vee p_{s1} \neq p_{s2}.$$

$$\text{Or, } t_{i1} \neq t_{i2} \Rightarrow t_{i1} < t_{i2} \vee t_{i1} > t_{i2}$$

$$\begin{aligned} &\Rightarrow [(t_{i1}, p_{i1}(t_{i1})) \in B_1 \wedge (t_{i1}, p_{i1}(t_{i1})) \notin B_2] \vee [(t_{i2}, p_{i2}(t_{i2})) \notin B_1 \wedge (t_{i2}, p_{i2}(t_{i2})) \in B_2] \\ &((t_{i1}, p_{i1}(t_{i1})) \notin B_2 \text{ car } t_{i1} < t_{i2} \Rightarrow t_{i1} \notin [t_{i2}, t_{s2}[\end{aligned}$$

Or, ceci est absurde car tout point appartenant à B_1 appartient aussi à B_2 .

Par conséquent, $t_{i1} = t_{i2}$. De la même manière, on montre que $t_{s1} = t_{s2}$.

$$\text{De plus, } p_{i1} \neq p_{i2} \Rightarrow \exists t_0 \in [t_{i1}, t_{s1}[, p_{i1}(t_0) \neq p_{i2}(t_0).$$

(En effet, deux fonctions égales pour une infinité d'instants appartenant à $[t_{i1}, t_{s1}[$ sont égales.

Or, pour les polynômes, l'égalité en tant que fonction implique l'égalité syntaxique)

$$\begin{aligned} \text{D'où } p_{i1} \neq p_{i2} &\Rightarrow \exists t_0 \in [t_{i1}, t_{s1}[, p_{i1}(t_0) < p_{i2}(t_0) \vee p_{i1}(t_0) > p_{i2}(t_0) \\ &\Rightarrow \exists t_0 \in [t_{i1}, t_{s1}[, \quad [(t_0, p_{i1}(t_0)) \in B_1 \wedge (t_0, p_{i1}(t_0)) \notin B_2] \\ &\quad \vee \quad [(t_0, p_{i2}(t_0)) \notin B_1 \wedge (t_0, p_{i2}(t_0)) \in B_2] \end{aligned}$$

Ce qui est absurde, donc $p_{i1} = p_{i2}$. De la même manière, on montre que $p_{s1} = p_{s2}$.

Donc, $t_{i1} = t_{i2}$, $t_{s1} = t_{s2}$, $p_{i1} = p_{i2}$ et $p_{s1} = p_{s2}$. Ce qui est absurde, donc $B_1 = B_2$.

□

Nous étendons maintenant la définition de l'égalité aux suites de bandes temporelles.

3.2 Egalité et équivalence de deux suites de bandes temporelles

Définition 3.2 : "Suites de bandes temporelles égales"

Deux suites de bandes temporelles sont égales ssi elles ont la même taille et les bandes temporelles de ces suites sont égales deux à deux.

D'une manière formelle :

$$SB_1 = SB_2 \Leftrightarrow \text{card}(SB_1) = \text{card}(SB_2) \wedge \forall k \in [1, \text{card}(SB_1)], SB_1[k] = SB_2[k]$$

Deux suites de bandes temporelles égales délimitent la même portion de plan. Par exemple, la figure 3.1 décrit deux suites de bandes temporelles SB_1 et SB_2 égales :

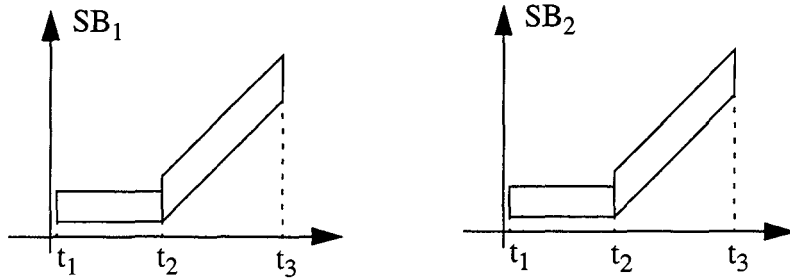


Figure 3.1 : Suites de bandes temporelles égales

Nous avons montré que deux bandes temporelles égales correspondent à des ensembles de couples (x, y) égaux. Cette propriété s'étend naturellement aux SBT : si deux suites de bandes temporelles SB_1, SB_2 sont égales, alors tout couple (x, y) appartenant à SB_1 appartient aussi à SB_2 et inversement. En effet,

$$\begin{aligned} (x, y) \in SB_1 &\Rightarrow \exists B_1 \in SB_1, (x, y) \in B_1 \\ &\Rightarrow B_1 \in SB_2 \wedge (x, y) \in B_1 \text{ (car } SB_1 = SB_2) \\ &\Rightarrow (x, y) \in SB_2 \end{aligned}$$

Par contre, l'inverse n'est pas nécessairement vrai ; deux suites de bandes temporelles différentes peuvent correspondre à des ensembles de couples égaux. C'est le cas par exemple des suites de bandes temporelles SB_3 et SB_4 illustrées figure 3.2.

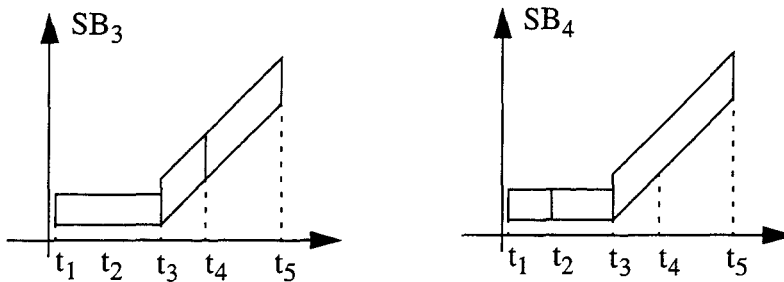


Figure 3.2 : Suites de bandes temporelles différentes et contenant les mêmes couples

Nous définissons formellement deux suites ayant cette propriété :

Définition 3.3 : “Suites de bandes temporelles équivalentes”

Deux suites de bandes temporelles SB_1 et SB_2 sont équivalentes, et on note $SB_1 \sim SB_2$, ssi tout couple (x, y) appartenant à SB_1 appartient aussi à SB_2 , et inversement.

D'une manière formelle :

$$SB_1 \sim SB_2 \Leftrightarrow \forall (x, y) \in \mathbf{R}^2, ((x, y) \in SB_1 \Leftrightarrow (x, y) \in SB_2)$$

Les suites de bandes temporelles SB_3 et SB_4 de la figure 3.2 sont équivalentes. On remarque qu'elles délimitent la même surface du plan à un découpage temporel près.

Il découle directement des définitions de l'égalité et de l'équivalence de deux suites de bandes temporelles que :

- $SB_1 = SB_2 \Rightarrow SB_1 \sim SB_2$
- $SB_1 \sim SB_2 \wedge SB_2 \sim SB_3 \Rightarrow SB_1 \sim SB_3$

Nous caractérisons maintenant les bandes temporelles appartenant à deux suites de bandes temporelles équivalentes.

Lemme 3.2 :

Deux suites de bandes temporelles sont équivalentes ssi leurs extensions temporelles sont égales et les polynômes des bandes temporelles, dont les extensions temporelles s'intersectent, sont égaux.

D'une manière formelle :

$$SB_1 \sim SB_2 \Leftrightarrow [t_i(SB_1), t_s(SB_1)[= [t_i(SB_2), t_s(SB_2)[\\ \wedge \forall (t_{i1}, t_{s1}, p_{i1}, p_{s1}) \in SB_1, \forall (t_{i2}, t_{s2}, p_{i2}, p_{s2}) \in SB_2, \\ [t_{i1}, t_{s1}[\cap [t_{i2}, t_{s2}[= \emptyset \vee (p_{i1} = p_{i2} \wedge p_{s1} = p_{s2})$$

Preuve :

\Rightarrow

Soient deux suites de bandes temporelles équivalentes SB_1, SB_2 .

Raisonnons par l'absurde : supposons que

$$[t_i(SB_1), t_s(SB_1)[\neq [t_i(SB_2), t_s(SB_2)[\\ \vee \exists (t_{i1}, t_{s1}, p_{i1}, p_{s1}) \in SB_1, \exists (t_{i2}, t_{s2}, p_{i2}, p_{s2}) \in SB_2, \\ [t_{i1}, t_{s1}[\cap [t_{i2}, t_{s2}[\neq \emptyset \wedge (p_{i1} \neq p_{i2} \vee p_{s1} \neq p_{s2})$$

Supposons d'abord que les extensions temporelles des suites sont différentes, soit :

$$[t_i(SB_1), t_s(SB_1)[\neq [t_i(SB_2), t_s(SB_2)[$$

Le fait que ces intervalles sont différents nous assure qu'il existe un instant t_0 appartenant à l'un mais n'appartenant pas à l'autre. En effet:

$$[t_i(SB_1), t_s(SB_1)[\neq [t_i(SB_2), t_s(SB_2)[\\ \Rightarrow \exists t_0 \in \mathbf{R}, (t_0 \in [t_i(SB_1), t_s(SB_1)[\wedge t_0 \notin [t_i(SB_2), t_s(SB_2)[) \\ \vee (t_0 \notin [t_i(SB_1), t_s(SB_1)[\wedge t_0 \in [t_i(SB_2), t_s(SB_2)[)$$

Or, par définition d'une SBT, pour chaque instant t_0 de son extension temporelle, il existe nécessairement un couple (t_0, y) appartenant à cette suite. D'où:

$$[t_i(SB_1), t_s(SB_1)[\neq [t_i(SB_2), t_s(SB_2)[\\ \Rightarrow \exists t_0 \in \mathbf{R}, \exists y \in \mathbf{R}, ((t_0, y) \in SB_1 \wedge (t_0, y) \notin SB_2) \\ \vee ((t_0, y) \notin SB_1 \wedge (t_0, y) \in SB_2)$$

Car, si t_0 n'appartient pas à l'extension temporelle de SB_2 (resp. SB_1), alors aucun couple $(t_0,$

y) n'appartient à SB_2 (resp. SB_1).

Or, ceci est absurde car tout point appartenant à SB_1 appartient aussi à SB_2 , et inversement (définition de l'équivalence). Par conséquent, les extensions temporelles de deux suites équivalentes sont égales.

Supposons maintenant qu'il existe deux bandes temporelles $B_1 = (t_{i1}, t_{s1}, p_{i1}, p_{s1})$ et $B_2 = (t_{i2}, t_{s2}, p_{i2}, p_{s2})$ appartenant respectivement à SB_1 et SB_2 telles que

$$[t_{i1}, t_{s1}[\cap [t_{i2}, t_{s2}[\neq \emptyset \wedge (p_{i1} \neq p_{i2} \vee p_{s1} \neq p_{s2})$$

La définition des suites de bandes temporelles équivalentes nous assure que :

$$\begin{aligned} \forall x \in [t_{i1}, t_{s1}[\cap [t_{i2}, t_{s2}[, \forall y \in \mathbf{R}, (x, y) \in B_1 &\Leftrightarrow (x, y) \in B_2 \\ \Leftrightarrow \forall x \in [t_{i1}, t_{s1}[\cap [t_{i2}, t_{s2}[, \forall y \in \mathbf{R}, y \in [p_{i1}(x), p_{s1}(x)] &\Leftrightarrow y \in [p_{i2}(x), p_{s2}(x)] \\ \Leftrightarrow \forall x \in [t_{i1}, t_{s1}[\cap [t_{i2}, t_{s2}[, [p_{i1}(x), p_{s1}(x)] = [p_{i2}(x), p_{s2}(x)] & \\ \Leftrightarrow \forall x \in [t_{i1}, t_{s1}[\cap [t_{i2}, t_{s2}[, p_{i1}(x) = p_{i2}(x) \wedge p_{s1}(x) = p_{s2}(x) & \end{aligned}$$

Comme l'ensemble \mathbf{R} est dense et que les intervalles $[t_{i1}, t_{s1}[$ et $[t_{i2}, t_{s2}[$ sont semi-ouverts, ces intervalles s'intersectent en une infinité d'instant. Donc, les polynômes p_{i1}, p_{i2} et p_{s1}, p_{s2} sont égaux car ils sont égaux en une infinité d'instant. Or, ceci est absurde. Donc, les polynômes des bandes temporelles, dont les extensions temporelles s'intersectent, sont égaux.

⇐

Raisonnons par l'absurde. Supposons que les suites de bandes temporelles SB_1 et B_2 ne sont pas équivalentes, c'est à dire qu'il existe un couple (x_0, y_0) appartenant à SB_1 mais pas à SB_2 , ou inversement. Montrons que si (x_0, y_0) appartient à SB_1 , il appartient aussi à SB_2 .

Par définition de l'appartenance à une suite de bandes temporelles, on a :

$$(x_0, y_0) \in SB_1 \Rightarrow \exists (t_{i1}, t_{s1}, p_{i1}, p_{s1}) \in SB_1, x_0 \in [t_{i1}, t_{s1}[\wedge y_0 \in [p_{i1}(x_0), p_{s1}(x_0)[$$

De plus,

$$\begin{aligned} x_0 \in [t_{i1}, t_{s1}[&\Rightarrow x_0 \in [t_i(SB_1), t_s(SB_1)[\\ &\Rightarrow \exists (t_{i2}, t_{s2}, p_{i2}, p_{s2}) \in SB_2, x_0 \in [t_{i2}, t_{s2}[\text{ (car } [t_i(SB_1), t_s(SB_1)[= [t_i(SB_2), t_s(SB_2)[) \\ &\Rightarrow \exists (t_{i2}, t_{s2}, p_{i2}, p_{s2}) \in SB_2, x_0 \in [t_{i2}, t_{s2}[\cap [t_{i1}, t_{s1}[\\ &\Rightarrow p_{i1} = p_{i2} \text{ et } p_{s1} = p_{s2} \text{ (par hypothèse)} \end{aligned}$$

Donc, $y_0 \in [p_{i2}(x_0), p_{s2}(x_0)[$ et (x_0, y_0) appartient à SB_2 .

□

Les fonctions temporelles représentées par deux suites équivalentes sont maintenant caractérisées.

Propriété 3.1 : “Fonctions temporelles représentées par deux suites de bandes temporelles équivalentes”

Deux suites de bandes temporelles équivalentes représentent les mêmes fonctions temporelles.

D'une manière formelle :

$$SB_1 \sim SB_2 \Leftrightarrow \forall g(t) \in \mathbf{G}, g(t) \text{ Rep } SB_1 \Leftrightarrow g(t) \text{ Rep } SB_2$$

Preuve :

$$\begin{aligned} \Rightarrow \\ g(t) \text{ Rep } SB_1 &\Leftrightarrow \forall t \in [t_i(SB_1), t_s(SB_1)[, (t, g(t)) \in SB_1 \\ &\Leftrightarrow \forall t \in [t_i(SB_2), t_s(SB_2)[, (t, g(t)) \in SB_2 \text{ (car } SB_1 \text{ et } SB_2 \text{ sont équivalentes)} \\ &\Leftrightarrow g(t) \text{ Rep } SB_2 \end{aligned}$$

\Leftarrow
Raisonnons par l'absurde : supposons que deux suites de bandes temporelles SB_1 et SB_2 représentent des fonctions identiques et qu'il existe un point (t_0, y) appartenant à SB_1 mais pas à SB_2 . Il existe nécessairement une fonction $g(t)$ représentée par SB_1 telle que $g(t_0) = y$. Or, cette fonction n'est pas représentée par SB_2 car $(t_0, g(t_0))$ n'appartient pas à SB_2 . Donc, c'est absurde et les suites SB_1 et SB_2 sont équivalentes.

□

Il existe une infinité de suites de bandes temporelles équivalentes. Nous en caractérisons une : celle contenant le nombre minimum de bandes temporelles ; en d'autres termes, celle représentant la plus forte concentration d'information.

Définition 3.4 : "Suite de bandes temporelles minimale"

Une suite de bandes temporelles SB est dite minimale ssi la taille de toutes les suites équivalentes à SB est supérieure ou égale à la taille de SB .

D'une manière formelle :

$$SB \text{ minimale} \Leftrightarrow \forall SB', (SB' \sim SB \Rightarrow \text{card}(SB') \geq \text{card}(SB))$$

La figure 3.3 décrit une suite de bandes temporelles minimale SB équivalente aux suites de bandes temporelles SB_3 et SB_4 de la figure 3.2.

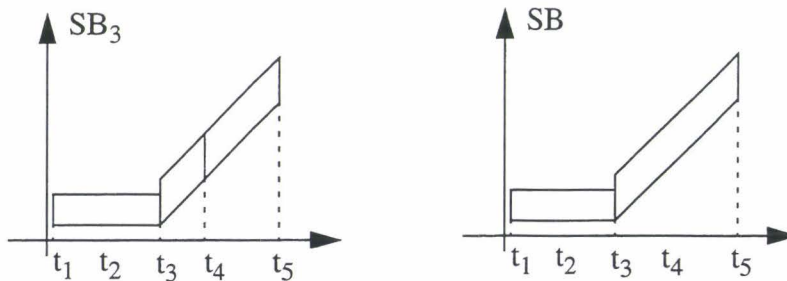


Figure 3.3 : Suite de bandes temporelles minimale

Afin de caractériser plus précisément les suites de bandes temporelles minimales, nous introduisons d'abord la notion de bandes temporelles contiguës :

On dit que deux bandes temporelles B_1, B_2 sont contiguës ssi $t_i(B_1) = t_s(B_2)$ ou $t_i(B_2) = t_s(B_1)$.

Lemme 3.3 :

Une suite de bandes temporelles est minimale ssi les polynômes inférieurs ou supérieurs de deux bandes temporelles contiguës (et quelconques) sont différents.

Chapitre 3: Relations entre les suites de bandes temporelles

D'une manière formelle :

Soit une suite de bandes temporelles $SB = \{(t_1, t_2, p_{i_1}, p_{s_1}), \dots, (t_n, t_{n+1}, p_{i_n}, p_{s_n})\}$.

SB minimale $\Leftrightarrow \forall k \in [1, n-1], p_{i_k} \neq p_{i_{k+1}} \vee p_{s_k} \neq p_{s_{k+1}}$

Preuve :

\Rightarrow

Soit $SB = \{(t_1, t_2, p_{i_1}, p_{s_1}), (t_2, t_3, p_{i_2}, p_{s_2}), \dots, (t_n, t_{n+1}, p_{i_n}, p_{s_n})\}$ une suite de bandes temporelles minimale. Raisonnons par l'absurde et supposons qu'il existe $k \in [1, n-1]$ tel que

$$p_{i_k} = p_{i_{k+1}} \text{ et } p_{s_k} = p_{s_{k+1}}.$$

Il découle directement du lemme 3.2 que la suite de bandes temporelles $\{(t_1, t_2, p_{i_1}, p_{s_1}), \dots, (t_{k-1}, t_k, p_{i_{k-1}}, p_{s_{k-1}}), (t_k, t_{k+2}, p_{i_k}, p_{s_k}), (t_{k+2}, t_{k+3}, p_{i_{k+2}}, p_{s_{k+2}}), \dots, (t_n, t_{n+1}, p_{i_n}, p_{s_n})\}$ est équivalente à SB . De plus, cette suite contient $n-1$ bandes temporelles, donc la suite SB n'est pas minimale. Comme c'est absurde, on a :

SB minimale $\Rightarrow \forall k \in [1, n-1], p_{i_k} \neq p_{i_{k+1}} \vee p_{s_k} \neq p_{s_{k+1}}$

\Leftarrow

Soit $SB = \{(t_1, t_2, p_{i_1}, p_{s_1}), (t_2, t_3, p_{i_2}, p_{s_2}), \dots, (t_n, t_{n+1}, p_{i_n}, p_{s_n})\}$ une suite de bandes temporelles telle que, pour tout k appartenant à $[1, n-1]$, $p_{i_k} \neq p_{i_{k+1}}$ ou $p_{s_k} \neq p_{s_{k+1}}$.

Raisonnons par l'absurde et supposons que SB n'est pas minimale. D'après la définition d'une suite de bandes temporelles minimale, il existe une suite de bandes temporelles contenant au plus $n-1$ bandes temporelles et équivalente à SB . Soit $SB' = \{(t'_1, t'_2, p_{i'_1}, p_{s'_1}), (t'_2, t'_3, p_{i'_2}, p_{s'_2}), \dots, (t'_p, t'_{p+1}, p_{i'_p}, p_{s'_p})\}$ ($p < n$) cette suite.

Comme les suites SB et SB' sont équivalentes, leurs extensions temporelles sont égales (lemme 3.2). Soit : $t'_1 = t_1$ et $t'_{p+1} = t_{n+1}$

Montrons qu'il existe une bande temporelle de SB' dont l'extension temporelle intersecte les extensions temporelles de deux bandes temporelles contiguës de SB . Formellement :

$$\exists j \in [1, p], \exists k \in [1, n-1], [t'_j, t'_{j+1}] \cap [t_k, t_{k+1}] \neq \emptyset \wedge [t'_j, t'_{j+1}] \cap [t_{k+1}, t_{k+2}] \neq \emptyset$$

Pour montrer cette relation, nous raisonnons par l'absurde et supposons qu'elle est fausse.

Ceci implique que, comme $t_1 = t'_1, t'_2 \leq t_2$ (sinon l'extension temporelle de $SB'[1]$ intersecte les extensions temporelles de $SB[1]$ et $SB[2]$)

De même, $t'_2 \leq t_2 \Rightarrow t'_3 \leq t_3 \Rightarrow \dots \Rightarrow t'_{p+1} \leq t_{p+1}$.

Or, par définition des suites de bandes temporelles, on a $t_{p+1} < t_{n+1}$ (car $p < n$).

Ceci est donc absurde car on a $t'_{p+1} \leq t_{p+1} < t_{n+1}$ et $t'_{p+1} = t_{n+1}$. Par conséquent,

$$\exists j \in [1, p], \exists k \in [1, n-1], [t'_j, t'_{j+1}] \cap [t_k, t_{k+1}] \neq \emptyset \wedge [t'_j, t'_{j+1}] \cap [t_{k+1}, t_{k+2}] \neq \emptyset$$

De plus, il découle directement du lemme 3.2 que : $p'_{i'_j} = p_{i_k} = p_{i_{k+1}}$ et $p'_{s'_j} = p_{s_k} = p_{s_{k+1}}$.

Or, ceci est absurde car nous avons supposé que $p_{i_k} \neq p_{i_{k+1}}$ ou $p_{s_k} \neq p_{s_{k+1}}$. Donc, la suite SB est minimale.

□

Il découle de ce lemme que vérifier si une suite de bandes temporelles est minimale consiste à comparer les polynômes des bandes temporelles contiguës de cette suite. Par conséquent, toute suite de bandes temporelles de taille 1 est nécessairement minimale (pas de bandes temporelles contiguës). Nous caractérisons maintenant l'existence des suites de bandes temporelles minimales.

Lemme 3.4 :

Pour toute suite de bandes temporelles SB , il existe une et une seule suite de bandes temporelles minimale et équivalente à SB .

D'une manière formelle :

$\forall SB, \exists ! SB', SB' \sim SB \wedge SB \text{ minimale}$

Preuve :

Preuve de l'existence d'une suite minimale

Soit SB_n une suite comprenant n bandes temporelles. Raisonnons par l'absurde : supposons qu'il n'existe pas de suite équivalente à SB_n et minimale. Comme SB_n n'est pas minimale, il existe une suite SB_{n-1} équivalente à SB_n et dont la taille est strictement inférieure à celle de SB_n (i.e. SB_{n-1} contient au plus $n-1$ bandes temporelles). Comme la suite SB_{n-1} est équivalente à SB_n , cette suite n'est pas minimale (hypothèse émise).

Répétons ce raisonnement avec SB_{n-1} , SB_{n-2} , ... jusqu'à obtenir la suite SB_1 qui contient 1 seule bande temporelle. Or, SB_1 est minimale (car de taille 1) et est équivalente à SB_n . L'hypothèse émise est donc absurde. Par conséquent, pour toute suite de bandes temporelles SB , il existe une suite de bandes temporelles SB' minimale et équivalente à SB .

Unicité de la suite minimale

Supposons qu'il existe deux suites de bandes temporelles, SB et SB' , différentes, équivalentes et minimales. Comme les suites SB et SB' sont minimales, elles contiennent le même nombre de bandes temporelles.

Un argument semblable à celui utilisé dans la preuve du lemme 3.3 permet de montrer que si aucune bande $SB[k]$ n'avait une extension temporelle intersectant les extensions temporelles de deux bandes temporelles contiguës de SB' , alors chaque $SB[k]$ aurait la même extension temporelle que $SB'[k]$. Mais, l'équivalence de SB et de SB' implique alors que $SB[k] = SB'[k]$ pour tout k , soit $SB = SB'$. Or ceci est impossible. Par conséquent, il existe nécessairement une bande temporelle de SB , $SB[k]$, dont l'extension temporelle intersecte les extensions temporelles de deux bandes temporelles contiguës de SB' , $SB'[j]$ et $SB'[j+1]$.

Il découle du lemme 3.2 que, comme les suites SB et SB' sont équivalentes, les polynômes de $SB'[j]$, $SB'[j+1]$ et $SB[k]$ sont égaux.

Or, ceci est absurde car, d'après le lemme 3.3, les polynômes de deux bandes temporelles contiguës d'une suite minimale sont différents.

Par conséquent, pour toute suite de bandes temporelles SB , il existe une et une seule suite de bandes temporelles SB' minimale et équivalente à SB .

□

En conclusion, les suites de bandes temporelles équivalentes forment une classe d'équivalence. Un élément générateur est la suite de bandes temporelles minimale, dont la particularité est d'être unique. Générer la suite de bandes temporelles minimale équivalente à SB consiste donc à comparer les polynômes des bandes contiguës SB[k] et SB[k+1] de SB, pour k variant de 1 à card(SB) - 1. Lorsque ces polynômes sont égaux, il faut alors former la bande temporelle dont l'extension temporelle est l'union des extensions temporelles de SB[k] et SB[k+1] (i.e. $[t_i(SB[k]), t_s(SB[k+1])]$) et dont les polynômes sont ceux de SB[k].

3.3 Synchronisation de deux suites de bandes temporelles

Deux suites de bandes temporelles équivalentes sont égales à un découpage temporel près. Nous définissons maintenant deux suites de bandes temporelles différentes, mais ayant le même découpage temporel.

Définition 3.5 : “Couple de suites de bandes temporelles synchrones”

Un couple de suites de bandes temporelles (SB_1, SB_2) est synchrone ssi

- le nombre de bandes temporelles de ces suites est identique,
- les extensions temporelles des bandes temporelles de SB_1 et SB_2 sont deux à deux égales.

D'une manière formelle :

Soient SB_1 et SB_2 deux suites de bandes temporelles.

(SB_1, SB_2) synchrone

$$\Leftrightarrow \text{card}(SB_1) = \text{card}(SB_2) = n \wedge \forall k \in [1, n], [t_i(SB_1[k]), t_s(SB_1[k])] = [t_i(SB_2[k]), t_s(SB_2[k])]$$

Par la suite, on dira indifféremment que les suites de bandes temporelles SB_1, SB_2 sont synchrones ou que le couple (SB_1, SB_2) est synchrone.

La figure 3.4 décrit un couple de suites de bandes temporelles synchrones.

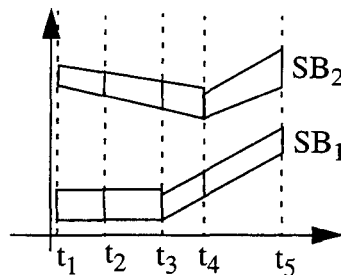


Figure 3.4 : Suites de bandes temporelles synchrones

Notons que deux suites de bandes temporelles synchrones (SB_1, SB_2) ont la même extension temporelle. En effet, les extensions temporelles de SB_1 et SB_2 sont respectivement $[t_i(SB_1[1]), t_s(SB_1[n])]$ et $[t_i(SB_2[1]), t_s(SB_2[n])]$ et $t_i(SB_1[1]) = t_i(SB_2[1])$ et $t_s(SB_1[n]) = t_s(SB_2[n])$ car les suites sont synchrones.

Nous caractérisons maintenant les suites de bandes temporelles synchrones représentant la meilleure concentration des informations.

Définition 3.6 : “Couple de suites de bandes temporelles synchrones minimal”

Un couple de suites de bandes temporelles synchrones (SB_1, SB_2) est minimal ssi, pour tout couple de suites de bandes temporelles synchrones (SB'_1, SB'_2) tel que SB'_1 (resp. SB'_2) est équivalente à SB_1 (resp. SB_2), le nombre de bandes temporelles de SB'_1 est supérieur ou égal au nombre de bandes temporelles de SB_1 .

D'une manière formelle :

Soient (SB_1, SB_2) un couple de suites de bandes temporelles synchrones.

(SB_1, SB_2) synchrones minimale

$$\Leftrightarrow (\forall SB'_1, \forall SB'_2, (SB'_1, SB'_2) \text{ synchrones} \wedge SB'_1 \sim SB_1 \wedge SB'_2 \sim SB_2 \Rightarrow \text{card}(SB_1) \leq \text{card}(SB'_1))$$

La figure 3.5 illustre un couple de suites de bandes temporelles synchrones minimale dont les suites sont équivalentes aux suites de la figure 3.4.

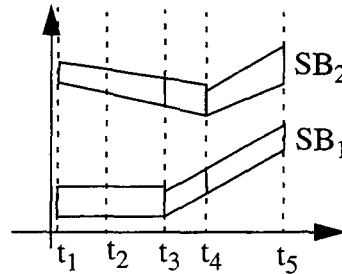


Figure 3.5 : Couple de suites de bandes temporelles synchrones minimale

Nous caractérisons maintenant l'existence et le nombre maximum de bandes temporelles de ces suites.

Lemme 3.5 :

Pour toutes suites de bandes temporelles quelconques SB_1 et SB_2 , il existe un et un seul couple de suites de bandes temporelles synchrones minimal (SB'_1, SB'_2) tel que SB'_1 (resp. SB'_2) est équivalente à SB_1 (resp. SB_2).

D'une manière formelle :

$$\forall SB_1, \forall SB_2, \exists! (SB'_1, SB'_2) \text{ tel que } (SB'_1, SB'_2) \text{ synchrones minimal.}$$

Preuve :

Soient deux suites de bandes temporelles SB_1 et SB_2 de taille n et p , dont les extensions temporelles sont identiques.

Montrons d'abord qu'il existe nécessairement un couple de SBT (SB'_1, SB'_2) synchrones et tel que $SB'_1 \sim SB_1$ et $SB'_2 \sim SB_2$.

Soit TI l'ensemble ordonné des instants de début des bandes temporelles de SB_1 et SB_2 tel que TI ne contient pas d'élément redondant. Cet ensemble contient au plus $(n + p)$ éléments.

Construisons une suite de bandes temporelles SB'_1 à partir de TI et de SB_1 . Les bandes temporelles $B'_1 = (t'_i, t'_s, p'_i, p'_s)$ de SB'_1 doivent respecter les critères suivants :

$$t'_i \in TI \text{ et } t'_s \in TI \cup \{t_s(SB_1)\} \text{ et } \forall t \in TI, t \leq t'_i \text{ ou } t \geq t'_s$$

$$\begin{aligned} \exists(t_i, t_s, p_i, p_s) \in SB_1, [t'_i, t'_s] \subseteq [t_i, t_s], \\ p'_i = p_i, p'_s = p_s \end{aligned}$$

D'après le lemme 3.2, SB_1' est équivalente à SB_1 .

Nous construisons la suite de bandes temporelles SB_2' à partir de SB_2 et de TI de manière identique.

Les suites SB_1' et SB_2' sont synchrones car elles ont le même découpage temporel.

Montrons qu'il existe nécessairement un couple de suites de bandes temporelles synchrone minimal et dont les SBT sont respectivement équivalentes à SB_1' et SB_2' . Raisonnons par l'absurde et supposons qu'il n'existe pas de couples de SBT synchrone minimaux. Par hypothèse, (SB_1', SB_2') n'est pas synchrone minimal. Il existe donc un couple (SB_1'', SB_2'') synchrone tel que $SB_1'' \sim SB_1'$, $SB_2'' \sim SB_2'$ et la taille de ces suites est inférieure à celle de SB_1' . En répétant ce raisonnement, on aboutit à l'existence d'un couple de SBT dont la taille est 0 et qui sont respectivement équivalentes à SB_1' et SB_2' . Or, ceci est absurde (Ce raisonnement est similaire à celui utilisé dans la preuve de l'existence d'une suite de bandes temporelles minimale).

Nous montrons l'unicité de cette suite en raisonnant par l'absurde ; supposons qu'il existe deux couples de suites de bandes temporelles synchrones minimaux (SB_1, SB_2) et (SB'_1, SB'_2) tels que SB_1 (resp. SB_2) soit équivalente à et différente de SB'_1 (resp. SB'_2). Comme ces couples sont synchrones minimaux, les suites SB_1, SB'_1, SB_2 et SB'_2 sont de taille identique.

De la même manière que dans la preuve du lemme 3.3, il est possible de montrer qu'il existe une bande temporelle $SB_1[j]$ de SB_1 dont l'extension temporelle intersecte les extensions temporelles de deux bandes temporelles contiguës $SB'_1[k]$ et $SB'_1[k+1]$ de SB'_1 .

Comme les couples de suites (SB_1, SB_2) et (SB'_1, SB'_2) sont synchrones, les extensions temporelles de $SB_2[j]$, $SB'_2[k]$ et $SB'_2[k+1]$ sont respectivement égales aux extensions temporelles de $SB_1[j]$, $SB'_1[k]$ et $SB'_1[k+1]$.

Comme les suites SB_1 et SB'_1 (resp. SB_2 et SB'_2) sont équivalentes, les polynômes des bandes temporelles $SB_1[j]$, $SB'_1[k]$, $SB'_1[k+1]$ (resp. $SB_2[j]$, $SB'_2[k]$, $SB'_2[k+1]$) sont égaux (lemme 3.2). Soient p_{i1} et p_{s1} (resp. p_{i2} et p_{s2}) les polynômes inférieur et supérieur des bandes temporelles $SB_1[j]$, $SB'_1[k]$, $SB'_1[k+1]$ (resp. $SB_2[j]$, $SB'_2[k]$, $SB'_2[k+1]$).

D'après le lemme 3.2, les suites de bandes temporelles

$\{SB'_1[1], \dots, SB'_1[k-1], (t_i(SB'_1[k]), t_s(SB'_1[k+1]), p_{i1}, p_{s1}), SB'_1[k+2], \dots, SB'_1[n]\}$ et $\{SB'_2[1], \dots, SB'_2[k-1], (t_i(SB'_1[k]), t_s(SB'_1[k+1]), p_{i2}, p_{s2}), SB'_2[k+2], \dots, SB'_2[n]\}$ sont respectivement équivalentes aux suites SB_1 et SB_2 .

De plus, ces suites sont synchrones (car (SB'_1, SB'_2) est synchrone). Par conséquent, il existe un couple de suites synchrone, équivalent à (SB_1, SB_2) et contenant moins de bandes temporelles que les suites SB_1 et SB_2 . Ce qui est absurde, donc il n'existe qu'un seul couple de suites de bandes temporelles synchrone minimal.

□

Nous caractérisons la taille maximale de suites de bandes des couples synchrones minimaux.

Lemme 3.6 :

Soit (SB_1, SB_2) un couple de suites de bandes temporelles synchrone minimal.

Le nombre de bandes temporelles de SB_1 est strictement inférieur à la somme des nombres de bandes temporelles de toutes suites SB_1' et SB_2' respectivement équivalentes à SB_1 et SB_2 .

D'une manière formelle :

(SB_1, SB_2) synchrone minimale

$$\Rightarrow (\forall SB_1', \forall SB_2', SB_1' \sim SB_1 \wedge SB_2' \sim SB_2 \Rightarrow \text{card}(SB_1) < \text{card}(SB_1') + \text{card}(SB_2'))$$

Preuve :

Soient deux suites de bandes temporelles SB_1 et SB_2 de taille n et p , dont les extensions temporelles sont identiques.

Soit TI l'ensemble ordonné des instants de début des bandes temporelles de SB_1 et SB_2 tel que TI ne contient pas d'élément redondant. Cet ensemble contient au plus $(n + p - 1)$ éléments.

De la même manière que dans le lemme 3.5, il est possible de construire deux SBT SB_1' et SB_2' dont les instants de coupure sont les éléments de TI telles que ces SBT soient respectivement équivalentes à SB_1 et à SB_2 et (SB_1', SB_2') soit synchrone.

Les suites SB_1' et SB_2' sont synchrones et contiennent un nombre de bandes temporelles égal au nombre d'instants de TI . Or, le nombre d'instants de TI est inférieur à la somme des nombres d'instants de début des extensions temporelles des bandes temporelles de SB_1 et des bandes temporelles de SB_2 .

De plus, le nombre de bandes temporelles d'une suite est égal au nombre d'instants de début des extensions temporelles des bandes temporelles. (Chaque bande temporelle ne contenant qu'un et un seul instant de début). Donc, le nombre d'instants de TI est inférieur ou égal à la somme des nombres de bandes temporelles de SB_1 et de bandes temporelles de SB_2 .

Il en découle : $\text{card}(SB_1') = \text{card}(SB_2') = \text{card}(TI) < \text{card}(SB_1) + \text{card}(SB_2)$.

Or, le nombre de bandes temporelles du couple de suites de bandes temporelles synchrone minimal est inférieur au nombre de bandes temporelles de tout couple de suites synchrones.

Donc, si (SB_1'', SB_2'') est le couple de suites de bandes temporelles synchrone minimal tel que SB_1'' (resp. SB_2'') est équivalente à SB_1 (resp. SB_2), alors le nombre de bandes temporelles de SB_1'' est inférieur ou égal à la somme des nombres de bandes temporelles de SB_1 et de SB_2 .

□

Nous pouvons maintenant définir l'opération permettant d'obtenir une suite de bandes temporelles synchrone minimale à partir d'une suite de bandes temporelles quelconque.

Définition 3.7 : “Synchronisation de deux suites de bandes temporelles”

La synchronisation est l'opération qui, à tout couple de suites de bandes temporelles (SB_1, SB_2) , associe le couple de bandes temporelles synchrone minimal (SB_1', SB_2') tel que SB_1' (resp. SB_2') est équivalente à SB_1 (resp. SB_2).

Nous définissons maintenant d'autres relations entre les suites de bandes temporelles équivalentes.

3.4 Inclusion de suites de bandes temporelles

Nous définissons d'abord l'inclusion pour deux bandes temporelles.

Définition 3.8 : "Inclusion de deux bandes temporelles"

Soient B_1 et B_2 deux bandes temporelles d'extensions temporelles identiques.

La bande temporelle B_1 est incluse dans B_2 , et on notera $B_1 \subseteq B_2$, ssi tout couple (x, y) appartenant à B_1 appartient aussi à B_2 .

D'une manière formelle :

$$B_1 \subseteq B_2 \Leftrightarrow (\forall (x, y) \in \mathbf{R}^2, (x, y) \in B_1 \Rightarrow (x, y) \in B_2)$$

La figure 3.6 illustre deux bandes temporelles incluses l'une dans l'autre :

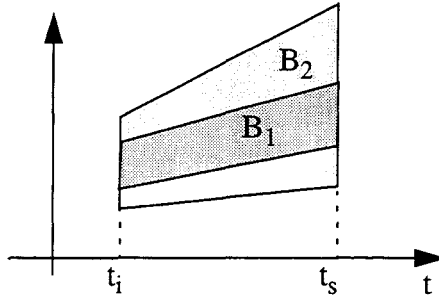


Figure 3.6 : Bandes temporelles incluses

Il découle de la définition de l'égalité de deux bandes temporelles (définition 3.1) que :

$$B_1 \subseteq B_2 \wedge B_2 \subseteq B_1 \Leftrightarrow B_1 = B_2$$

La relation suivante permet de caractériser deux bandes temporelles incluses l'une dans l'autre :

$$(t_i, t_s, p_{i1}, p_{s1}) \subseteq (t_i, t_s, p_{i2}, p_{s2}) \Leftrightarrow \forall t \in [t_i, t_s[, p_{i2}(t) \leq p_{i1}(t) \leq p_{s1}(t) \leq p_{s2}(t)$$

(La preuve est identique à celle du lemme 3.1)

Nous étendons maintenant la définition de l'inclusion aux suites de bandes temporelles synchrones.

Définition 3.9 : "Inclusion de deux suites de bandes temporelles synchrones"

Soient SB_1 et SB_2 deux suites de bandes temporelles synchrones.

La suite de bandes temporelles SB_1 est incluse dans SB_2 ssi toutes les bandes temporelles de SB_1 sont incluses dans celles de SB_2 .

D'une manière formelle :

$$SB_1 \subseteq SB_2 \Leftrightarrow \forall k \in [1, n], SB_1[k] \subseteq SB_2[k]$$

L'inclusion de deux suites de bandes temporelles peut aussi s'exprimer en terme de relations entre les fonctions temporelles représentées.

Propriété 3.2 : "Fonctions temporelles représentées par deux suites de bandes temporelles incluses l'une dans l'autre"

Soient SB_1 et SB_2 deux suites de bandes temporelles.

Si SB_1 est incluse dans SB_2 alors toute fonction temporelle représentée par SB_1 est aussi représentée par SB_2 .

D'une manière formelle :

$$SB_1 \subseteq SB_2 \Leftrightarrow \forall g(t) \in \mathbf{G}, g(t) \text{ Rep } SB_1 \Rightarrow g(t) \text{ Rep } SB_2$$

(La preuve est similaire à celle de la propriété équivalente de l'égalité, i.e. propriété 3.1)

3.5 Intersection de deux suites de bandes temporelles

Dans ce paragraphe, nous proposons de définir l'opération d'intersection qui, à deux SBT quelconques SB_1 et SB_2 , associe une SBT représentant les fonctions représentées à la fois par SB_1 et par SB_2 . Cette SBT n'existe pas toujours. C'est pourquoi nous définissons dans un premier temps les SBT qui ont une intersection non vide.

Définition 3.10 : "Intersection vide de deux bandes temporelles"

Soient B_1 et B_2 deux bandes temporelles d'extension temporelle $[t_i, t_s[$.

L'intersection des bandes temporelles B_1, B_2 est vide et on notera $B_1 \cap B_2 = \emptyset$ ssi il existe un instant t_0 de $[t_i, t_s[$ tel que tout couple (t_0, y) n'appartienne pas à la fois à B_1 et à B_2 .

D'une manière formelle :

$$B_1 \cap B_2 = \emptyset \Leftrightarrow \exists t_0 \in [t_i(B_1), t_s(B_1)[, \forall y \in \mathbf{R}, (t_0, y) \notin B_1 \vee (t_0, y) \notin B_2$$

Graphiquement, l'intersection de deux bandes temporelles est vide lorsqu'elles ne se chevauchent pas en un instant de leurs extensions temporelles (cf. figure 3.7).

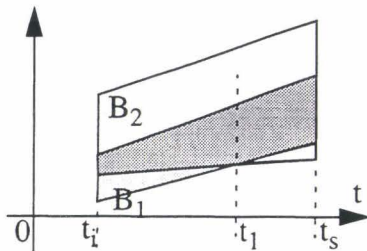


Figure 3.7a : Intersection non vide

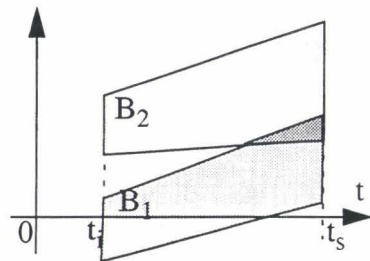


Figure 3.7b : Intersection vide

Nous proposons maintenant une relation existant entre les polynômes des bandes temporelles dont l'intersection est vide.

Lemme 3.7 : “Polynômes de deux bandes temporelles dont l’intersection est vide”

L’intersection de deux bandes temporelles B_1, B_2 est vide ssi le polynôme supérieur de B_1 (resp. de B_2) est inférieur au polynôme inférieur de B_1 (resp. de B_2) en un instant.

D’une manière formelle :

$$(t_i, t_s, p_{i1}, p_{s1}) \cap (t_i, t_s, p_{i2}, p_{s2}) = \emptyset \Leftrightarrow \exists t_0 \in [t_i, t_s[, p_{s1}(t_0) < p_{i2}(t_0) \vee p_{s2}(t_0) < p_{i1}(t_0)$$

Preuve :

$$\begin{aligned} & (t_i, t_s, p_{i1}, p_{s1}) \cap (t_i, t_s, p_{i2}, p_{s2}) = \emptyset \\ & \Leftrightarrow \exists t_0 \in [t_i, t_s[, \forall y \in \mathbf{R}, (t_0, y) \notin (t_i, t_s, p_{i1}, p_{s1}) \vee (t_0, y) \notin (t_i, t_s, p_{i2}, p_{s2}) \\ & \Leftrightarrow \exists t_0 \in [t_i, t_s[, \forall y \in \mathbf{R}, y \notin [p_{i1}(t_0), p_{s1}(t_0)] \vee y \notin [p_{i2}(t_0), p_{s2}(t_0)] \\ & (\text{car } (t_0, y) \notin (t_i, t_s, p_{i1}, p_{s1}) \Leftrightarrow t_0 \notin [t_i, t_s[\vee y \notin [p_{i1}(t_0), p_{s1}(t_0)] \text{ et } t_0 \in [t_i, t_s[) \\ & \Leftrightarrow \exists t_0 \in [t_i, t_s[, \forall y \in \mathbf{R}, y \notin [p_{i1}(t_0), p_{s1}(t_0)] \cap [p_{i2}(t_0), p_{s2}(t_0)] \\ & \Leftrightarrow \exists t_0 \in [t_i, t_s[, [p_{i1}(t_0), p_{s1}(t_0)] \cap [p_{i2}(t_0), p_{s2}(t_0)] = \emptyset \\ & \Leftrightarrow \exists t_0 \in [t_i, t_s[, [\max(p_{i1}(t_0), p_{i2}(t_0)), \min(p_{s1}(t_0), p_{s2}(t_0))] = \emptyset \\ & \Leftrightarrow \exists t_0 \in [t_i, t_s[, p_{s1}(t_0) < p_{i2}(t_0) \vee p_{s2}(t_0) < p_{i1}(t_0) \end{aligned}$$

□

Nous proposons maintenant un algorithme permettant de vérifier si deux bandes temporelles s’intersectent. Cet algorithme est directement issu du lemme précédent (lemme 3.7).

Vérifier si l’intersection de $(t_i, t_s, p_{i1}, p_{s1})$ et $(t_i, t_s, p_{i2}, p_{s2})$ est non vide consiste à :

- Vérifier si on n’a pas $p_{s1}(t_i) < p_{i2}(t_i)$ ou $p_{s1}(t_s) < p_{i2}(t_s)$ ou $p_{s2}(t_i) < p_{i1}(t_i)$ ou $p_{s2}(t_s) < p_{i1}(t_s)$
- Calculer les solutions de l’équation $p_{i1}(t) = p_{s2}(t)$.
Pour toute solution t_x appartenant à $[t_i, t_s[$, vérifier si on n’a pas $p_{s1}(t_x) < p_{i2}(t_x)$ ou $p_{s1}(t_x) < p_{i2}(t_x)$.
- Répéter cette opération pour les solutions de $p_{i2}(t) = p_{s1}(t)$.

La relation d’intersection est maintenant étendue aux suites de bandes temporelles.

Définition 3.11 : “Intersection vide de deux suites de bandes temporelles”

L’intersection de deux suites de bandes temporelles synchrones SB_1, SB_2 est vide et on notera

$SB_1 \cap SB_2 = \emptyset$ ssi il existe deux bandes temporelles de même rang qui ne s’intersectent pas.

D’une manière formelle :

$$SB_1 \cap SB_2 = \emptyset \Leftrightarrow \exists k \in [1, \text{card}(B_1)] \text{ tel que } B_1[k] \cap B_2[k] = \emptyset$$

Nous introduisons maintenant l’opération d’intersection.

Définition 3.12 : “Intersection de deux bandes temporelles”

Soient deux bandes temporelles B_1, B_2 de même extension temporelle et dont l’intersection n’est pas vide.

L’intersection de deux bandes temporelles B_1, B_2 , noté $B_1 \cap B_2$, est l’ensemble des couples de réels (x, y) appartenant à la fois à B_1 et à B_2 .

D'une manière formelle :

$$B_1 \cap B_2 = \{(x, y) \mid (x, y) \in B_1 \wedge (x, y) \in B_2\}$$

L'hypothèse que l'intersection des bandes temporelles n'est pas vide est introduite pour éviter une ambiguïté au niveau des notations : l'intersection de deux bandes temporelles B_1, B_2 dont l'intersection est vide (par définition) n'est pas nécessairement égal à l'ensemble vide (définition de l'intersection). C'est le cas par exemple de l'intersection des bandes temporelles illustrées figure 3.7b.

Nous nous intéressons maintenant au calcul de l'intersection de deux bandes temporelles. L'opération d'intersection des bandes temporelles (définition 3.12) n'est pas interne à l'ensemble des bandes temporelles. Par exemple, l'intersection des bandes temporelles illustrées figure 3.7a n'est pas une bande temporelle. Dans la suite de ce paragraphe, nous allons montrer qu'un découpage adéquat de l'extension temporelle des bandes temporelles permet de rendre l'opération d'intersection interne dans l'ensemble des suites de bandes temporelles.

Nous proposons dans un premier temps de caractériser les bandes temporelles dont l'intersection est une bande temporelle.

Définition 3.13 : “Bandes temporelles compatibles pour l'intersection”

Deux bandes temporelles B_1, B_2 sont compatibles pour l'intersection ssi le polynôme inférieur de B_1 est représenté par B_2 et le polynôme supérieur de B_2 est représenté par B_1 ou inversement.

D'une manière formelle :

$$(t_i, t_s, p_{i1}, p_{s1}) \text{ et } (t_i, t_s, p_{i2}, p_{s2}) \text{ sont compatibles pour l'intersection ssi } (p_{i1}(t) \text{ Rep } B_2 \wedge p_{s2}(t) \text{ Rep } B_1) \vee (p_{s1}(t) \text{ Rep } B_2 \wedge p_{i2}(t) \text{ Rep } B_1)$$

Graphiquement, deux bandes temporelles sont compatibles pour l'intersection lorsqu'elles se chevauchent toujours et que les polynômes des bandes temporelles ne se recoupent pas (cf. figure 3.8).

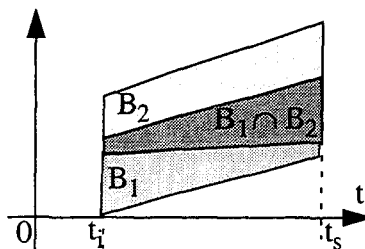


Figure 3.8 : Bandes temporelles compatibles pour l'intersection

Afin de faciliter l'écriture, les opérations inf et sup sont introduites.

Définition 3.14 : “Opérations $inf(T ; p_1, p_2)$ et $sup(T ; p_1, p_2)$ ”

L'opération *inf* (resp. *sup*) est l'opération qui associe à un intervalle temporel $[t_1, t_2]$ et à deux polynômes p_1, p_2 dont l'un est toujours inférieur (au sens large) à l'autre, le polynôme qui est toujours inférieur (resp. supérieur) à l'autre sur l'intervalle $[t_1, t_2]$.

D'une manière formelle :

Soient deux polynômes p_1, p_2 ne s'intersectant pas sur l'intervalle temporel $[t_1, t_2]$.

$$\begin{aligned} \inf([t_1, t_2] ; p_1, p_2) &= p_1 \text{ si pour } t_0 \in [t_1, t_2], p_1(t_0) \leq p_2(t_0) \\ &= p_2 \text{ si pour } t_0 \in [t_1, t_2], p_2(t_0) \leq p_1(t_0) \end{aligned}$$

$$\begin{aligned} \sup([t_1, t_2] ; p_1, p_2) &= p_1 \text{ si pour } t_0 \in [t_1, t_2], p_1(t_0) \geq p_2(t_0) \\ &= p_2 \text{ si pour } t_0 \in [t_1, t_2], p_2(t_0) \geq p_1(t_0) \end{aligned}$$

La signification de la compatibilité est introduite dans le paragraphe suivant.

Lemme 3.8 : “Bandes temporelles dont l'intersection est une bande temporelle”

L'intersection de deux bandes temporelles $(t_i, t_s, p_{i1}, p_{s1}), (t_i, t_s, p_{i2}, p_{s2})$ compatibles pour l'intersection est la bande temporelle $(t_i, t_s, \sup([t_i, t_s] ; p_{i1}, p_{i2}), \inf([t_i, t_s] ; p_{s1}, p_{s2}))$.

Preuve :

Par définition de la propriété de représentation, un polynôme représenté par une bande temporelle B est toujours compris entre les polynômes de B.

Il découle directement de cette remarque que si deux bandes temporelles B_1 et B_2 sont compatibles pour l'intersection alors le polynôme inférieur de B_1 est toujours inférieur à celui de B_2 ou inversement. De même, pour les polynômes supérieurs.

Par définition de l'intersection, on a :

$$\begin{aligned} (t_i, t_s, p_{i1}, p_{s1}) \cap (t_i, t_s, p_{i2}, p_{s2}) &= \{(x, y) \mid (x, y) \in (t_i, t_s, p_{i1}, p_{s1}) \wedge (x, y) \in (t_i, t_s, p_{i2}, p_{s2})\} \\ &= \{(x, y) \mid x \in [t_i, t_s[\wedge y \in [p_{i1}(x), p_{s1}(x)] \wedge y \in [p_{i2}(x), p_{s2}(x)]\} \\ &= \{(x, y) \mid x \in [t_i, t_s[\wedge y \in [p_{i1}(x), p_{s1}(x)] \cap [p_{i2}(x), p_{s2}(x)]\} \\ &= \{(x, y) \mid x \in [t_i, t_s[\wedge y \in [\max(p_{i1}(x), p_{i2}(x)), \min(p_{s1}(x), p_{s2}(x))]\} \end{aligned}$$

Comme p_{i1} est toujours inférieur à p_{i2} ou inversement, $\max(p_{i1}(x), p_{i2}(x))$ est égal à la valeur de $\sup([t_i, t_s] ; p_{i1}, p_{i2})$ en x et ce pour tout x de $[t_i, t_s[$. De même pour $\min(p_{s1}(x), p_{s2}(x))$.

D'où

$$(t_i, t_s, p_{i1}, p_{s1}) \cap (t_i, t_s, p_{i2}, p_{s2}) = (t_i, t_s, \sup([t_i, t_s] ; p_{i1}, p_{i2}), \inf([t_i, t_s] ; p_{s1}, p_{s2}))$$

□

Nous étendons maintenant la définition de la compatibilité aux suites de bandes temporelles.

Définition 3.15 : “suites de bandes temporelles compatibles pour l'intersection”

Deux suites de bandes temporelles synchrones sont compatibles pour l'intersection ssi les bandes temporelles de ces suites sont deux à deux compatibles pour l'intersection.

D'une manière formelle :

Soient SB_1 et SB_2 deux suites de bandes temporelles synchrones.

SB_1 et SB_2 sont compatibles pour l'intersection ssi

$\forall k \in [1, \text{card}(SB_1)], SB_1[k]$ et $SB_2[k]$ sont compatibles pour l'intersection.

Nous étudions maintenant l'existence de telles suites.

Lemme 3.9 : “Existence de suites de bandes temporelles compatibles pour l’intersection”

Soient $B_1 = (t_i, t_s, p_{i1}, p_{s1})$ et $B_2 = (t_i, t_s, p_{i2}, p_{s2})$ deux bandes temporelles et $\{t_1, \dots, t_n\}$ l’ensemble ordonné des solutions dans $[t_i, t_s[$ des équations $p_{i1}(t) = p_{i2}(t)$ (si $p_{i1}(t) \neq p_{i2}(t)$) et $p_{s1}(t) = p_{s2}(t)$ (si $p_{s1}(t) \neq p_{s2}(t)$).

Les suites de bandes temporelles $SB_1 = \{(t_i, t_1, p_{i1}, p_{s1}), (t_1, t_2, p_{i1}, p_{s1}), \dots, (t_n, t_s, p_{i1}, p_{s1})\}$ et $SB_2 = \{(t_i, t_1, p_{i2}, p_{s2}), (t_1, t_2, p_{i2}, p_{s2}), \dots, (t_n, t_s, p_{i2}, p_{s2})\}$ sont

- synchrones,
- respectivement équivalentes à B_1 et B_2 ,
- compatibles pour l’intersection.

La preuve est immédiate.

Il découle directement des lemmes 3.8 et 3.9 que

$$(t_i, t_s, p_{i1}, p_{s1}) \cap (t_i, t_s, p_{i2}, p_{s2}) \\ = \{(t_i, t_1, \sup([t_i, t_1]; p_{i1}, p_{i2}), \inf([t_i, t_1]; p_{s1}, p_{s2})), (t_1, t_2, \sup([t_1, t_2]; p_{i1}, p_{i2}), \inf([t_1, t_2]; p_{s1}, p_{s2})), \\ \dots, (t_n, t_s, \sup([t_i, t_1]; p_{i1}, p_{i2}), \inf([t_n, t_s]; p_{s1}, p_{s2}))\}$$

où $\{t_1, \dots, t_n\}$ l’ensemble ordonné des solutions dans $[t_i, t_s[$ des équations $p_{i1}(t) = p_{i2}(t)$ (si $p_{i1}(t) \neq p_{i2}(t)$) et $p_{s1}(t) = p_{s2}(t)$ (si $p_{s1}(t) \neq p_{s2}(t)$).

Nous étendons maintenant la définition de l’intersection aux suites de bandes temporelles.

Définition 3.16 : “Intersection de deux suites de bandes temporelles”

L’intersection de deux suites de bandes temporelles synchrones SB_1 et SB_2 est une suite de bandes temporelles, notée $SB_1 \cap SB_2$, égale à l’union des suites de bandes temporelles résultant de l’intersection des bandes temporelles de même rang.

D’une manière formelle :

$$SB_1 \cap SB_2 = \{SB_1[1] \cap SB_2[1]\} \cup \dots \cup \{SB_1[n] \cap SB_2[n]\} \\ \text{où } n = \text{card}(SB_1)$$

Nous nous intéressons maintenant aux propriétés de l’intersection.

L’intersection de deux suites de bandes temporelles

- est **associative**,
 $(SB_1 \cap SB_2) \cap SB_3 = SB_1 \cap (SB_2 \cap SB_3)$
- est **commutative**,
 $SB_1 \cap SB_2 = SB_2 \cap SB_1$

Nous exprimons maintenant la signification de l’intersection des deux suites de bandes temporelles en terme de représentation des fonctions. Cette propriété fait référence aux définitions 3.11 et 3.16.

Propriété 3.3 : “Relation entre la représentation et l’intersection”

- Les fonctions représentées par l’intersection de deux suites de bandes temporelles sont celles représentées à la fois par SB_1 et par SB_2 .
- L’intersection de deux suites de bandes temporelles est vide ssi aucune fonction n’est représentée à la fois par SB_1 et par SB_2 .

D’une manière formelle :

- $\forall f(t) \in G, f(t) \text{ Rep } SB_1 \cap SB_2 \Leftrightarrow f(t) \text{ Rep } SB_1 \wedge f(t) \text{ Rep } SB_2$
- $SB_1 \cap SB_2 = \emptyset \Leftrightarrow \forall f(t) \in G, \neg(f(t) \text{ Rep } SB_1) \vee \neg(f(t) \text{ Rep } SB_2)$

3.6 **Signe d’une suite de bandes temporelles**

Nous donnons d’abord une définition pour les bandes temporelles.

Définition 3.17 : “Signe d’une bande temporelle”

Une bande temporelle B est positive (resp. négative), et on notera $B \geq 0$ (resp. $B \leq 0$), ssi pour tout couple (x, y) appartenant à B , y est positif (resp. négatif).

D’une manière formelle :

$$B \geq 0 \Leftrightarrow \forall (x, y) \in B, y \geq 0$$

$$B \leq 0 \Leftrightarrow \forall (x, y) \in B, y \leq 0$$

La figure 3.9 illustre deux bandes temporelles, une positive et une négative :

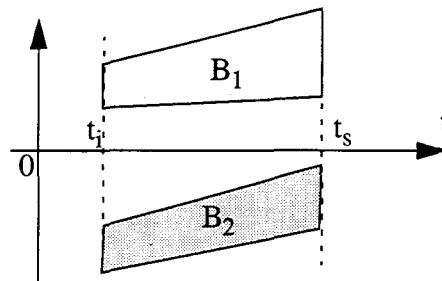


Figure 3.9 : Bandes temporelles positives et négatives

Nous caractérisons les polynômes de ces bandes temporelles :

$$(t_i, t_s, p_i, p_s) \geq 0 \Leftrightarrow \forall t \in [t_i, t_s], p_i(t) \geq 0$$

$$(t_i, t_s, p_i, p_s) \leq 0 \Leftrightarrow \forall t \in [t_i, t_s], p_s(t) \leq 0$$

Nous étendons cette définition aux suites de bandes temporelles.

Définition 3.18 : “Signe d’une suite de bandes temporelles”

Une suite de bandes temporelles SB est positive (resp. négative) ssi toutes les bandes temporelles de SB sont positives (resp. négatives).

D’une manière formelle :

$$SB \geq 0 \Leftrightarrow \forall k \in [1, n], SB[k] \geq 0$$

$$SB \leq 0 \Leftrightarrow \forall k \in [1, n], SB[k] \leq 0$$

Il découle directement de cette définition qu'une suite de bandes temporelles positive (resp. négative) représente uniquement des fonctions temporelles positives (négatives).

Chapitre 4

Opérations primitives

Ce chapitre propose de définir trois opérations sur les suites de bandes temporelles :

- l'addition,
- la multiplication,
- l'intégration.

L'objectif est de calculer les suites de bandes temporelles représentant les fonctions $f_1(t) + f_2(t)$, $f_1(t) * f_2(t)$ et $\int_{\theta}^t f_1(x)dx$ à partir des suites de bandes temporelles représentant $f_1(t)$ et $f_2(t)$.

Ces opérations seront d'abord définies pour les bandes temporelles, puis ces définitions sont étendues aux suites de bandes temporelles synchrones. Enfin, nous caractérisons leurs propriétés.

Ces opérations sont appelées opérations primitives car toutes les opérations peuvent être définies à partir de ces trois opérations.

4.1 Somme de deux suites de bandes temporelles

Le but de ce paragraphe est de définir la suite de bandes temporelles représentant la fonction temporelle $f_1(t) + f_2(t)$ à partir des suites de bandes temporelles représentant $f_1(t)$ et $f_2(t)$.

4.1.1 Définitions

Cette opération est d'abord définie pour deux bandes temporelles.

Définition 4.1 : “Somme de deux bandes temporelles”

La somme de deux bandes temporelles B_1, B_2 de même extension temporelle $[t_i, t_s]$, notée $B_1 +_B B_2$, est l'ensemble des couples de réels $(x, y_1 + y_2)$ tels que les couples (x, y_1) et (x, y_2) appartiennent respectivement aux bandes temporelles B_1 et B_2 .

D'une manière formelle :

$$B_1 +_B B_2 = \{(x, y_1 + y_2) / (x, y_1) \in B_1, (x, y_2) \in B_2\}$$

Nous montrons maintenant que la somme de deux bandes temporelles est interne à l'ensemble des bandes temporelles.

Lemme 4.1 :

La somme de deux bandes temporelles B_1, B_2 de même extension temporelle $[t_i, t_s[$ est la bande temporelle dont

- l'extension temporelle est identique à celle de B_1 ,
- le polynôme inférieur est la somme des polynômes inférieurs des bandes temporelles B_1 et B_2 ,
- le polynôme supérieur est la somme des polynômes supérieurs des bandes temporelles B_1 et B_2 .

D'une manière formelle :

$$(t_i, t_s, P_{i1}, P_{s1}) +_B (t_i, t_s, P_{i2}, P_{s2}) = (t_i, t_s, P_{i1} + P_{i2}, P_{s1} + P_{s2})$$

La preuve de ce lemme est immédiate.

La figure 4.1 décrit deux bandes temporelles et leur somme.

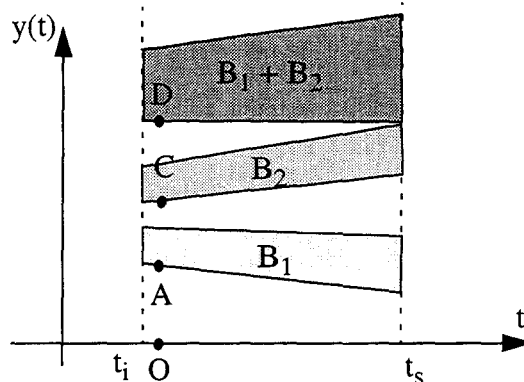


Figure 4.1 : Somme de deux bandes temporelles

Il découle directement de la définition de la somme de deux bandes temporelles que la mesure algébrique \overline{OD} est la somme des mesures algébriques \overline{OA} et \overline{OC} .

Nous étendons maintenant cette opération aux suites de bandes temporelles.

Définition 4.2 : “Somme de deux suites de bandes temporelles synchrones”

La somme de deux suites de bandes temporelles de taille n et synchrones SB_1 et SB_2 est une suite de bandes temporelles SB , notée $SB_1 +_B SB_2$ (synchrones avec SB_1) dont les bandes temporelles sont égales à la somme des bandes temporelles de SB_1 et SB_2 .

D'une manière formelle :

$$\forall k \in [1, n], SB[k] = SB_1[k] +_B SB_2[k]$$

Tous les couples de suites de bandes temporelles ne sont pas synchrones. Nous définissons la somme de ces couples.

Définition 4.3 : “Somme de deux suites de bandes temporelles quelconques”

La somme de deux suites de bandes temporelles SB_1 et SB_2 est la somme des suites de bandes temporelles SB'_1 et SB'_2 , telles que le couple (SB'_1, SB'_2) résulte de la synchronisation de (SB_1, SB_2) .

Comme il existe un et un seul couple de suites de bandes temporelles synchrones minimal, la somme de deux suites de bandes temporelles est définie de manière unique. C'est à dire qu'il existe une et une seule suite de bandes temporelles égale à la somme de deux suites de bandes temporelles. Cette suite n'est pas nécessairement minimale.

Exemple :

$$\{(1, 2, 2, 4), (2, 3, 3, 5)\} +_B \{(1, 2, 2, 4), (2, 3, 1, 3)\} = \{(1, 2, 4, 8), (2, 3, 4, 8)\}$$

et

$$\{(1, 2, 4, 8), (2, 3, 4, 8)\} = \{(1, 3, 4, 8)\}.$$

Toutefois, en règle générale, la somme de deux suites de bandes temporelles minimales est une suite minimale.

Par la suite, la notation (abusive) $SB_1 + SB_2$ sera utilisée.

Nous nous intéressons maintenant aux propriétés de cette opération.

4.1.2 Propriétés

Nous définissons d'abord quelques propriétés générales. La somme de deux suites de bandes temporelles :

- est **associative**,

$$SB_1 + (SB_2 + SB_3) = (SB_1 + SB_2) + SB_3$$
- est **commutative**,

$$SB_1 + SB_2 = SB_2 + SB_1$$
- **possède un unique élément neutre** $\{(t_i, t_s, \mathbf{0}, \mathbf{0})\}$ pour les suites de bandes temporelles d'extension temporelle égale à $[t_i, t_s]$.

$$SB + \{(t_i, t_s, \mathbf{0}, \mathbf{0})\} = \{(t_i, t_s, \mathbf{0}, \mathbf{0})\} + SB = SB$$

Par contre, l'addition de deux suites de bandes temporelles ne possède pas d'inverse. En effet, l'inverse de la bande temporelle (t_i, t_s, p_i, p_s) serait $(t_i, t_s, -p_i, -p_s)$. Or, $(t_i, t_s, -p_i, -p_s)$ n'est pas toujours une bande temporelle car la contrainte $-p_i(t) \leq -p_s(t)$ n'est pas toujours respectée. (En effet, on a $p_i(t) \leq p_s(t)$ car (t_i, t_s, p_i, p_s) est une bande temporelle et $p_i(t) \leq p_s(t) \Rightarrow -p_i(t) \geq -p_s(t)$). Par conséquent, seules les bandes temporelles de largeur nulle (i.e. $p_i(t) = p_s(t)$) ont un inverse.

Nous exprimons maintenant quelques propriétés spécifiques aux suites de bandes temporelles.

Propriété 4.1 : “*Conservation de la propriété de représentation*”

L'addition de deux suites de bandes temporelles conserve la propriété de représentation, c'est à dire que, pour toutes grandeurs $g_1(t)$ et $g_2(t)$ représentées respectivement par les suites de bandes temporelles SB_1 et SB_2 , la fonction temporelle $f(t) = g_1(t) + g_2(t)$ est représentée par la suite de bandes temporelles $SB_1 + SB_2$.

D'une manière formelle :

$$\forall g_1(t) \in \mathbf{G}, \forall g_2(t) \in \mathbf{G}, g_1(t) \text{ Rep } SB_1 \wedge g_2(t) \text{ Rep } SB_2 \Rightarrow g_1(t) + g_2(t) \text{ Rep } (SB_1 + SB_2)$$

Preuve :

Soient SB_1 et SB_2 deux suites de bandes temporelles.

Soient $SB'_1 = \{(t_1, t_2, pi1_1, ps1_1), \dots, (t_n, t_{n+1}, pi1_n, ps1_n)\}$ et

$SB'_2 = \{(t_1, t_2, pi2_1, ps2_1), \dots, (t_n, t_{n+1}, pi2_n, ps2_n)\}$ les suites de bandes temporelles telles que

$SB_1 \sim SB'_1$ et $SB_2 \sim SB'_2$ et (SB'_1, SB'_2) synchrone minimal.

Par définition de la somme de deux suites de bandes temporelles, on a :

$SB_1 + SB_2 = \{(t_1, t_2, pi1_1 + pi2_1, ps1_1 + ps2_1), \dots, (t_n, t_{n+1}, pi1_n + pi2_n, ps1_n + ps2_n)\}$

D'après la propriété 3.1, on a :

$\forall g_1(t) \in \mathbf{G}, g_1(t) \text{ Rep } SB_1 \wedge SB_1 \sim SB'_1 \Rightarrow g_1(t) \text{ Rep } SB'_1$

$\forall g_2(t) \in \mathbf{G}, g_2(t) \text{ Rep } SB_2 \wedge SB_2 \sim SB'_2 \Rightarrow g_2(t) \text{ Rep } SB'_2$

Par définition de la propriété de représentation, on a :

$g_1(t) \text{ Rep } SB'_1 \Leftrightarrow \forall k \in [1, n], \forall t_0 \in [t_k, t_{k+1}[, pi1_k(t_0) \leq g(t_0) \leq ps1_k(t_0)$

D'où $\forall g_1(t) \in \mathbf{G}, \forall g_2(t) \in \mathbf{G},$

$g_1(t) \text{ Rep } SB_1 \wedge g_2(t) \text{ Rep } SB_2$

$\Rightarrow g_1(t) \text{ Rep } SB'_1 \wedge g_2(t) \text{ Rep } SB'_2$

$\Rightarrow \forall k \in [1, n], \forall t_0 \in [t_k, t_{k+1}[, pi1_k(t_0) \leq g(t_0) \leq ps1_k(t_0) \wedge pi2_k(t_0) \leq g(t_0) \leq ps2_k(t_0)$

$\Rightarrow \forall k \in [1, n], \forall t_0 \in [t_k, t_{k+1}[, pi1_k(t_0) + pi2_k(t_0) \leq g(t_0) \leq ps1_k(t_0) + ps2_k(t_0)$

$\Rightarrow g_1(t) + g_2(t) \text{ Rep } SB'_1 + SB'_2$

□

Nous introduisons maintenant la deuxième propriété de la somme.

Propriété 4.2 : "Non introduction d'imprécision"

La somme de deux suites de bandes temporelles continues n'introduit pas d'imprécision, c'est à dire que, toute fonction temporelle continue $g(t)$ représentée par la somme de deux suites de bandes temporelles $SB_1 + SB_2$, est la somme de deux fonctions temporelles continues $g_1(t)$ et $g_2(t)$ représentées par les suites de bandes temporelles continues SB_1 et SB_2 .

D'une manière formelle :

$\forall g(t) \in \mathbf{G}, g(t) \text{ Rep } SB_1 + SB_2 \Rightarrow \exists g_1(t) \in \mathbf{G}, \exists g_2(t) \in \mathbf{G},$

$g_1(t) \text{ Rep } SB_1 \wedge g_2(t) \text{ Rep } SB_2 \wedge g(t) = g_1(t) + g_2(t)$

Preuve :

Soient SB_1 et SB_2 deux suites de bandes temporelles continues.

Soient $SB'_1 = \{(t_1, t_2, pi1_1, ps1_1), \dots, (t_n, t_{n+1}, pi1_n, ps1_n)\}$ et

$SB'_2 = \{(t_1, t_2, pi2_1, ps2_1), \dots, (t_n, t_{n+1}, pi2_n, ps2_n)\}$ les suites de bandes temporelles telles que

$SB_1 \sim SB'_1$ et $SB_2 \sim SB'_2$ et (SB'_1, SB'_2) synchrone minimal.

Par définition de la somme de deux suites de bandes temporelles, on a :

$SB_1 + SB_2 = \{(t_1, t_2, pi1_1 + pi2_1, ps1_1 + ps2_1), \dots, (t_n, t_{n+1}, pi1_n + pi2_n, ps1_n + ps2_n)\}$

Soient les fonctions :

$g_1(t) = pi1_k(t) \quad \text{si } t \in [t_k, t_{k+1}[\text{ et } g(t) \in [pi1_k(t) + pi2_k(t), pi1_k(t) + ps2_k(t)]$

$= g(t) - ps2_k(t) \quad \text{si } t \in [t_k, t_{k+1}[\text{ et } g(t) \in]pi1_k(t) + ps2_k(t), ps1_k(t) + ps2_k(t)]$

et $g_2(t) = g(t) - g_1(t)$

Montrons que $g_1(t)$ est continue.

On sait d'après la continuité de SB'_1 et SB'_2 que:

$$pi1_k(t) \xrightarrow[t \rightarrow t_{k+1}^-]{} pi1_{k+1}(t_{k+1}) \text{ et } ps2_k(t) \xrightarrow[t \rightarrow t_{k+1}^-]{} ps2_{k+1}(t_{k+1})$$

1) Si $g(t_{k+1}) < pi1_{k+1}(t_{k+1}) + ps2_{k+1}(t_{k+1})$ alors, par continuité, on a:

$$\exists \alpha > 0, \forall t \in [t_{k+1} - \alpha, t_{k+1}[, g(t) < pi1_k(t_k) + ps2_k(t_k).$$

$$\text{D'où } g_1(t) = pi1_k(t) \text{ sur } [t_{k+1} - \alpha, t_{k+1}[\text{ et } g_1(t_{k+1}) = pi1_{k+1}(t_{k+1}), \text{ donc } g_1(t) \xrightarrow[t \rightarrow t_{k+1}^-]{} g_1(t_{k+1})$$

2) Si $g(t_{k+1}) > pi1_{k+1}(t_{k+1}) + ps2_{k+1}(t_{k+1})$ alors, par continuité, on a:

$$\exists \alpha > 0, \forall t \in [t_{k+1} - \alpha, t_{k+1}[, g(t) > pi1_k(t_k) + ps2_k(t_k).$$

$$\text{D'où } g_1(t) = g(t) - ps2_k(t) \text{ sur } [t_{k+1} - \alpha, t_{k+1}[\text{ et } g_1(t_{k+1}) = g(t_{k+1}) - ps2_{k+1}(t_{k+1}),$$

$$\text{donc } g_1(t) \xrightarrow[t \rightarrow t_{k+1}^-]{} g_1(t_{k+1})$$

3) Si $g(t_{k+1}) = pi1_{k+1}(t_{k+1}) + ps2_{k+1}(t_{k+1})$ alors $g_1(t_{k+1}) = g(t_{k+1}) - ps2_{k+1}(t_{k+1}) = pi1_{k+1}(t_{k+1})$.

$$\text{D'où } pi1_k(t) \xrightarrow[t \rightarrow t_{k+1}^-]{} g_1(t_{k+1}) \text{ et } g(t) - ps2_k(t) \xrightarrow[t \rightarrow t_{k+1}^-]{} g_1(t_{k+1}).$$

$$\text{Soit } \forall \varepsilon > 0, \exists \alpha > 0, |pi1_k(t) - g_1(t_{k+1})| < \varepsilon \text{ et } |g(t) - ps2_k(t) - g_1(t_{k+1})| < \varepsilon \text{ pour } t \in [t_{k+1} - \alpha, t_{k+1}[.$$

$$\text{D'où } \forall \varepsilon > 0, \exists \alpha > 0, |g_1(t) - g_1(t_{k+1})| < \varepsilon \text{ pour } t \in [t_{k+1} - \alpha, t_{k+1}[. \text{ Soit}$$

$$\text{Soit } g_1(t) \xrightarrow[t \rightarrow t_{k+1}^-]{} g_1(t_{k+1})$$

De plus, $g_2(t)$ est continue car c'est la somme de deux fonctions continues.

Montrons que les fonctions $g_1(t)$ et $g_2(t)$ sont respectivement représentées par SB'_1 et SB'_2 .

Si $g_1(t) = pi1_k(t)$, alors $g_1(t)$ est représentée par $SB'_1[k]$ (par définition de la propriété de représentation).

De plus, $g_2(t) = g(t) - g_1(t) = g(t) - pi1_k(t)$ est représentée par $SB'_2[k]$

$$\text{car } g(t) \in]pi1_k(t) + pi2_k(t), pi1_k(t) + ps2_k(t)] \Rightarrow g_2(t) \in [pi2_k(t), ps2_k(t)]$$

Si $g_1(t) = g(t) - ps2_k(t)$, alors $g_1(t)$ est représentée par $SB'_1[k]$ car

$$g(t) \in]pi1_k(t) + ps2_k(t), ps1_k(t) + ps2_k(t)] \Rightarrow g(t) - ps2_k(t) \in]pi1_k(t), ps1_k(t)]$$

et $g_2(t) = ps2_k(t)$ est représentée par $SB'_2[k]$.

Donc, $g_1(t)$ et $g_2(t)$ sont toujours respectivement représentées par une des bandes temporelles de SB'_1 et SB'_2 , et sont donc représentées par SB'_1 et SB'_2 .

Il existe donc deux fonctions continues $g_1(t)$ et $g_2(t)$ représentées respectivement par SB'_1 et SB'_2 (et donc aussi par SB_1 et SB_2) et vérifiant $g(t) = g_1(t) + g_2(t)$.

□

Exemple :

Soient les bandes temporelles $B_1 = \{(2, 4, p_i, p_s)\}$ et $B_2 = \{2, 4, 1, 2\}$ où $p_i(t) = t + 1$ et $p_s(t) = 2t + 2$.

Soient les fonctions temporelles $g(t) = 4t - 4$,

$$g_1(t) = t + 1 \quad \text{si } t \in [2, 7/3] \quad g_2(t) = 3t - 5 \quad \text{si } t \in [2, 7/3]$$

$$4t - 6 \quad \text{si } t \in]7/3, 4] \quad 2 \quad \text{si } t \in]7/3, 4]$$

Comme le montre la figure 4.2, les fonctions $g(t)$, $g_1(t)$ et $g_2(t)$ sont continues, sont

respectivement représentées par $B_1 + B_2$, B_1 et B_2 et vérifient $g_1(t) + g_2(t) = g(t)$.

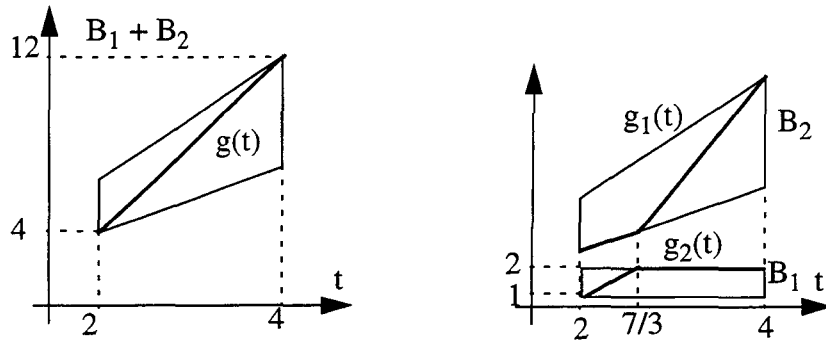


Figure 4.2 : Propriété de non introduction d'imprécision

Par contre, cette propriété ne tient pas pour la somme de deux suites de bandes temporelles non continues.

En effet, soient les suites de bandes temporelles

$$SB_1 = \{(t_1, t_2, 2, 4), (t_2, t_3, 2, 5)\} \text{ et } SB_2 = \{(t_1, t_2, 1, 3), (t_2, t_3, 1, 2)\}$$

Le calcul sur les bandes temporelles nous assure que :

$$SB_1 + SB_2 = \{(t_1, t_2, 3, 7), (t_2, t_3, 3, 7)\}$$

Soit $g(t)$ une fonction temporelle représentée par $SB_1 + SB_2$ telle que $g(t_2) > 6$.

Cherchons deux fonctions $g_1(t)$ et $g_2(t)$ représentées par SB_1 et SB_2 telles que $g(t) = g_1(t) + g_2(t)$.

Toute fonction $g_2(t)$ représentée par $SB_2[2]$ sera telle que $g_2(t) \leq 2$. Ce qui implique que la fonction associée $g_1(t)$ représentée par $SB_1[2]$ sera telle que $g_1(t_2) > 4$ (car $g_1(t_2) = g(t_2) - g_2(t_2)$)

Or, toute fonction $g_1(t)$ représentée par $SB_1[1]$ sera telle que $g_1(t) \leq 4$.

Donc, la fonction $g_1(t)$ ne peut pas être continue.

Nous étendons maintenant cette opération à la somme d'un intervalle à une suite de bandes temporelles.

4.1.3 Somme d'un intervalle et d'une suite de bandes temporelles

Le but de cette opération est de déterminer une suite de bandes temporelles représentant la fonction $f(t) + c$ où c est une constante réelle appartenant à un intervalle $[a, b]$.

Nous remarquons pour cela que la fonction temporelle $g(t) = c$ appartient à toute bande (t_i, t_s, a, b) où t_i et t_s sont deux instants quelconques.

Définition 4.4 : "Somme d'un intervalle et d'une suite de bandes temporelles"

La somme d'un intervalle $[a, b]$ et d'une suite de bandes temporelles SB , notée $[a, b] +_1 SB$, est la suite de bandes temporelles égale à la somme des suites de bandes temporelles SB et $\{(t_i(SB), t_s(SB), a, b)\}$.

D'une manière formelle :

$$[a, b] +_1 SB = \{(t_i(SB), t_s(SB), a, b)\} +_B SB$$

Les opérations $[a, b] +_I SB$ et $\{(t_i(SB), t_s(SB), a, b)\} +_B SB$ fournissent par définition le même résultat. Toutefois, ces opérations ne sont pas identiques car elles font intervenir des objets de natures différentes. Nous allons montrer qu'elles n'ont pas les mêmes propriétés.

Les relations suivantes découlent directement de la propriété d'associativité de l'addition de deux bandes temporelles. Soient deux intervalles I_1 et I_2 et deux suites de bandes temporelles SB_1 et SB_2 :

- $I_1 +_I (I_2 +_B SB_1) = (I_1 \oplus I_2) +_B SB_1$ où \oplus est la somme de deux intervalles [Moore, 66]
- $I_1 +_I (SB_1 +_B SB_2) = (I_1 +_I SB_1) +_B SB_2$

Comme pour la somme de deux suites de bandes temporelles, la notation (abusive) $[a, b] + SB$ sera utilisée.

Deux propriétés de la somme de deux suites de bandes temporelles ont été définies (propriétés 4.1 et 4.2). Nous vérifions si ces propriétés restent valables pour la somme d'une suite de bandes temporelles et d'un intervalle :

Propriété 4.3 : “Conservation de la propriété de représentation”

La somme d'une suite de bandes temporelles et d'un intervalle conserve la propriété de représentation, c'est à dire que, pour toute grandeur $g(t)$ représentée par la suite de bandes temporelles SB_1 et toute constante c appartenant à $[a, b]$, la fonction temporelle $f(t) = g(t) + c$ est représentée par la suite de bandes temporelles $[a, b] + SB$.

D'une manière formelle :

$$\forall g(t) \in \mathbf{G}, \forall c \in [a, b], g(t) \text{ Rep } SB \Rightarrow g(t) + c \text{ Rep } (SB + [a, b])$$

La preuve découle directement de la propriété équivalente sur la somme de deux suites de bandes temporelles et du fait que toute fonction $f(t) = c$, où c appartient à $[a, b]$, est représentée par la suite de bandes temporelles $\{(t_i(SB), t_s(SB), a, b)\}$.

Propriété 4.4 : “Introduction d'imprécision”

La somme d'un intervalle $[a, b]$ et d'une suite de bandes temporelles continue SB augmente l'imprécision, c'est à dire que, en général, il existe une fonction temporelle représentée par $SB + [a, b]$ qui n'est pas la somme d'une fonction temporelle $f(t)$ représentée par SB et d'une constante c appartenant à $[a, b]$.

Exemple :

Soit la suite de bandes temporelles $SB = \{(2, 4, p_i, p_s)\}$ où $p_i = t + 1$ et $p_s = 2t + 2$.

La fonction temporelle $g(t) = 4t - 4$, illustrée figure 4.3, est représentée par $SB + [1, 2]$, mais n'est pas la somme d'une fonction temporelle $f(t)$ représentée par SB et d'une constante c appartenant à $[1, 2]$.

En effet, $g(4) = 12$ implique que $c = 2$ (car $f(4) \leq p_s(4) = 10$).

et $g(2) = 4$ implique que $c = 1$ (car $f(2) \geq p_i(2) = 3$).

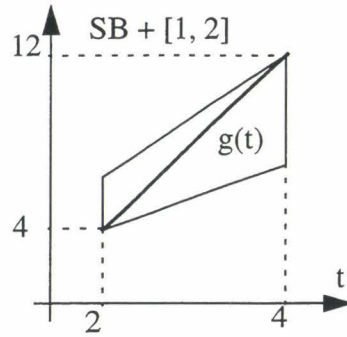


Figure 4.3 : Introduction d'imprécision de l'opération d'addition d'un intervalle à une suite de bandes temporelles

Le fait que l'addition d'un intervalle $[a, b]$ et d'une suite de bandes temporelles SB introduise de l'imprécision signifie que, en général, il existe une fonction temporelle $g(t)$ représentée par $[a, b] + SB$ alors qu'elle ne devrait pas être représentée par cette suite (i.e. $g(t)$ n'est pas la somme d'une fonction représentée par SB et d'une constante appartenant à $[a, b]$).

Nous posons alors le problème suivant : existe-t'il une suite de bandes temporelles ne représentant pas $g(t)$ et conservant la propriété de représentation. La propriété suivante répond négativement à cette question.

Propriété 4.5 : "Meilleure représentation ou semi-imprécision"

La somme d'un intervalle $[a, b]$ et d'une suite de bandes temporelles continue SB est la meilleure représentation, c'est à dire que, pour tout couple (x, y) appartenant à $SB + [a, b]$, il existe une fonction $g(t)$ représentée par SB et une constante c appartenant à $[a, b]$ telles que $g(x) + c = y$.

D'une manière formelle :

Soit SB une suite de bandes temporelles continue.

$$\forall (x, y) \in SB + [a, b], \exists g(t) \text{ Rep } SB, \exists c \in [a, b], g(x) + c = y$$

Preuve :

La preuve découle de la propriété de non introduction d'imprécision de l'addition de deux bandes temporelles :

$$\forall g(t) \text{ Rep } B_1 + B_2, \exists g_1(t) \text{ Rep } B_1, \exists g_2(t) \text{ Rep } B_2, g(t) = g_1(t) + g_2(t)$$

Il en découle que pour toutes bandes temporelles $B_1 = (t_i, t_s, p_{i1}, p_{s1})$ et $B_2 = (t_i, t_s, p_{i2}, p_{s2})$, on a :

$$\forall (x, y) \in (t_i, t_s, p_{i1} + p_{i2}, p_{s1} + p_{s2}),$$

$$\exists g_1(t) \text{ Rep } (t_i, t_s, p_{i1}, p_{s1}), \exists g_2(t) \text{ Rep } (t_i, t_s, p_{i2}, p_{s2}), y = g_1(x) + g_2(x)$$

En remplaçant p_{i2} et p_{s2} par deux réels a et b et en posant $g_2(x) = c$, on obtient :

$$\forall (x, y) \in (t_i, t_s, p_{i1} + a, p_{s1} + b), \exists g_1(t) \text{ Rep } (t_i, t_s, p_{i1}, p_{s1}), \exists c \in [a, b], y = g_1(x) + c$$

Cette preuve s'étend naturellement aux suites de bandes temporelles continues.

□

On a vu qu'en règle générale la suite de bandes temporelles résultant de l'addition d'un intervalle $[a, b]$ à une suite de bandes temporelles SB représente des fonctions $g(t)$ qui ne

devraient pas être représentées par $SB + [a, b]$.

La propriété 4.5 signifie que,

- soit l'addition d'un intervalle $[a, b]$ à une suite de bandes temporelles SB n'aura pas la propriété de non introduction d'imprécision (comme la définition que nous avons donnée),
- soit l'addition d'un intervalle $[a, b]$ à une suite de bandes temporelles SB n'aura pas la propriété de conservation de la représentation (ce qui se passerait si l'on redéfinissait l'addition afin d'obtenir la propriété de non introduction d'imprécision).

De manière générale, la représentation des fonctions temporelles par des suites de bandes temporelles ne permet pas de conserver les contraintes sur les fonctions temporelles. En d'autres termes, il n'est pas explicite que la suite de bandes temporelles $[a, b] + SB$ représente la somme de deux fonctions temporelles dont une est constante.

Nous allons maintenant caractériser une manière de déterminer les fonctions qui devraient être représentées par $SB + [a, b]$. Nous exprimons d'abord la propriété de non introduction d'imprécision de l'addition d'un intervalle dégénéré à une suite de bandes temporelles.

Propriété 4.6 : *“Non introduction d'imprécision pour un intervalle dégénéré”*

L'addition d'un intervalle dégénéré $[a, a]$ à une suite de bandes temporelles SB n'introduit pas d'imprécision, c'est à dire que, si une fonction temporelle $g(t)$ est représentée par $SB + [a, a]$, alors la fonction temporelle $g(t) - a$ est représentée par SB .

D'une manière formelle :

$$\forall g(t) \in \mathbf{G}, g(t) \text{ Rep } [a, a] + SB \Rightarrow g(t) - a \text{ Rep } SB$$

La preuve découle immédiatement de la définition de la propriété de représentation.

Il découle de cette propriété que, pour savoir si une fonction temporelle $g(t)$ devrait être représentée par une suite de bandes temporelles $[a, b] + SB$, une condition nécessaire et suffisante est l'existence d'un réel c appartenant à l'intervalle $[a, b]$ telle que la fonction temporelle $g(t) - c$ soit représentée par $SB + [c, c]$. Un tel c est nécessairement au plus égal au minimum de la différence entre la fonction $g(t)$ et les polynômes inférieurs des bandes temporelles de SB .

Par exemple, la fonction $g(t) = 4t - 4$ illustrée figure 4.3 ne devrait pas être représentée par la suite de bandes temporelles $SB = \{(2, 4, p_i, p_s)\} + [1, 2]$ ($p_i = t + 1$ et $p_s = 2t + 2$) car $\min(g(t) - p_i(t)) = 1$ et la fonction $f(t) = g(t) - 1 = 4t - 5$ n'est pas représentée par SB car $f(4) = 11 > p_s(4) = 10$.

Nous définissons maintenant la deuxième opération primitive.

4.2 Multiplication de deux suites de bandes temporelles

Le but de ce paragraphe est de définir la suite de bandes temporelles représentant la fonction temporelle $f_1(t) * f_2(t)$ à partir des suites de bandes temporelles représentant $f_1(t)$ et $f_2(t)$.

4.2.1 Définitions

Cette opération est d'abord définie pour deux bandes temporelles.

Définition 4.5 : "Multiplication de deux bandes temporelles"

La multiplication de deux bandes temporelles B_1, B_2 de même extension temporelle $[t_i, t_s]$, notée $B_1 *_B B_2$, est l'ensemble des couples de réels (x, y_1y_2) tels que les couples (x, y_1) et (x, y_2) appartiennent respectivement aux bandes temporelles B_1 et B_2 .

D'une manière formelle :

$$B_1 *_B B_2 = \{(x, y_1y_2) / (x, y_1) \in B_1, (x, y_2) \in B_2 \}$$

A partir des définitions des bandes temporelles et de l'appartenance d'un couple à une bande temporelle (définitions 2.1 et 2.2), la définition formelle précédente (définition 4.5) peut se réécrire :

$$\begin{aligned} & (t_i, t_s, p_{i1}, p_{s1}) *_B (t_i, t_s, p_{i2}, p_{s2}) \\ &= \{(x, y_1y_2) / x \in [t_i, t_s[, y_1 \in [p_{i1}(x), p_{s1}(x)], y_2 \in [p_{i2}(x), p_{s2}(x)]\} \\ &= \{(x, y) / x \in [t_i, t_s[, y \in [p_{i1}(x), p_{s1}(x)] \otimes [p_{i2}(x), p_{s2}(x)]\} \end{aligned}$$

où \otimes est la multiplication de deux intervalles :

$$[a, b] \otimes [c, d] = [\min(ac, bc, ad, bd), \max(ac, bc, ad, bd)]$$

D'où $(t_i, t_s, p_{i1}, p_{s1}) *_B (t_i, t_s, p_{i2}, p_{s2})$

$$= \{(x, y) / x \in [t_i, t_s[, y \in [\min(p_{i1}(x)p_{i2}(x), p_{s1}(x)p_{i2}(x), p_{i1}(x)p_{s2}(x), p_{s1}(x)p_{s2}(x)), \max(p_{i1}(x)p_{i2}(x), p_{s1}(x)p_{i2}(x), p_{i1}(x)p_{s2}(x), p_{s1}(x)p_{s2}(x))]\} \}$$

Exemple :

Le produit des bandes temporelles $B_1 = (0, 2.5, p_{i1}, p_{s1})$ et $B_2 = (0, 2.5, p_{i2}, p_{s2})$, où $p_{i1}(t) = 0.5t - 2$, $p_{s1}(t) = t - 1$, $p_{i2}(t) = -3t + 2$ et $p_{s2}(t) = -2t + 3$, est l'ensemble des couples illustré figure 4.4.

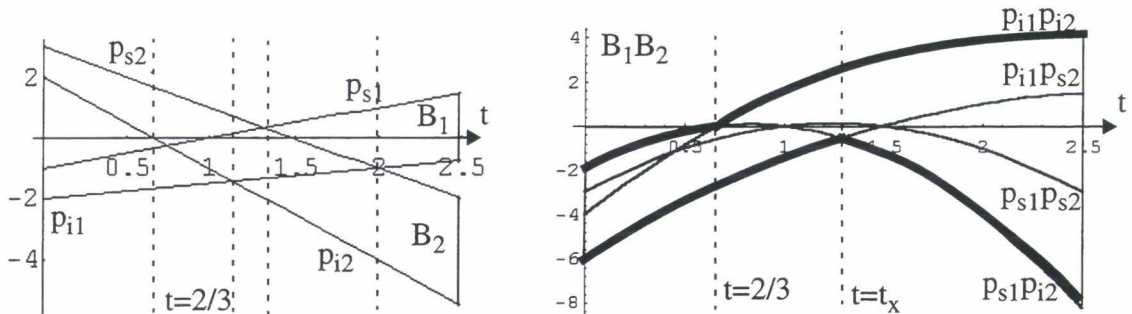


Figure 4.4 : Multiplication de deux bandes temporelles

La multiplication de deux bandes temporelles n'est pas, en règle générale, une bande temporelle, sauf pour certaines bandes temporelles, que nous étudions maintenant.

La multiplication de deux bandes temporelles est définie de la façon suivante :

$$\begin{aligned} & (t_i, t_s, p_{i1}, p_{s1}) *_B (t_i, t_s, p_{i2}, p_{s2}) \\ &= \{(x, y) / x \in [t_i, t_s[, y \in [\min(p_{i1}(x)p_{i2}(x), p_{s1}(x)p_{i2}(x), p_{i1}(x)p_{s2}(x), p_{s1}(x)p_{s2}(x)), \max(p_{i1}(x)p_{i2}(x), p_{s1}(x)p_{i2}(x), p_{i1}(x)p_{s2}(x), p_{s1}(x)p_{s2}(x))]\} \}. \end{aligned}$$

Pour que $(t_i, t_s, p_{i1}, p_{s1}) *_B (t_i, t_s, p_{i2}, p_{s2})$ soit une bande temporelle, il faut donc que les fonctions $\min(p_{i1}(x)p_{i2}(x), p_{s1}(x)p_{i2}(x), p_{i1}(x)p_{s2}(x), p_{s1}(x)p_{s2}(x))$ et

$\max(p_{i1}(x)p_{i2}(x), p_{s1}(x)p_{i2}(x), p_{i1}(x)p_{s2}(x), p_{s1}(x)p_{s2}(x))$ soient des polynômes.

Une condition suffisante évidente est que l'un des quatre produits de polynômes est toujours supérieur aux trois autres et que l'un est toujours inférieur.

Exemple (suite) :

Le produit des bandes temporelles $B_1 = (0, 2/3, p_{i1}, p_{s1})$ et $B_2 = (0, 2/3, p_{i2}, p_{s2})$ est la bande temporelle $(0, 2/3, p_{i1}p_{s2}, p_{s1}p_{i2})$. (Le polynôme $p_{i1}p_{s2}$ est toujours inférieur aux polynômes $p_{i1}p_{i2}$, $p_{s1}p_{i2}$ et $p_{s1}p_{s2}$, et le polynôme $p_{s1}p_{i2}$ est toujours supérieur aux trois autres).

La définition suivante découle directement de la remarque ci dessus :

Définition 4.6 : “Bandes temporelles compatibles pour le produit”

Deux bandes temporelles $(t_i, t_s, p_{i1}, p_{s1})$ et $(t_i, t_s, p_{i2}, p_{s2})$ sont compatibles pour le produit ssi un des quatre produits de polynômes $p_{i1}p_{i2}$, $p_{i1}p_{s2}$, $p_{s1}p_{i2}$ et $p_{s1}p_{s2}$ est toujours inférieur aux trois autres et un est toujours supérieur.

D'une manière formelle :

$(t_i, t_s, p_{i1}, p_{s1})$ et $(t_i, t_s, p_{i2}, p_{s2})$ sont compatibles pour le produit ssi

$\forall t \in [t_i, t_s[$, $p_i(t) = \min(p_{i1}(t)p_{i2}(t), p_{s1}(t)p_{i2}(t), p_{i1}(t)p_{s2}(t), p_{s1}(t)p_{s2}(t))$

$\wedge (p_i = p_{i1}p_{i2} \vee p_i = p_{i1}p_{s2} \vee p_i = p_{s1}p_{i2} \vee p_i = p_{s1}p_{s2})$

$\wedge \forall t \in [t_i, t_s[$, $p_s(t) = \max(p_{i1}(t)p_{i2}(t), p_{s1}(t)p_{i2}(t), p_{i1}(t)p_{s2}(t), p_{s1}(t)p_{s2}(t))$

$\wedge (p_s = p_{i1}p_{i2} \vee p_s = p_{i1}p_{s2} \vee p_s = p_{s1}p_{i2} \vee p_s = p_{s1}p_{s2})$

Exemple (suite) :

Les bandes temporelles $B_1 = (0, 2/3, p_{i1}, p_{s1})$ et $B_2 = (0, 2/3, p_{i2}, p_{s2})$ sont compatibles pour le produit.

Nous avons vu que la multiplication de deux bandes temporelles compatibles pour le produit est une bande temporelle (ses polynômes inférieur et supérieur sont égaux à p_i et p_s). Toutefois, la définition des bandes temporelles compatibles fait intervenir les opérations minimum et maximum de quatre fonctions (polynomiales). Ces opérations ne sont pas calculatoires. Nous présentons donc une condition suffisante pour détecter si le minimum ou le maximum de quatre fonctions (polynomiales) est une des fonctions.

Lemme 4.2 :

Soient deux bandes temporelles $B_1 = (t_i, t_s, p_{i1}, p_{s1})$ et $B_2 = (t_i, t_s, p_{i2}, p_{s2})$.

Si les produits de polynômes $p_{i1}p_{i2}(t)$, $p_{s1}p_{i2}(t)$, $p_{i1}p_{s2}(t)$ et $p_{s1}p_{s2}(t)$ sont différents pour tout instant t de $]t_i, t_s[$, alors les bandes temporelles B_1 et B_2 sont compatibles pour le produit.

D'une manière formelle :

$\forall t \in [t_i, t_s[$, $p_{i1}(t)p_{i2}(t) \neq p_{s1}(t)p_{i2}(t) \wedge p_{i1}(t)p_{i2}(t) \neq p_{i1}(t)p_{s2}(t) \wedge p_{i1}(t)p_{i2}(t) \neq p_{s1}(t)p_{s2}(t)$

$\wedge p_{s1}(t)p_{i2}(t) \neq p_{i1}(t)p_{s2}(t) \wedge p_{s1}(t)p_{i2}(t) \neq p_{s1}(t)p_{s2}(t) \wedge p_{i1}(t)p_{s2}(t) \neq p_{s1}(t)p_{s2}(t)$

$\Rightarrow B_1$ et B_2 compatibles pour le produit

La condition de ce lemme permet d'assurer qu'un des polynômes est toujours supérieur aux

trois autres et que l'un est toujours inférieur.

Une preuve par l'absurde est immédiate : si l'un des polynômes n'était pas toujours inférieur aux trois autres, alors deux des polynômes seraient égaux pour un instant.

Il découle de ce lemme qu'une méthode pour vérifier que l'un des polynômes p_1 et p_2 est toujours inférieur à l'autre sur $]t_1, t_2[$ consiste à calculer les solutions de l'équation $p_1(t) = p_2(t)$, puis à vérifier que celles-ci n'appartiennent pas à $]t_1, t_2[$. Trouver les polynômes inférieur et supérieur consiste alors à calculer, puis à comparer les valeurs $p_1(t)$ et $p_2(t)$ pour un quelconque instant t de $]t_1, t_2[$. Afin de faciliter l'écriture des définitions, les opérations inf et sup seront réutilisées (définition 3.14).

Nous définissons maintenant la multiplication de deux bandes temporelles compatibles pour le produit.

Définition 4.7 : *“Multiplication de deux bandes temporelles compatibles pour le produit”*

La multiplication de deux bandes temporelles compatibles pour le produit $(t_i, t_s, p_{i1}, p_{s1})$ et $(t_i, t_s, p_{i2}, p_{s2})$ est une bande temporelle dont l'extension temporelle est $[t_i, t_s[$ et dont les polynômes inférieur et supérieur sont respectivement égaux aux minimum et maximum des quatre polynômes $p_{i1}p_{i2}$, $p_{i1}p_{s2}$, $p_{s1}p_{i2}$ et $p_{s1}p_{s2}$.

D'une manière formelle :

Soient $(t_i, t_s, p_{i1}, p_{s1})$ et $(t_i, t_s, p_{i2}, p_{s2})$ deux bandes temporelles compatibles pour le produit.

$(t_i, t_s, p_{i1}, p_{s1}) *_B (t_i, t_s, p_{i2}, p_{s2})$

$= (t_i, t_s, \inf([t_i, t_s[; p_{i1}p_{i2}, p_{i1}p_{s2}, p_{s1}p_{i2}, p_{s1}p_{s2}), \sup([t_i, t_s[; p_{i1}p_{i2}, p_{i1}p_{s2}, p_{s1}p_{i2}, p_{s1}p_{s2}))$

Les définitions relatives au produit sont maintenant étendues aux suites de bandes temporelles:

Définition 4.8 : *“Suites de bandes temporelles compatibles pour le produit”*

Deux suites de bandes temporelles SB_1 et SB_2 sont compatibles pour le produit ssi

- elles sont synchrones,
- les bandes temporelles de ces suites sont deux à deux compatibles pour le produit.

D'une manière formelle :

SB_1 et SB_2 sont compatibles pour le produit ssi

(SB_1, SB_2) synchrone $\wedge \forall k \in [1, \text{card}(SB_1)]$, $SB_1[k]$ et $SB_2[k]$ compatibles pour le produit

Nous étudions maintenant l'existence de telles suites :

Lemme 4.3 :

Pour toutes bandes temporelles B_1 et B_2 , il existe deux suites de bandes temporelles SB'_1 et SB'_2 compatibles pour le produit et respectivement équivalentes aux suites de bandes temporelles $\{B_1\}$ et $\{B_2\}$.

D'une manière formelle :

$$\forall B_1, \forall B_2, \exists SB'_1, \exists SB'_2, SB'_1 \sim \{B_1\} \wedge SB'_2 \sim \{B_2\}$$

$$\wedge SB'_1 \text{ et } SB'_2 \text{ compatibles pour le produit}$$

La preuve découle directement du lemme 4.2. Il suffit de découper les extensions temporelles des bandes temporelles $B_1 = (t_i, t_s, p_{i1}, p_{s1})$ et $B_2 = (t_i, t_s, p_{i2}, p_{s2})$ de manière à obtenir des suites dont les bandes temporelles ont pour instants de début, soit l'instant de début des bandes temporelles B_1 ou B_2 (i.e. t_i), soit une solution des équations $p_{i1}p_{i2}(t) = p_{i1}p_{s2}(t)$, $p_{i1}p_{i2}(t) = p_{s1}p_{i2}(t)$, $p_{i1}p_{i2}(t) = p_{s1}p_{s2}(t)$, $p_{i1}p_{s2}(t) = p_{s1}p_{i2}(t)$, $p_{i1}p_{s2}(t) = p_{s1}p_{s2}(t)$ et $p_{s1}p_{i2}(t) = p_{s1}p_{s2}(t)$. (Seules les égalités $p_i(t) = p_j(t)$ où $p_i \neq p_j$ sont considérées).

Il découle de ce lemme que la multiplication de deux bandes temporelles quelconques consiste à déterminer les suites de bandes temporelles compatibles pour le produit et équivalentes à ces deux bandes, puis à déterminer les produits de polynômes inférieur et supérieur. Nous présentons un algorithme permettant de réaliser ce calcul. Il consiste à :

- Collecter les solutions t_x des équations

$$p_{i1}(t) = 0, p_{i2}(t) = 0, p_{s1}(t) = 0, p_{s2}(t) = 0,^1$$

$$p_{i1}p_{s2}(t) = p_{s1}p_{i2}(t), p_{i1}p_{i2}(t) = p_{s1}p_{s2}(t)$$

- Conserver les solutions t_x de l'équation $p_{i1}p_{s2}(t) = p_{s1}p_{i2}(t)$ lorsque la valeur à l'instant t_x de ces polynômes (i.e. $p_{i1}p_{s2}(t_x) = p_{s1}p_{i2}(t_x)$) est supérieure ou inférieure aux valeurs à l'instant t_x des deux autres polynômes (i.e. $p_{i1}p_{i2}(t_x)$ et $p_{s1}p_{s2}(t_x)$).

Répéter cette opération pour les cinq autres équations.

- Créer la suite de bandes temporelles dont les instants de début et de fin sont deux solutions consécutives des 6 équations. Calculer les polynômes de la bande temporelle d'extension temporelle $[t_{x1}, t_{x2}[$ consiste à calculer les valeurs à un instant t_0 de $]t_{x1}, t_{x2}[$ des 4 produits de polynômes. Les polynômes inférieur et supérieur de la bande temporelle correspondent aux polynômes de plus petite et de plus grande valeurs.

La multiplication de deux bandes temporelles augmente donc le nombre de bandes temporelles. Cette augmentation est fonction du nombre de solutions des 6 équations. Si SB est une suite de bandes temporelles résultant de la multiplication de deux bandes temporelles $(t_i, t_s, p_{i1}, p_{s1})$ et $(t_i, t_s, p_{i2}, p_{s2})$, la taille de SB est, dans le cas le plus défavorable :

$$\text{card}(SB) = 1 + d^0 p_{i1} + d^0 p_{i2} + d^0 p_{s1} + d^0 p_{s2} + \max(d^0 p_{i1} + d^0 p_{i2}, d^0 p_{s1} + d^0 p_{s2})$$

$$+ \max(d^0 p_{i1} + d^0 p_{s2}, d^0 p_{s1} + d^0 p_{i2})$$

(Nombre de solutions des 6 équations plus 1).

Nous étendons maintenant la multiplication de deux bandes temporelles aux suites de bandes temporelles.

1. Les solutions de l'équation $p_{i1}p_{i2}(t) = p_{i1}p_{s2}(t)$ correspondent aux solutions des équations $p_{i1}(t) = 0$ et $p_{i2}(t) = p_{s2}(t)$. Les solutions de l'équation $p_{i2}(t) = p_{s2}(t)$ ne sont pas conservées car elles ne peuvent pas correspondre à un changement du polynôme inférieur ou supérieur.

Définition 4.9 : “*Multiplication de deux suites de bandes temporelles synchrones*”

La multiplication de deux suites de bandes temporelles SB_1 et SB_2 synchrones est une suite de bandes temporelles SB , notée $SB_1 *_B SB_2$, égale à l’union des suites de bandes temporelles résultant de la multiplication des bandes temporelles de même rang de SB_1 et de SB_2 (i.e $SB_1[k]$ et $SB_2[k]$).

D’une manière formelle :

$$SB_1 *_B SB_2 = \{SB_1[1] *_B SB_2[1]\} \cup \dots \cup \{SB_1[n] *_B SB_2[n]\}$$

où $n = \text{card}(SB_1)$

Par la suite, nous utiliserons indifféremment les notations $SB_1 *_B SB_2$, $SB_1 * SB_2$ ou $SB_1 SB_2$ pour la multiplication des suites SB_1 et SB_2 .

Nous nous intéressons maintenant aux propriétés de cette opération.

4.2.2 Propriétés

Nous définissons d’abord quelques propriétés générales. La multiplication de deux suites de bandes temporelles :

- est **associative**,
 $SB_1 (SB_2 SB_3) \sim (SB_1 SB_2) SB_3$
- est **commutative**,
 $SB_1 SB_2 = SB_2 SB_1$
- **possède un unique élément neutre** $\{(t_i, t_s, 1, 1)\}$ pour les suites de bandes temporelles d’extension temporelle égale à $[t_i, t_s]$.
 $SB \{(t_i, t_s, 1, 1)\} = \{(t_i, t_s, 1, 1)\} SB = SB$

Par contre, la **multiplication de deux suites de bandes temporelles ne possède pas d’inverse**. En effet, l’inverse de la bande temporelle (t_i, t_s, p_i, p_s) serait $(t_i, t_s, 1/p_i, 1/p_s)$. Or, $(t_i, t_s, 1/p_i, 1/p_s)$ n’est pas une bande temporelle car $1/p_i$ et $1/p_s$ ne sont pas des polynômes en règle générale (ceux-ci sont des polynômes si p_i et p_s sont de degré 0). De plus, comme pour la somme, $1/p_i$ est parfois supérieur à $1/p_s$.

De plus, en règle générale, **la multiplication n’est pas distributive par rapport à l’addition**.

Par exemple,

$$(t_i, t_s, 1, 2) [(t_i, t_s, -1, 1) + (t_i, t_s, 1, 2)] = (t_i, t_s, 1, 2) (t_i, t_s, 0, 3) = (t_i, t_s, 0, 6)$$

$$\text{et } (t_i, t_s, 1, 2) (t_i, t_s, -1, 1) + (t_i, t_s, 1, 2)(t_i, t_s, 1, 2) = (t_i, t_s, -2, 2) + (t_i, t_s, 1, 4) = (t_i, t_s, 1, 6).$$

Toutefois, dans certains cas, la distributivité tient :

Soient SB_1, SB_2 et SB_3 trois suites de bandes temporelles.

$$(SB_2 \geq 0 \wedge SB_3 \geq 0) \vee (SB_2 \leq 0 \wedge SB_3 \leq 0) \Rightarrow SB_1(SB_2 + SB_3) = SB_1SB_2 + SB_1SB_3$$

Par analogie avec l’arithmétique des intervalles [Moore, 66], cette propriété sera appelée la sous distributivité.

Nous exprimons maintenant les propriétés de la multiplication vis-à-vis de la propriété de représentation des grandeurs.

Propriété 4.7 : “*Conservation de la propriété de représentation*”

La multiplication de deux suites de bandes temporelles conserve la propriété de représentation, c’est à dire que, si deux fonctions temporelles $g_1(t)$ et $g_2(t)$ sont respectivement représentées par les suites de bandes temporelles SB_1 et SB_2 , alors la fonction temporelle $f(t) = g_1(t) g_2(t)$ est représentée par la suite de bandes temporelles $SB_1 SB_2$.

D’une manière formelle :

$$\forall g_1(t) \in \mathbf{G}, \forall g_2(t) \in \mathbf{G}, g_1(t) \text{ Rep } SB_1 \wedge g_2(t) \text{ Rep } SB_2 \Rightarrow g_1(t) g_2(t) \text{ Rep } (SB_1 SB_2)$$

Preuve :

Nous montrons cette propriété pour deux bandes temporelles. L’extension aux suites de bandes temporelles est immédiate.

Soient deux fonctions temporelles $g_1(t)$ et $g_2(t)$ et deux bandes temporelles $B_1 = (t_i, t_s, p_{i1}, p_{s1})$ et $B_2 = (t_i, t_s, p_{i2}, p_{s2})$.

$$g_1(t) \text{ Rep } SB_1 \wedge g_2(t) \text{ Rep } SB_2$$

$$\Leftrightarrow \forall t \in [t_i, t_s], g_1(t) \in [p_{i1}(t), p_{s1}(t)] \wedge g_2(t) \in [p_{i2}(t), p_{s2}(t)]$$

$$\Rightarrow \forall t \in [t_i, t_s], g_1(t)g_2(t) \in [\min(p_{i1}(t)p_{i2}(t), p_{s1}(t)p_{i2}(t), p_{i1}(t)p_{s2}(t), p_{s1}(t)p_{s2}(t)), \max(p_{i1}(t)p_{i2}(t), p_{s1}(t)p_{i2}(t), p_{i1}(t)p_{s2}(t), p_{s1}(t)p_{s2}(t))]$$

$$\Rightarrow g_1(t)g_2(t) \text{ Rep } B_1 B_2$$

□

La deuxième propriété de la multiplication est exprimée ci dessous :

Propriété 4.8 : “*Non introduction d’imprécision*”

La multiplication de deux suites de bandes temporelles continues n’introduit pas d’imprécision, c’est à dire que, toute fonction temporelle continue $g(t)$ représentée par la multiplication de deux suites de bandes temporelles continues $SB_1 * SB_2$, est égale au produit de deux fonctions temporelles continues $g_1(t)$ et $g_2(t)$ représentées par les suites de bandes temporelles SB_1 et SB_2 .

D’une manière formelle :

Soient SB_1, SB_2 deux suites de bandes temporelles continues.

$$\forall g(t) \in \mathbf{G}, g(t) \text{ Rep } SB_1 SB_2 \Rightarrow \exists g_1(t) \in \mathbf{G}, \exists g_2(t) \in \mathbf{G},$$

$$g_1(t) \text{ Rep } SB_1 \wedge g_2(t) \text{ Rep } SB_2 \wedge g(t) = g_1(t) g_2(t)$$

La démonstration de cette propriété est analogue à la démonstration de la propriété équivalente de l’addition de deux suites de bandes temporelles (propriété 4.2) en traitant les différents cas possibles un par un (cas qui correspondent aux quatre différents polynômes possibles pour les bandes temporelles)

Comme pour la somme, cette propriété ne s’étend pas à la multiplication de deux suites de bandes temporelles quelconques.

Nous étendons maintenant cette opération à la multiplication d'une suite de bandes temporelles par un intervalle.

4.2.3 Multiplication d'une suite de bandes temporelles par un intervalle

Le but de cette opération est de déterminer une suite de bandes temporelles représentant la fonction $c f(t)$ où c est une constante réelle appartenant à un intervalle $[a, b]$.

Définition 4.10 : “*Multiplication d'une suite de bandes temporelles par un intervalle*”

La multiplication d'une suite de bandes temporelles SB par un intervalle $[a, b]$, notée $[a, b] *_I SB$, est une suite de bandes temporelles égale à la multiplication des suites de bandes temporelles SB et $\{(t_i(SB), t_s(SB), a, b)\}$.

D'une manière formelle :

$$[a, b] *_I SB = \{(t_i(SB), t_s(SB), a, b)\} *_B SB$$

Les relations suivantes découlent directement de la propriété d'associativité de la multiplication de deux suites de bandes temporelles. Soient deux intervalles I_1 et I_2 et deux suites de bandes temporelles SB_1 et SB_2 :

- $I_1 *_I (I_2 *_B SB_1) \sim (I_1 \otimes I_2) *_B SB_1$ où \otimes est la multiplication de deux intervalles.
- $I_1 *_I (SB_1 *_B SB_2) \sim (I_1 *_I SB_1) *_B SB_2$

Comme pour la multiplication de deux suites de bandes temporelles, la multiplication d'une suite de bandes temporelles SB par un intervalle I sera notée (abusivement) $I SB$.

Deux propriétés de la multiplication de deux suites de bandes temporelles ont été définies (propriétés 4.7 et 4.8). Nous vérifions si ces propriétés restent valables pour le produit d'une suite de bandes temporelles par un intervalle :

Propriété 4.9 : “*Conservation de la propriété de représentation*”

La multiplication d'une suite de bandes temporelles par un intervalle conserve la propriété de représentation, c'est à dire que, si une fonction temporelle $g(t)$ est représentée par une suite de bandes temporelles SB et si une constante c appartient à $[a, b]$, alors la fonction temporelle $f(t) = c g(t)$ est représentée par $[a, b] SB$.

D'une manière formelle :

$$\forall g(t) \in \mathbf{G}, \forall c \in [a, b], g(t) \text{ Rep } SB \Rightarrow g(t) c \text{ Rep } (SB [a, b])$$

La preuve découle directement de la propriété équivalente sur la multiplication de deux suites de bandes temporelles et du fait que toute fonction $f(t) = c$, où c appartient à $[a, b]$, est représentée par la suite de bandes temporelles $\{(t_i(SB), t_s(SB), a, b)\}$.

Par contre, comme pour la somme d'un intervalle et d'une suites de bandes temporelles, la multiplication d'une suite de bandes temporelles par un intervalle augmente l'imprécision :

Propriété 4.10 : “Introduction d'imprécision”

La multiplication d'une suite de bandes temporelles continue SB par un intervalle [a, b] augmente l'imprécision, c'est à dire que, en général, il existe une fonction temporelle représentée par SB + [a, b] qui n'est pas le produit d'une fonction temporelle représentée par SB et d'une constante c appartenant à [a, b].

Cette propriété provient du fait que la représentation des fonctions temporelles par des suites de bandes temporelles ne permet pas de conserver les contraintes sur les fonctions temporelles (contrainte exprimant que la fonction $g(t) = c$ est constante).

Nous posons alors le problème suivant : est-il possible de définir l'opération de multiplication d'une suite de bandes temporelles par un intervalle de manière à conserver la propriété de non introduction d'imprécision. La propriété suivante répond négativement à cette question.

Propriété 4.11 : “Meilleure représentation ou semi-imprécision”

La multiplication d'une suite de bandes temporelles continue SB par un intervalle [a, b] est la meilleure représentation, c'est à dire que, pour tout couple (x, y) appartenant à [a, b] SB, il existe une fonction g(t) représentée par SB et une constante c appartenant à [a, b] telles que $cf(x) = y$.

D'une manière formelle :

Soit SB une suite de bandes temporelles continues.
 $\forall (x, y) \in [a, b] \text{ SB}, \exists g(t) \text{ Rep SB}, \exists c \in [a, b], cg(x) = y$

Comme pour l'addition d'un intervalle et d'une suite de bandes temporelles, la preuve de cette propriété découle de la propriété de non introduction d'imprécision de la multiplication de deux suites de bandes temporelles continues.

Nous allons maintenant caractériser une manière de déterminer les fonctions qui devraient être représentées par [a, b]SB. Nous exprimons d'abord la propriété de non introduction d'imprécision de la multiplication d'une suite de bandes temporelles par un intervalle dégénéré.

Propriété 4.12 : “Non introduction d'imprécision pour un intervalle dégénéré”

La multiplication d'une suite de bandes temporelles SB par un intervalle dégénéré [a, a] ($a \neq 0$) n'introduit pas d'imprécision, c'est à dire que, si une fonction temporelle g(t) est représentée par [a, a] SB, alors la fonction temporelle $g(t) / a$ est représentée par SB.

D'une manière formelle :

$\forall g(t) \in \mathbf{G}, g(t) \text{ Rep } [a, a] \text{ SB} \Rightarrow g(t) / a \text{ Rep SB}$

La preuve découle immédiatement de la définition de la propriété de représentation.

Il découle de cette propriété que, pour savoir si une fonction temporelle g(t) peut être représentée par une suite de bandes temporelles [a, b] SB, il faut et il suffit de trouver un réel c appartenant à l'intervalle [a, b] tel que la fonction temporelle $g(t) / c$ soit représentée par [c, c]

SB. De plus, trouver la constante c consiste à trouver un extremum de la division de la fonction $g(t)$ par les polynômes inférieurs des bandes temporelles de SB (ceci dépendant du signe des polynômes).

La soustraction est maintenant définie à partir de la somme et de la multiplication.

4.3 Soustraction de deux suites de bandes temporelles

Définition 4.11 : “*Soustraction de deux suites de bandes temporelles*”

La soustraction de deux suites de bandes temporelles SB_1 et SB_2 , notée $SB_1 - SB_2$, est une suite de bandes temporelles égale à la somme de SB_1 et de la multiplication de SB_2 par l'intervalle $[-1, -1]$.

D'une manière formelle :

$$SB_1 - SB_2 = SB_1 + [-1, -1] SB_2$$

Par extension, nous noterons $-SB$, la suite de bandes temporelles $[-1, -1] SB$.

Notons que la soustraction n'est pas l'inverse de la somme. En effet, en règle générale, les suites $SB = SB_1 - SB_2$ et $SB'_1 = SB + SB_2$ sont différentes.

Par exemple, soient les suites de bandes temporelles $SB_1 = \{(t_i, t_s, 1, 2)\}$ et $SB_2 = \{(t_i, t_s, 1, 2)\}$

$$SB_1 - SB_2 = \{(t_i, t_s, 1, 2)\} - \{(t_i, t_s, 1, 2)\} = \{(t_i, t_s, -1, 1)\} = SB$$

$$\text{et } SB + SB_2 = \{(t_i, t_s, -1, 1)\} + \{(t_i, t_s, 1, 2)\} = \{(t_i, t_s, 0, 3)\} = SB'_1$$

De même, la suite $SB - SB$ est différente de $\{(t_i(SB), t_s(SB), 0, 0)\}$ (ce qui est logique car SB n'a pas d'inverse pour la somme).

En fait, comme pour les intervalles, le lien entre les suites de bandes temporelles est perdue. Nous discuterons plus longuement de cette propriété dans le chapitre suivant.

Enfin, la soustraction de deux suites de bandes temporelles est la composé des opérations d'addition de deux suites de bandes temporelles et de multiplication d'une suite de bandes temporelles par l'intervalle dégénéré $[-1, 1]$. Cette opération a les mêmes propriétés que ces opérations : elle conserve la propriété de représentation et n'introduit pas d'imprécision.

Nous définissons maintenant la troisième opération primitive

4.4 Intégrale d'une suite de bandes temporelles

Le but de ce paragraphe est de définir la suite de bandes temporelles représentant la fonction temporelle $\int_{\theta}^t f(x)dx$ (θ étant un instant quelconque) à partir de la suite de bandes temporelles représentant $f(t)$. On rappelle que les fonctions $f(t)$ considérées sont continues sur \mathbf{R} , ou au moins sur $[t_i(SB), t_s(SB)[$ (SB étant la suite de bandes temporelles considérées). Elles sont donc intégrables sur cet intervalle.

4.4.1 Définitions

Cette opération est d'abord définie pour une bande temporelle.

Définition 4.12 : "Intégrale d'une bande temporelle"

L'intégrale d'une bande temporelle $B = (t_i, t_s, p_i, p_s)$ à partir d'un instant θ de $]t_i, t_s[$ est une suite de bandes temporelles continue $\{(t_i, \theta, P_s, P_i), (\theta, t_s, P_i, P_s)\}$, notée $\int_{\theta} B$, telle que

$$P_i(t) = \int_{\theta}^t p_i(x)dx \text{ et } P_s(t) = \int_{\theta}^t p_s(x)dx$$

Les propriétés de l'intégration garantissent l'existence des bandes temporelles (t_i, θ, P_s, P_i) et (θ, t_s, P_i, P_s) , i.e. P_i et P_s sont des polynômes et $\forall t \in [t_i, \theta[, P_s(t) \leq P_i(t)$ et $\forall t \in [\theta, t_s[, P_i(t) \leq P_s(t)$.

- Si le polynôme p_i de B est le n -uplet (a_0, \dots, a_n) (i.e. $p_i(t) = a_0 + a_1t + \dots + a_nt^n$), alors P_i est le polynôme $P'_i - P'_i(\theta)$ où P'_i est le polynôme $(0, a_0, a_1/2, \dots, a_n/n+1)$. (P'_i est l'intégrale indéfini de p_i et $P'_i(\theta)$ est une constante réelle égale à $a_0\theta + a_1/2\theta^2 + \dots + a_n/n+1\theta^{n+1}$).
- Comme B est une bande temporelle, on a la relation $\forall t \in [t_i, t_s[, p_i(t) \leq p_s(t)$. Les propriétés de l'intégrale (cf annexe) nous assurent que :

$$\forall t \in [t_i, t_s[, p_i(t) \leq p_s(t) \Rightarrow \forall t \in [\theta, t_s[, \int_{\theta}^t p_i(x)dx \leq \int_{\theta}^t p_s(x)dx \text{ (car } \theta \leq t)$$

$$\text{et } \forall t \in [t_i, t_s[, p_i(t) \leq p_s(t) \Rightarrow \forall t \in [t_i, \theta[, \int_{\theta}^t p_s(x)dx \leq \int_{\theta}^t p_i(x)dx$$

$$\text{(car } \int_{\theta}^t p_i(x)dx = - \int_t^{\theta} p_i(x)dx \text{ et } \theta > t).$$

Calculer l'intégrale d'une bande temporelle B consiste donc à calculer les intégrales des polynômes de B , calcul réalisé en $O(p)$ où p est le degré de B .

Un exemple d'intégrale d'une bande temporelle d'ordre 0 est illustré ci dessous.

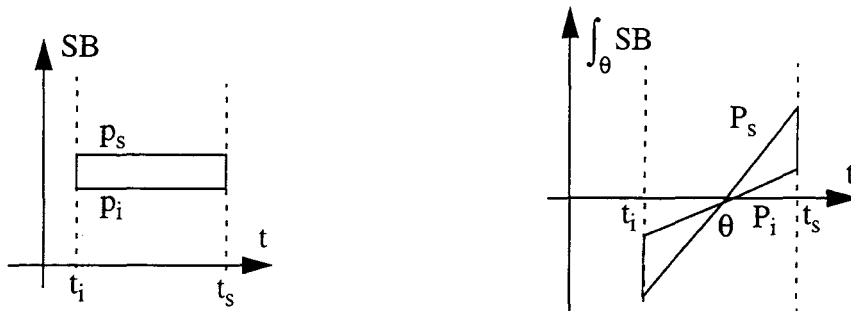


Figure 4.5 : Intégrale d'une bande temporelle

Cette définition est étendue pour les instants de début et de fin des extensions temporelles. En conservant les notations de la définition 4.12, on obtient :

$$\int_{t_i} B = \{(t_i, t_s, P_i, P_s)\} \text{ et } \int_{t_s} B = \{(t_i, t_s, P_s, P_i)\}$$

Nous étendons maintenant la définition de l'intégrale d'une bande temporelle aux suites de bandes temporelles.

Définition 4.13 : “Intégrale d’une suite de bandes temporelles”

L’intégrale à partir d’un instant quelconque θ de $]t_i(B_j), t_s(B_j)[$ ($1 \leq j \leq n$) d’une suite de bandes temporelles $SB = \{B_1, \dots, B_n\}$ de taille n et de degré p est la suite continue de bandes temporelles $\{B'_1, \dots, B'_{j-1}\} \cup \{B'_{j1}, B'_{j2}\} \cup \{B'_{j+1}, \dots, B'_n\}$ de taille $n+1$ et de degré $p+1$, notée $\int_{\theta} SB$, telle que

- $\{B'_{j1}, B'_{j2}\} = \int_{\theta} B_j$,
- $B'_{j-1} = B'_{j1}(t_i(B_j)) + \int_{t_s(B_{j-1})}^{t_i(B_j)} B_{j-1} \wedge \forall k \in [1, j-2], B'_k = B'_{k+1}(t_i(B_{k+1})) + \int_{t_s(B_k)}^{t_i(B_{k+1})} B_k$,
- $B'_{j+1} = B'_{j2}(t_s(B_j)) + \int_{t_i(B_{j+1})}^{t_s(B_j)} B_{j+1} \wedge \forall k \in [j+2, n], B'_k = B'_{k-1}(t_s(B_{k-1})) + \int_{t_i(B_k)}^{t_s(B_{k-1})} B_k$.

Cette définition est étendue aux intégrales à partir des instants de début des extensions temporelles des bandes de la suite $SB : \int_{t_i(B_j)} SB = \{B'_1, \dots, B'_{j-1}, B'_{j1}, B'_{j2}, B'_{j+1}, \dots, B'_n\}$ où $\int_{t_i(B_j)} B_j = \{B'_{j1}, B'_{j2}\}$.

On remarque que la taille de cette suite est identique à la taille de la suite SB .

On étend de même aux instants de fin des extensions temporelles des bandes de SB :

$$: \int_{t_s(B_j)} SB = \{B'_1, \dots, B'_{j-1}, B'_{j1}, B'_{j2}, B'_{j+1}, \dots, B'_n\} \text{ où } \int_{t_s(B_j)} B_j = \{B'_{j1}, B'_{j2}\}$$

On vérifie aisément que $\forall k \in [2, n], \int_{t_i(B_k)} B_j = \int_{t_s(B_{k-1})} B_j$

Le calcul de l’intégrale d’une suite de bandes temporelles SB est directement issu de la définition. Ce calcul se fait par récurrence. Pour calculer la $k^{\text{ième}}$ bande temporelle de $\int_{\theta} SB$ ($k > j$), il est nécessaire de calculer la $k-1^{\text{ième}}$ bande temporelle de $\int_{\theta} SB$. La récurrence s’arrête à la $j^{\text{ième}}$ bande temporelle de $\int_{\theta} SB$, dont la valeur est l’intégrale de $SB[j]$.

La même récurrence est réalisée pour les bandes temporelles de $\int_{\theta} SB$ de rang $k < j$.

Il en découle que le calcul de l’intégrale d’une suite de bandes temporelles de taille n consiste à calculer n intégrales de bandes temporelles et $n-1$ valeurs de ces intégrales.

La figure 4.6 illustre l’intégrale d’une suite de bandes temporelles.

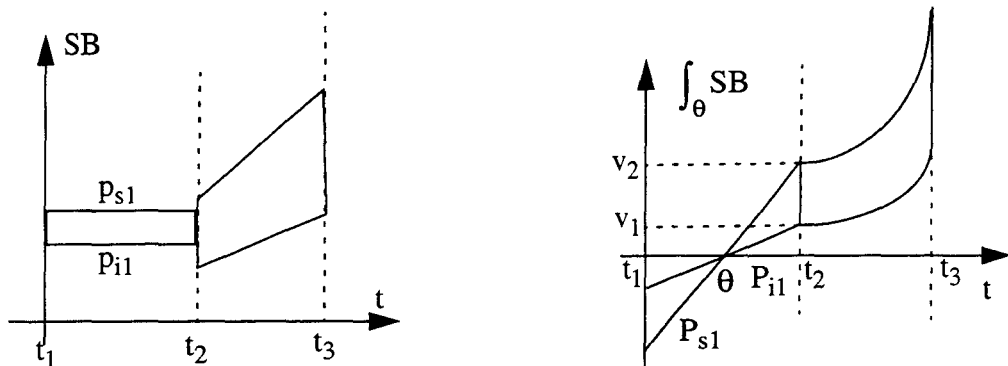


Figure 4.6 : Intégrale d’une suite de bandes temporelles

Nous noterons $\int_{\theta}^{t_0} SB$ la valeur de $\int_{\theta} SB$ à l'instant t_0 (i.e. si $\int_{\theta} SB = SB'$, alors $\int_{\theta}^{t_0} SB = SB'(t_0)$).

Par exemple, la valeur à l'instant t_2 de la suite de bandes temporelles illustrée figure 4.6 est $\int_{\theta}^{t_2} SB = [v_1, v_2]$.

Graphiquement, la valeur $\int_{\theta}^{t_0} SB$ correspond à la surface entre les polynômes de SB et l'axe des instants délimitée par les axes verticaux d'abscisses θ et t_0 (cf. figure 4.7).

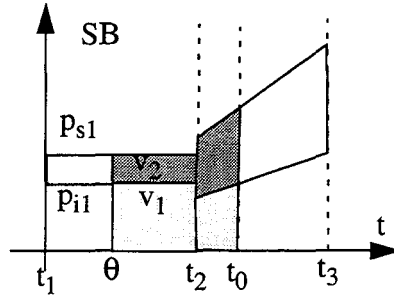


Figure 4.7 : Valeur de l'intégrale d'une bande temporelle

Avec cette notation, l'expression de la $k^{\text{ième}}$ bande temporelle de $SB' = \int_{\theta} SB$ ($k > j$) s'écrit :

$$B'_k = SB'[k] = \int_{\theta}^{t_s(B_j)} B_j \oplus \int_{t_s(B_j)}^{t_s(B_{j+1})} B_{j+1} \oplus \dots \oplus \int_{t_s(B_{k-2})}^{t_s(B_{k-1})} B_{k-1} + \int_{t_s(B_{k-1})}^{t_s(B_k)} B_k$$

(\oplus étant la somme d'intervalles)

$$= B'_{k-1}(t_s(B_{k-1})) + \int_{t_s(B_{k-1})}^{t_s(B_k)} B_k$$

La constante $B'_{k-1}(t_s(B_{k-1}))$ correspond donc à la somme des intégrales des bandes intermédiaires B_j, \dots, B_{k-1} sur leurs extensions temporelles respectives (sur $[\theta, t_s(B_j)[$ en ce qui concerne B_j).

Nous nous intéressons maintenant aux propriétés de cette opération.

4.4.2 Propriétés

Nous exprimons les propriétés de l'intégrale vis-à-vis de la représentation.

Propriété 4.13 : "Conservation de la propriété de représentation"

L'intégration d'une suite de bandes temporelles conserve la propriété de représentation, c'est-à-dire que l'intégrale de toute fonction temporelle $g(t)$, représentée par une suite de bandes temporelles SB, est représentée par l'intégrale de SB.

D'une manière formelle :

$$\forall g(t) \in \mathbf{G}, g(t) \text{ Rep SB} \Rightarrow \int_{\theta}^t g(x) dx \text{ Rep } \int_{\theta} SB$$

Preuve :

Nous montrons cette propriété pour une bande temporelle $B = (t_i, t_s, p_i, p_s)$. La démonstration s'étend naturellement aux suites de bandes temporelles.

$g(t) \text{ Rep } (t_i, t_s, p_i, p_s)$

$$\Leftrightarrow \forall t \in [t_i, t_s[, g(t) \in [p_i(t), p_s(t)]$$

$$\Rightarrow \forall t \in [\theta, t_s[, \int_{\theta}^t g(x)dx \in [\int_{\theta}^t p_i(x)dx, \int_{\theta}^t p_s(x)dx]$$

$$\text{car } g(t) \in [p_i(t), p_s(t)] \text{ et } t_2 \geq t_1 \Rightarrow \int_{t_1}^{t_2} g(x)dx \in [\int_{t_1}^{t_2} p_i(x)dx, \int_{t_1}^{t_2} p_s(x)dx]$$

$$\Rightarrow \int_{\theta}^t g(x)dx \text{ Rep } (\theta, t_s, P_i, P_s) \text{ où } P_i(t) = \int_{\theta}^t p_i(x)dx \text{ et } P_s(t) = \int_{\theta}^t p_s(x)dx$$

De plus, $g(t) \text{ Rep } (t_i, t_s, p_i, p_s)$

$$\Leftrightarrow \forall t \in [t_i, t_s[, g(t) \in [p_i(t), p_s(t)]$$

$$\Rightarrow \forall t \in [t_i, \theta[, \int_{\theta}^t g(x)dx \in [\int_{\theta}^t p_s(x)dx, \int_{\theta}^t p_i(x)dx]$$

$$\text{car } \int_{t_1}^{t_2} g(x)dx = - \int_{t_2}^{t_1} g(x)dx$$

$$\Rightarrow \int_{\theta}^t g(x)dx \text{ Rep } (t_i, \theta, P_s, P_i) \text{ où } P_i(t) = \int_{\theta}^t p_i(x)dx \text{ et } P_s(t) = \int_{\theta}^t p_s(x)dx$$

□

Il découle de cette propriété que si $g(t)$ est représentée par une suite de bandes temporelles SB, alors $\int_{\theta}^t g(x)dx$ et $\int_{\theta_1}^t g(x)dx$ sont représentées par $\int_{\theta}^t \text{SB}$ et $\int_{\theta_1}^t \text{SB}$.

De plus, la relation de Chasles nous assure que $\int_{\theta_1}^t g(x)dx = \int_{\theta_1}^{\theta} g(x)dx + \int_{\theta}^t g(x)dx$.

Donc, $\int_{\theta_1}^t g(x)dx$ est à la fois représentée par $\int_{\theta_1}^t \text{SB}$ et par $\int_{\theta}^t \text{SB} - \int_{\theta}^{\theta_1} \text{SB}$

(car $\int_{\theta_1}^{\theta} g(x)dx = - \int_{\theta}^{\theta_1} g(x)dx$ et $\int_{\theta}^{\theta_1} g(x)dx$ appartient à $\int_{\theta}^{\theta_1} \text{SB}$).

Il est donc possible d'exprimer la représentation de $\int_{\theta_1}^t g(x)dx$ en fonction de $\int_{\theta}^t \text{SB}$. Toutefois, cette opération nécessite d'ajouter un intervalle de réels (i.e. $-\int_{\theta}^{\theta_1} \text{SB}$) à une suite de bandes temporelles et va donc augmenter l'imprécision.

Nous montrons qu'il est possible de réaliser cette opération sans passer par l'addition.

Lemme 4.4 :

Soit SB une suite de bandes temporelles.

Si $\int_{\theta}^t \text{SB} = \{ (t_1, t_2, P_{i1}, P_{s1}), \dots, (t_{j-1}, t_j, P_{i,j-1}, P_{s,j-1}), (t_j, \theta, P_{s_j}, P_{i_j}), (\theta, t_{j+1}, P_{i_j}, P_{s_j}), (t_{j+1}, t_{j+2}, P_{i,j+1}, P_{s,j+1}), \dots, (t_n, t_{n+1}, P_{i_n}, P_{s_n}) \}$

alors, pour tout instant θ_1 de $]t_k, t_{k+1}[$ ($k > j$),

$$\int_{\theta_1}^t \text{SB} = \{ (t_1, t_2, P_{i1} - P_{s_k}(\theta_1), P_{s1} - P_{i_k}(\theta_1)), \dots, (t_{j-1}, t_j, P_{i,j-1} - P_{s_k}(\theta_1), P_{s,j-1} - P_{i_k}(\theta_1)),$$

$$(t_j, t_{j+1}, P_{s_j} - P_{s_k}(\theta_1), P_{i_j} - P_{i_k}(\theta_1)), \dots, (t_k, \theta_1, P_{s_k} - P_{s_k}(\theta_1), P_{i_k} - P_{i_k}(\theta_1)),$$

$$(\theta_1, t_{k+1}, P_{i_k} - P_{i_k}(\theta_1), P_{s_k} - P_{s_k}(\theta_1)), \dots, (t_n, t_{n+1}, P_{i_n} - P_{i_k}(\theta_1), P_{s_n} - P_{s_k}(\theta_1)) \}$$

La preuve est immédiate lorsqu'on calcule les suites de bandes temporelles $\int_{\theta} SB$ et $\int_{\theta_1} SB$. Nous en donnons cependant une explication intuitive sur $[\theta_1, t_{n+1}]$. On mène le même type de raisonnement sur $[\theta, \theta_1]$ et sur $[t_1, \theta]$. La figure 4.8 illustre une suite de bandes temporelles SB et les surfaces entre l'axe des abscisses et les polynômes inférieurs et supérieurs de SB. Comme $S_1 = S_3 + S_5$ et $S_2 = S_4 + S_6$, on a $\int_{\theta_1}^t SB = [S_5, S_6] = [S_1 - S_3, S_2 - S_4]$. De plus, si $\int_{\theta} SB = \{(t_1, t_2, P_{i_1}, P_{s_1}), (t_2, t_3, P_{i_2}, P_{s_2}), (t_3, t_4, P_{i_3}, P_{s_3})\}$, alors $S_1 = P_{i_3}(t_0)$, $S_2 = P_{s_3}(t_0)$, $S_3 = P_{i_2}(\theta_1)$ et $S_4 = P_{s_2}(\theta_1)$. Par conséquent, $\int_{\theta_1}^t SB = [P_{i_3}(t_0) - P_{i_2}(\theta_1), P_{s_3}(t_0) - P_{s_2}(\theta_1)]$.

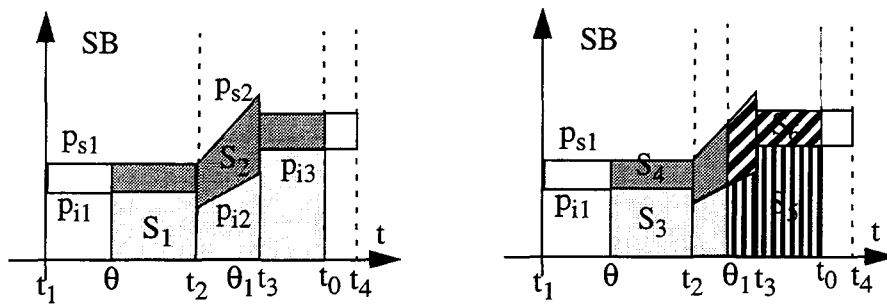


Figure 4.8 : Démonstration intuitive du lemme 4.4

Remarque :

La fonction $\int_{\theta_1}^t g(x)dx$ est aussi représentée par $\int_{\theta} SB - \int_{\theta}^{\theta_1} SB$. En conservant les notations du lemme 4.4, on a $\int_{\theta}^{\theta_1} SB = [P_{i_k}(\theta_1), P_{s_k}(\theta_1)]$ et

$$\int_{\theta} SB - \int_{\theta}^{\theta_1} SB = \{(t_1, t_2, P_{i_1} - P_{s_k}(\theta_1), P_{s_1} - P_{i_k}(\theta_1)), \dots, (t_n, t_{n+1}, P_{i_n} - P_{s_k}(\theta_1), P_{s_n} - P_{i_k}(\theta_1)) \}.$$

Cette suite de bandes temporelles est identique à $\int_{\theta_1} SB$ sur $[t_1, t_j]$ et contient cette suite sur $[t_j, t_{n+1}]$. En effet, les polynômes inférieurs (resp. supérieurs) des bandes temporelles de $\int_{\theta} SB - \int_{\theta}^{\theta_1} SB$ sont tous inférieurs (resp. supérieurs) aux polynômes inférieurs (resp. supérieurs) des bandes temporelles de $\int_{\theta_1} SB$. Soient:

- $P_{i_j} - P_{s_k}(\theta_1) \leq P_{s_j} - P_{s_k}(\theta_1)$ car $P_{i_j} \leq P_{s_j}$
- $P_{i_j} - P_{i_k}(\theta_1) \leq P_{s_j} - P_{i_k}(\theta_1)$ (idem pour $j = j+1, \dots, k$)
- $P_{i_k} - P_{s_k}(\theta_1) \leq P_{i_k} - P_{i_k}(\theta_1)$
- $P_{s_k} - P_{s_k}(\theta_1) \leq P_{s_k} - P_{i_k}(\theta_1)$ (idem pour $k = k+1, \dots, n$)

La deuxième propriété des opérations sur les suites de bandes temporelles est la non introduction d'imprécision. Nous exprimons cette propriété pour l'intégration.

Propriété 4.14 : “Introduction d'imprécision”

L'intégration d'une suite de bandes temporelles SB introduit de l'imprécision, c'est à dire que, en général, il existe une fonction temporelle continûment dérivable $G(t)$ représentée par $\int_{\theta} SB$ qui n'est pas égale à l'intégrale d'une fonction temporelle continue $g(t)$ représentée par SB.

Comme la fonction $G(t)$ est continûment dérivable, $G'(t) = g(t)$. Par conséquent, la propriété de non introduction d'imprécision peut s'exprimer : si une fonction continûment dérivable $G(t)$ est représentée par $\int_{\theta} SB$ alors sa dérivée $G'(t)$ est représentée par SB.

Nous montrons sur un exemple que cette propriété n'est pas vérifiée.

Soit la fonction $F(t)$ illustrée figure 4.9. Cette fonction est représentée par $\int_{\theta} SB$ et la dérivée de cette fonction n'est pas représentée par SB, car $F(t)$ est minimum à l'instant α_1 (i.e. $F'(\alpha_1) = 0$) et le couple $(\alpha_1, F'(\alpha_1))$ n'appartient pas à SB.

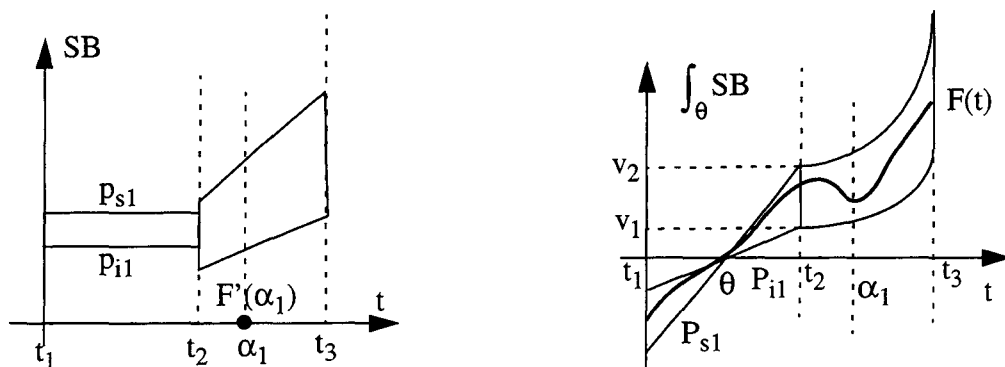


Figure 4.9 : Introduction d'imprécision de l'intégration

Il est possible de prouver que la dérivée de la fonction $F(t)$ illustrée figure 4.9 n'est pas représentée par SB sans calculer (ou déduire) sa valeur en un instant. Par exemple, la figure 4.10 illustre une fonction $F(t) - F(\theta_1)$ non représentée par l'intégrale à partir de l'instant θ_1 de la deuxième bande temporelle de SB.

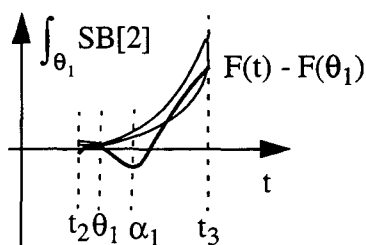


Figure 4.10 : Non représentation de la fonction $F(t) - F(\theta_1)$

Nous allons maintenant essayer de caractériser les fonctions continûment dérivables $G(t)$ représentées par l'intégrale d'une suite de bandes temporelles SB et dont les dérivées sont représentées par SB.

Il est évident que notre but n'est pas de calculer la dérivée des fonctions $G(t)$. Si on supposait connues les dérivées des fonctions $G(t)$, il suffirait alors de représenter $G'(t)$ par SB et le calcul de l'intégrale d'une suite de bandes temporelles serait inutile¹.

Le théorème suivant permet de caractériser ces fonctions lorsqu'elles sont représentées par l'intégrale d'une suite de bandes temporelles.

Théorème 4.1 : "Fonctions temporelles "représentables" par l'intégrale d'une suite de bandes temporelles"

Si pour tout instant θ appartenant à $[t_i(SB), t_s(SB)[$, l'intégrale à partir de θ d'une fonction temporelle continue $f(t)$ est représentée par l'intégrale à partir de θ d'une suite de bandes temporelles SB, alors la fonction $f(t)$ est représentée par SB.

D'une manière formelle :

$$\forall \theta \in [t_i(SB), t_s(SB)[, \int_{\theta}^t f(x)dx \text{ Rep } \int_{\theta}^t SB \Rightarrow f(t) \text{ Rep } SB$$

Preuve :

Nous montrons ce lemme pour une bande temporelle $B = (t_i, t_s, p_i, p_s)$. Nous expliquerons ensuite comment étendre cette preuve à une suite de bandes temporelles continue.

L'intégration d'une bande temporelle nous assure que :

$$\int_{t_i}^t B = \{(t_i, t_s, P_i, P_s)\} \text{ où } P_i(t) = \int_{t_i}^t p_i(x)dx \text{ et } P_s(t) = \int_{t_i}^t p_s(x)dx$$

Comme les fonctions $f(t)$, $p_i(t)$ et $p_s(t)$ sont continues, les fonctions $F(t) = \int_{t_i}^t f(x)dx$, $P_i(t)$ et $P_s(t)$ sont continues et dérivables et leurs dérivées sont $F'(t) = f(t)$, $P_i'(t) = p_i(t)$ et $P_s'(t) = p_s(t)$.

La relation de Chasles nous assure que :

$$\int_{\theta}^t f(x)dx = F(t) - F(\theta), \int_{\theta}^t p_i(x)dx = P_i(t) - P_i(\theta) \text{ et } \int_{\theta}^t p_s(x)dx = P_s(t) - P_s(\theta)$$

La propriété de représentation nous assure que :

$$\int_{\theta}^t f(x)dx \text{ Rep } \int_{\theta}^t B \Leftrightarrow \forall t \in [\theta, t_s[, \int_{\theta}^t f(x)dx = F(t) - F(\theta) \in [\int_{\theta}^t p_i(x)dx, \int_{\theta}^t p_s(x)dx]$$

D'où

$$\forall \theta \in [t_i, t_s[, \int_{\theta}^t f(x)dx \text{ Rep } \int_{\theta}^t B$$

$$\Rightarrow \forall \theta \in [t_i, t_s[, \forall t \in [\theta, t_s[, P_i(t) - P_i(\theta) \leq F(t) - F(\theta) \leq P_s(t) - P_s(\theta)$$

$$\Rightarrow \forall \theta \in [t_i, t_s[, \lim_{t \rightarrow \theta^+} \frac{P_i(t) - P_i(\theta)}{t - \theta} \leq \lim_{t \rightarrow \theta^+} \frac{F(t) - F(\theta)}{t - \theta} \leq \lim_{t \rightarrow \theta^+} \frac{P_s(t) - P_s(\theta)}{t - \theta}$$

$$\Rightarrow \forall \theta \in [t_i, t_s[, P_i'(\theta) \leq F'_d(\theta) \leq P_s'(\theta)$$

$$\Leftrightarrow \forall \theta \in [t_i, t_s[, p_i(\theta) \leq f(\theta) \leq p_s(\theta)$$

$$\Leftrightarrow f(t) \text{ Rep } (t_i, t_s, p_i, p_s) = B$$

Donc, $f(t)$ est représentée par (t_i, t_s, p_i, p_s) .

Pour étendre ce lemme aux suites de bandes temporelles, il suffit, pour un θ quelconque de $[t_i, t_s[$, de faire le même raisonnement que ci dessus en remplaçant B par $SB[k]$ où k est tel que $\theta \in [t_i(SB[k]), t_s(SB[k])[$.

□

1. En fait, la modélisation par des bandes temporelles a été introduite pour éviter de calculer les dérivées des fonctions, car ce calcul semble parfois aléatoire.

4.5 Conclusion

Trois opérations primitives sur les suites de bandes temporelles ont été définies :

- la somme,
- la multiplication,
- l'intégration.

La somme et la multiplication ont aussi été étendues aux intervalles.

Toutes ces opérations ont la propriété de conservation de la représentation. Par contre, seules la somme et la multiplication n'introduisent pas d'imprécision.

Nous généralisons maintenant les définitions des opérations sur les suites de bandes temporelles afin de pouvoir définir toutes les opérations.

Chapitre 5

Généralisation

Nous proposons dans ce chapitre de définir toutes les opérations sur les suites de bandes temporelles.

La démarche consiste à comparer les opérations d'addition et de multiplication sur les suites de bandes temporelles aux opérations équivalentes sur les intervalles [Moore, 66]. Cette comparaison permettra de déduire une analogie entre ces opérations.

En utilisant cette analogie, nous exprimons alors les définitions et propriétés des opérations polynomiales sur les suites de bandes temporelles.

Nous étendons ensuite à toutes les opérations en approximant celles ci par des polynômes.

5.1 Analogie entre les opérations sur les intervalles et les opérations sur les bandes temporelles

Soient $B_1 = (t_i, t_s, p_{i1}, p_{s1})$ et $B_2 = (t_i, t_s, p_{i2}, p_{s2})$ deux bandes temporelles.

La multiplication de deux bandes temporelles a été définie comme :

$$B_1 *_B B_2 = \{ (t, y_1 y_2) / (t, y_1) \in B_1, (t, y_2) \in B_2 \}$$

D'après la définition d'une bande temporelle, cette définition peut être réécrite :

$$\begin{aligned} B_1 *_B B_2 &= \{ (t, y_1 y_2) / t \in [t_i, t_s[, y_1 \in [p_{i1}(t), p_{s1}(t)], y_2 \in [p_{i2}(t), p_{s2}(t)] \} \\ &= \{ (t, y) / t \in [t_i, t_s[, y \in [p_{i1}(t), p_{s1}(t)] \otimes [p_{i2}(t), p_{s2}(t)] \} \\ &= \{ (t, y) / t \in [t_i, t_s[, y \in B_1(t) \otimes B_2(t) \} \end{aligned}$$

où \otimes est la multiplication de deux intervalles, définies de la manière suivante :

$$[a, b] \otimes [c, d] = \{ x_1 x_2 / x_1 \in [a, b], x_2 \in [c, d] \}$$

De même, la définition de l'addition de deux bandes temporelles est réécrite :

$$\begin{aligned} B_1 +_B B_2 &= \{ (t, y) / t \in [t_i, t_s[, y \in [p_{i1}(t), p_{s1}(t)] \oplus [p_{i2}(t), p_{s2}(t)] \} \\ &= \{ (t, y) / t \in [t_i, t_s[, y \in B_1(t) \oplus B_2(t) \} \end{aligned}$$

où \oplus est l'addition de deux intervalles :

$$\begin{aligned} [a, b] \oplus [c, d] &= \{ x_1 + x_2 / x_1 \in [a, b], x_2 \in [c, d] \} \\ &= [a + c, b + d] \end{aligned}$$

Il découle de ces notations que :

$$\text{Si } B = B_1 *_B B_2 \text{ alors } \forall t \in [t_i, t_s[, B(t) = B_1(t) \otimes B_2(t)$$

$$\text{Si } B = B_1 +_B B_2 \text{ alors } \forall t \in [t_i, t_s[, B(t) = B_1(t) \oplus B_2(t)$$

A partir de ces relations, il est facile de montrer que les opérations $+_B$ et \oplus d'une part, $*_B$ et \otimes d'autre part, ont les mêmes propriétés :

- associativité, commutativité, existence d'un élément neutre pour les quatre opérations,
- sous-distributivité de $*_B$ (resp. \otimes) par rapport à $+_B$ (resp. \oplus).

Par exemple, pour la commutativité de l'addition :

Soient les bandes temporelles $B = B_1 +_B B_2$ et $B' = B_2 +_B B_1$,

On a $\forall t \in [t_i, t_s[, B(t) = B_1(t) \oplus B_2(t) = B_2(t) \oplus B_1(t) = B'(t)$

Donc, $B = B_1 +_B B_2 = B_2 +_B B_1 = B'$.

Nous nous intéressons maintenant aux propriétés spécifiques des opérations sur les bandes temporelles. L'addition de deux bandes temporelles a les propriétés :

- de conservation de la représentation,

$$\forall f_1(t), f_2(t) \in G, f_1(t) \text{ Rep } B_1 \wedge f_2(t) \text{ Rep } B_2 \Rightarrow f_1(t) + f_2(t) \text{ Rep } B_1 +_B B_2$$

- de non introduction d'imprécision,

$$\forall f(t) \in G,$$

$$f(t) \text{ Rep } B_1 +_B B_2 \Rightarrow \exists f_1(t), f_2(t) \in G, f_1(t) \text{ Rep } B_1 \wedge f_2(t) \text{ Rep } B_2 \wedge f(t) = f_1(t) + f_2(t)$$

Comparons ces propriétés aux propriétés de l'addition d'intervalles. La définition de l'addition de deux intervalles $[a, b] \oplus [c, d] = \{x_1 + x_2 / x_1 \in [a, b], x_2 \in [c, d]\}$ nous assure que :

$$\forall x_1, x_2 \in \mathbf{R}, x_1 \in [a, b] \wedge x_2 \in [c, d] \Rightarrow x_1 + x_2 \in [a, b] \oplus [c, d]$$

et

$$\forall x \in \mathbf{R}, x \in [a, b] \oplus [c, d] \Rightarrow \exists x_1, x_2 \in \mathbf{R}, x_1 \in [a, b] \wedge x_2 \in [c, d] \wedge x = x_1 + x_2$$

Les propriétés de $+_B$ sont donc analogues aux propriétés de \oplus en remplaçant :

- **les variables réelles x_i par des fonctions temporelles $f_i(t)$,**
- **les intervalles I_i par des bandes temporelles B_i (ou par des suites de bandes temporelles continues SB_i),**
- **la relation d'appartenance \in par la relation de représentation Rep.**

Par contre, cette analogie ne permet pas de prendre en compte les opérations mixtes entre bandes temporelles et intervalles. Par exemple, l'introduction d'imprécision engendrée par l'addition d'un intervalle à une bande temporelle ne peut pas être exprimée à partir des opérations sur les intervalles. En effet, on a :

$$\forall f(t) \in G, f(t) \text{ Rep } B +_I [a, b] \not\Rightarrow \exists g(t) \in G, \exists c \in \mathbf{R}, g(t) \text{ Rep } B \wedge c \in [a, b] \wedge f(t) = g(t) + c$$

alors que pour l'addition d'intervalles, on a la propriété :

$$\forall x \in \mathbf{R}, x \in I \oplus [a, b] \Rightarrow \exists y \in \mathbf{R}, \exists c \in \mathbf{R}, y \in I \wedge c \in [a, b] \wedge x = y + c$$

Ceci est logique car l'imprécision introduite est due au fait que le réel c est équivalent à une fonction constante au cours du temps, or il n'existe pas de notion de temps dans l'arithmétique des intervalles.

Les propriétés liées aux intervalles ne peuvent pas être exprimées. Par contre, les opérations peuvent s'écrire

$$I +_1 B_2 = \{(t, y) / t \in [t_1, t_2[, y \in I \oplus B_2(t)\}$$

La même analogie peut être réalisée pour la multiplication.

Nous utilisons maintenant cette analogie pour définir les opérations polynomiales sur les suites de bandes temporelles.

5.2 Extension aux suites de bandes temporelles des opérations réelles

La définition de l'extension d'une opération réelle aux suites de bandes temporelles, exprimée ci dessous, est issue de la définition de l'extension des opérations réelles aux intervalles : L'extension d'une opération réelle aux intervalles est obtenue en remplaçant les variables réelles par des variables intervalles et les opérations réelles par des opérations intervalles.

Définition 5.1 : "Extension d'une opération réelle aux suites de bandes temporelles"

L'extension de l'opération réelle op aux suites de bandes temporelles, notée OP_B , est l'opération sur les suites de bandes temporelles obtenue en remplaçant :

- les variables réelles par des suites de bandes temporelles,
- les opérations réelles par les opérations équivalentes sur les suites de bandes temporelles,
- les constantes réelles par des intervalles.

Définir l'extension d'une opération réelle aux suites de bandes temporelles nécessite donc d'avoir défini au préalable quelques opérations sur les suites de bandes temporelles. Comme l'addition et la multiplication de suites de bandes temporelles sont définies, cette définition va (entre autres) permettre d'étendre aux suites de bandes temporelles toutes les opérations polynomiales réelles.

Par exemple, l'opération réelle $ca(x) = x^2 = x * x$

s'étend aux intervalles en $CA(I) = I \otimes I$

et s'étend aux suites de bandes temporelles en $CA_B(SB) = SB *_B SB$

De même, l'opération réelle $so(x_1, x_2) = x_1 + x_2$

s'étend aux suites de bandes temporelles en $SO_B(SB_1, SB_2) = SB_1 +_B SB_2$

Nous noterons les opérations réelles $op(x_1, \dots, x_n ; c_1, \dots, c_p)$ où x_1, \dots, x_n sont des variables et c_1, \dots, c_p sont des constantes et leurs extensions $OP_B(SB_1, \dots, SB_n ; I_1, \dots, I_p)$

Comme pour les opérations d'addition et de multiplication, l'extension aux suites de bandes temporelles OP_B de l'opération réelle op peut être écrite à partir de l'extension aux intervalles OP de op :

$$OP_B(SB_1, \dots, SB_n; I_1, \dots, I_p) = \{(x, y) / x \in [t_i, t_s[, y \in OP(SB_1(x), \dots, SB_n(x); I_1, \dots, I_p) \}$$

Par exemple, $CA_B(SB) = SB *_B SB = \{(x, y) / x \in [t_i, t_s[, y \in B(x) \otimes B(x)\}$

Remarque:

OP_B n'est pas, en général, une opération interne aux suites de bandes temporelles, i.e. $OP_B(SB_1, \dots, SB_n; I_1, \dots, I_p)$ n'est pas, en général, une suite de bandes temporelles.

Nous nous intéressons maintenant aux propriétés de ces opérations.

L'intérêt des méthodes de calculs sur les intervalles est qu'elles garantissent l'encadrement du résultat correct. Cette garantie provient de la propriété fondamentale de l'arithmétique des intervalles :

Soient I_1, \dots, I_n des intervalles, soit op une opération réelle, et OP l'extension de op aux intervalles. On a :

$$x_1 \in I_1 \wedge \dots \wedge x_n \in I_n \Rightarrow op(x_1, \dots, x_n; c_1, \dots, c_p) \in OP(I_1, \dots, I_n, C_1, \dots, C_p)$$

En remplaçant les x_i par des fonctions temporelles $g_i(t)$, les intervalles I_i par des suites de bandes temporelles SB_i et la relation \in par la relation Rep , on obtient :

$$g_1(t) Rep SB_1 \wedge \dots \wedge g_n(t) Rep SB_n \Rightarrow op(g_1(t), \dots, g_n(t)) Rep OP_B(SB_1, \dots, SB_n)$$

Par analogie, l'intérêt des méthodes de calcul sur les suites de bandes temporelles est qu'elles conservent la propriété de représentation.

En remarquant que si une constante c appartient à un intervalle $[a, b]$ alors la fonction $c(t) = c$ est représentée par la SBT (t_i, t_s, a, b) , on déduit que cette propriété s'étend aussi aux opérations invoquant des constantes.

Propriété 5.1 : *“Conservation de la propriété de représentation pour les extensions aux bandes temporelles des opérations réelles”*

L'extension aux suites de bandes temporelles d'une opération réelle op conserve la propriété de représentation, c'est-à-dire que, si n fonctions temporelles $g_1(t), \dots, g_n(t)$ sont représentées par n suites de bandes temporelles SB_1, \dots, SB_n et p constantes c_1, \dots, c_p appartiennent à p intervalles I_1, \dots, I_p alors $op(g_1(t), \dots, g_n(t); c_1, \dots, c_p)$ est représentée par $OP_B(SB_1, \dots, SB_n; I_1, \dots, I_p)$.

D'une manière formelle :

Soit op une opération réelle, et OP_B son extension aux suites de bandes temporelles.

$$\begin{aligned} &\forall g_1(t), \dots, g_n(t) \in \mathbf{G}, \forall c_1, \dots, c_p \in \mathbf{R}, \\ &g_1(t) Rep SB_1 \wedge \dots \wedge g_n(t) Rep SB_n \wedge c_1 \in I_1 \wedge \dots \wedge c_p \in I_p \\ &\Rightarrow op(g_1(t), \dots, g_n(t); c_1, \dots, c_p) Rep OP_B(SB_1, \dots, SB_n; I_1, \dots, I_p) \end{aligned}$$

La preuve découle immédiatement de la propriété fondamentale de l'arithmétique des intervalles.

Nous noterons $W(\text{op} ; I_1, \dots, I_n) = \{\text{op}(x_1, \dots, x_n) / \forall i = 1 \dots n, x_i \in I_i\}$ le domaine de variation de l'opération réelle op sur les intervalles I_1, \dots, I_n . (Cet ensemble est un intervalle si op est continue).

Avec cette notation, la propriété fondamentale de l'arithmétique des intervalles peut aussi s'écrire :

$$W(\text{op} ; I_1, \dots, I_n) \subseteq OP(I_1, \dots, I_n)$$

où OP est l'extension aux intervalles de op .

En règle générale, l'inverse de cette propriété n'est pas vrai (ce que Lhomme [Lhomme, 94] appelle la catastrophe fondamentale des intervalles) :

$$\text{en règle générale, } OP(I_1, \dots, I_n) \neq W(\text{op} ; I_1, \dots, I_n)$$

En d'autres termes,

en règle générale, il existe un réel y appartenant à $OP(I_1, \dots, I_n)$ tel que, pour tout réel x_i appartenant à l'intervalle I_i , on a $\text{op}(x_1, \dots, x_n) \neq y$.

Par analogie,

en règle générale, il existe une fonction temporelle $f(t)$ représentée par $OP_B(SB_1, \dots, SB_n)$ telle que, pour toute fonction temporelle $f_i(t)$ représentée par SB_i , on a $\text{op}(f_1(t), \dots, f_n(t)) \neq f(t)$.

En d'autres termes, **en règle générale, les extensions aux suites de bandes temporelles des opérations réelles introduisent de l'imprécision.**

Par exemple, le calcul sur les suites de bandes temporelles nous assure que

$$CA_B(\{(t_i, t_s, -1, 2)\}) = \{(t_i, t_s, -1, 2)\} *_B \{(t_i, t_s, -1, 2)\} = \{(t_i, t_s, -2, 4)\}$$

et

$$g(t) = -2 \neq f(t) * f(t) \text{ (pour toute fonction temporelle } f(t)).$$

Toutefois, il est possible de caractériser certaines opérations op pour lesquelles la catastrophe fondamentale ne se produit pas. (i.e. $W(\text{op} ; I_1, \dots, I_n) = OP(I_1, \dots, I_n)$)

La propriété suivante montre l'importance de ces opérations :

Propriété 5.2 : *“opérations réelles dont l'extension n'introduit pas d'imprécision”*

Soit op une opération réelle à n variables, OP son extension aux intervalles et OP_B son extension aux suites de bandes temporelles.

Si $OP(I_1, \dots, I_n) = W(\text{op} ; I_1, \dots, I_n)$ alors $OP_B(SB_1, \dots, SB_n)$ n'introduit pas d'imprécision, c'est à dire que, si une fonction temporelle $f(t)$ est représentée par $OP_B(SB_1, \dots, SB_n)$, alors il existe n fonctions temporelles $f_1(t), \dots, f_n(t)$ respectivement représentées par SB_1, \dots, SB_n telles que $f(t) = \text{op}(f_1(t), \dots, f_n(t))$.

La preuve est immédiate.

La validité de cette propriété est directement liée au fait qu'aucune constante n'a été remplacée

par un intervalle dans l'expression de la fonction.

Nous proposons maintenant de caractériser les opérations dont les extensions aux SBT n'introduisent pas d'imprécision.

Dans l'arithmétique des intervalles, quelques opérations n'engendrant pas la catastrophe fondamentale ont été définies : c'est la propriété sur les occurrences simples.

Soit $op(x_1, \dots, x_n)$ une opération réelle et $OP(I_1, \dots, I_n)$ son extension aux intervalles. Si toutes les variables n'apparaissent qu'une seule fois dans l'expression de op^1 , alors $OP(I_1, \dots, I_n) = W(op ; I_1, \dots, I_n)$

La propriété suivante est déduite par analogie.

Propriété 5.3 : “sur les occurrences simples”

Soit $op(x_1, \dots, x_n)$ une opération réelle et $OP_B(SB_1, \dots, SB_n)$ son extension aux SBT. Si toutes les variables n'apparaissent qu'une seule fois dans l'expression de op , alors OP_B n'introduit pas d'imprécision.

Par exemple, l'extension aux SBT de $op(x_1, x_2, x_3) = x_1 * x_2 + x_3$ n'introduit pas d'imprécision.

Par contre, l'extension aux SBT de l'opération $ca(x) = x^2 = x * x$ introduit de l'imprécision.

Pour augmenter le nombre d'opérations n'introduisant pas d'imprécision, une solution consiste à définir les extensions aux intervalles des opérations continues comme étant égales à leurs domaines de variation (ce que Lhomme appelle les opérations de base). Pour cela, on utilise le fait que pour les opérations monotones, le calcul sur intervalle se limite à un calcul sur les bornes des intervalles.

Par exemple, l'opération $ca(x) = x^2$ est monotone sur \mathbf{R}^+ et \mathbf{R}^- . Par conséquent,

- si $0 \notin [a, b]$ alors $[a, b]^2 = [\min(a^2, b^2), \max(a^2, b^2)]$,
- si $0 \in [a, b]$ alors $[a, b]^2 = [0, \max(a^2, b^2)]$.

C'est cette propriété qui sera utilisée pour définir les opérations polynomiales sur les SBT n'introduisant pas d'imprécision.

Nous introduisons d'abord la notion d'extension primitive aux SBT.

Définition 5.2 : “Extension primitive aux bandes temporelles des opérations réelles”

Soit $op(x_1, \dots, x_n)$ une opération réelle continue.

L'extension primitive de cette opération aux bandes temporelles, notée OP_{PB} , est une opération sur les bandes temporelles $B_1 = (t_i, t_s, p_{i1}, p_{s1}), \dots, B_n = (t_i, t_s, p_{in}, p_{sn})$ telle que

$$OP_{PB}(B_1, \dots, B_n) = \{(x, y) / x \in [t_i, t_s], y \in W(op ; [p_{i1}(x), p_{s1}(x)], \dots, [p_{in}(x), p_{sn}(x)])\}$$

1. On considère que la variable x apparaît deux fois dans l'expression $x^2 = x * x$.

Dans le cas où op est une opération unaire :

Le calcul de $W(op ; [p_{i1}(x), p_{s1}(x)])$ nous assure que, si op est croissante sur l'intervalle $[p_{i1}(x), p_{s1}(x)]$ alors $W(op ; [p_{i1}(x), p_{s1}(x)]) = [op(p_{i1}(x)), op(p_{s1}(x))]$.

De plus, $op(p_{i1})$ et $op(p_{s1})$ sont des polynômes lorsque op est une opération polynomiale. Il découle que l'extension primitive aux suites de bandes temporelles d'une opération polynomiale peut être interne lorsque op est une opération polynomiale.

Plus généralement,

si $\frac{\partial op}{\partial x_1}(x_1, \dots, x_n) \geq 0$ sur $[p_{i1}(x), p_{s1}(x)] \times \dots \times [p_{in}(x), p_{sn}(x)]$

alors $W(op; [p_{i1}(x), p_{s1}(x)], \dots, [p_{in}(x), p_{sn}(x)]) =$

$[W(op; [p_{i1}(x), p_{i1}(x)], \dots, [p_{in}(x), p_{sn}(x)])^-, (i.e. x_1 = p_{i1}(x))$

$W(op; [p_{i1}(x), p_{i1}(x)], \dots, [p_{in}(x), p_{sn}(x)])^+], (i.e. x_1 = p_{s1}(x))$

où $-$ et $+$ désignent les bornes inf et sup de l'intervalle.

Ainsi, si les dérivées partielles sont toutes de signe constant, on a:

$W(op; [p_{i1}(x), p_{i1}(x)], \dots, [p_{in}(x), p_{sn}(x)])^- = op(p_{k1}, \dots, p_{kn})$

et $W(op; [p_{i1}(x), p_{i1}(x)], \dots, [p_{in}(x), p_{sn}(x)])^+ = op(p_{k1}, \dots, p_{kn})$

avec $\forall j, p_{kj} = p_{ij} \vee p_{kj} = p_{sj}$

De plus, si op est polynomiale, alors $op(p_{k1}, \dots, p_{kn})$ est un polynôme.

Nous nous intéressons donc à ces opérations.

5.3 Opérations polynomiales sur les bandes temporelles

En règle générale, l'extension primitive aux bandes temporelles d'une opération polynomiale n'est pas une bande temporelle. Par exemple, l'extension primitive de l'opération $op(x) = x^2$ pour la bande temporelle $(0, 2, p_i, p_s)$ ($p_i(t) = t-1$ et $p_s(t) = t$) n'est pas une bande temporelle (cf.

figure 5.1)..

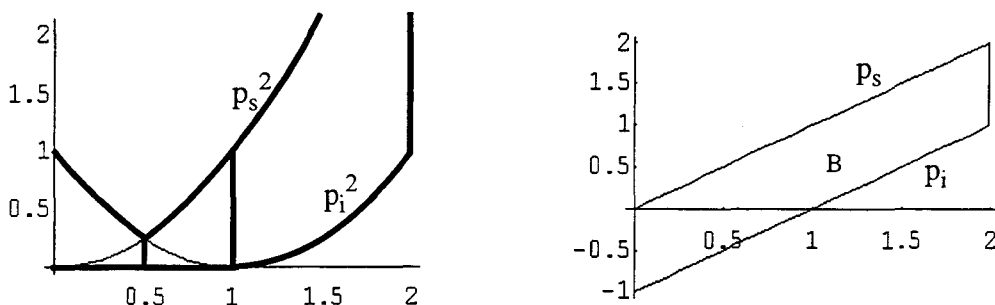


Figure 5.1 : Extension primitive de l'opération réelle $op(x) = x^2$

Nous traitons uniquement ici le cas d'opérations unaires.

Comme pour la multiplication, il est possible de déterminer des bandes temporelles pour lesquelles l'extension primitive d'une opération polynomiale unaire est interne, puis de montrer que toute suite de bandes temporelles est décomposable en une suite de bandes temporelles ayant cette propriété.

Nous nous limiterons pour les opérations polynomiales unaires à décrire un algorithme permettant de calculer ces suites de bandes temporelles¹.

Soit op une opération polynomiale réelle unaire, soient $S_m = \{m_1, \dots, m_n\}$ et $S_M = \{M_1, \dots, M_p\}$ les ensembles de minimum et maximum de op (i.e. $op'(m_i) = op'(M_i) = 0$).

Le calcul de l'extension primitive aux suites de bandes temporelles de l'opération op pour la bande temporelle (t_i, t_s, p_i, p_s) consiste à :

- a Collecter les solutions t_x des équations
- $$op(p_i(t)) = op(p_s(t)),$$
- $$p_i(t) = m_i, p_s(t) = m_i \quad (i = 1 \dots n)$$
- $$p_i(t) = M_i, p_s(t) = M_i \quad (i = 1 \dots p)$$

(pour les équations non universellement satisfaites).

- b Créer la suite de bandes temporelles dont les instants de début et de fin sont deux solutions consécutives des équations. Calculer les polynômes p_i' et p_s' de la bande temporelle d'extension temporelle $[t_{x1}, t_{x2}[$ consiste à

- calculer $p_i(t_0)$ et $p_s(t_0)$ pour un instant t_0 de $]t_{x1}, t_{x2}[$,
- chercher les minima m_i appartenant à $[p_i(t_0), p_s(t_0)]$,
si des minima m_i appartiennent à $[p_i(t_0), p_s(t_0)]$
et $op(p_i(t_0)) \geq \min_i op(m_i)$
et $op(p_s(t_0)) \geq \min_i op(m_i)$,
alors $p_i' = \min_i (m_i)$,

sinon p_i' est égal au polynôme $op(p_i)$ ou $op(p_s)$ dont la valeur à l'instant t_0 est la plus petite (i.e. $op(p_i(t_0))$ et $op(p_s(t_0))$),

si des maxima M_i appartiennent à $[p_i(t_0), p_s(t_0)]$
et $op(p_s(t_0)) \leq \max_i op(M_i)$
et $op(p_i(t_0)) \leq \max_i op(M_i)$,
alors $p_i' = \max_i (M_i)$,

sinon p_i' est égal au polynôme $op(p_i)$ ou $op(p_s)$ dont la valeur à l'instant t_0 est la plus grande (i.e. $op(p_i(t_0))$ et $op(p_s(t_0))$),

Par exemple, pour l'extension primitive de l'opération réelle $op(x) = x^2$ illustrée figure 5.1, on a :

1. Les analogies entre cette algorithme et celui permettant de calculer la multiplication de deux bandes temporelles sont évidentes et découlent de la remarque précédente.

- (a) $S_m = \{0\}$, $S_M = \{\}$.
- (b) Pour la bande temporelle $(0, 2, p_i, p_s)$, le découpage de l'extension temporelle donne la suite de bandes temporelles $\{(0, 0.5, p_{i1}, p_{s1}), (0.5, 1, p_{i2}, p_{s2}), (1, 2, p_{i3}, p_{s3})\}$ car $p_i^2(0.5) = p_s^2(0.5)$, $p_i(1) = 0$.

Le calcul des polynômes donne

$$p_{i1} = 0 \text{ car } 0 \in [p_i(0.25), p_s(0.25)] \text{ et } 0 < p_i^2(0.25) \text{ et } 0 < p_s^2(0.25)$$

$$p_{s1} = p_i^2 \text{ car } p_i^2(0.25) > p_s^2(0.25)$$

$$p_{i2} = 0 \text{ car } 0 \in [p_i(0.75), p_s(0.75)] \text{ et } 0 < p_i^2(0.25) \text{ et } 0 < p_s^2(0.25)$$

$$p_{s2} = p_s^2 \text{ car } p_i^2(0.75) < p_s^2(0.75)$$

$$p_{i3} = p_i^2 \text{ et } p_{s3} = p_s^2 \text{ car } p_i^2(1.5) < p_s^2(1.5)$$

Cet algorithme peut être étendue aux opérations n-aires. Il faut alors déterminer les intervalles temporels où les dérivées partielles sont de signe constant.

5.4 Extension primitive des opérations non polynomiales

En règle générale, le calcul de l'extension primitive aux suites de bandes temporelles d'une opération réelle non polynomiale op n'est pas interne. Pour rendre cette opération interne, nous allons approximer l'opération op par un polynôme à l'aide de la formule de Taylor. Ce traitement est réalisé uniquement pour les opérations unaires.

Définition 5.3 : “*Extension primitive interne d'ordre n aux suites de bandes temporelles des opérations réelles unaires non polynomiales*”

Soit op une opération unaire non polynomiale, soit $B = (t_i, t_s, p_i, p_s)$ une bande temporelle telle que un des polynômes op_i et op_s , définis ci-dessous, soit toujours inférieur à l'autre.

$$op_i(t) = op \circ p_i(t_i) + (t - t_i) (op \circ p_i)'(t_i) + \dots + (t - t_i)^n/n! (op \circ p_i)^{(n)}(t_i) \\ + (t - t_i)^{n+1}/(n+1)! \min_{x \in [t_i, t_s]} ((op \circ p_i)^{(n+1)}(x))$$

$$op_s(t) = op \circ p_s(t_i) + (t - t_i) (op \circ p_s)'(t_i) + \dots + (t - t_i)^n/n! (op \circ p_s)^{(n)}(t_i) \\ + (t - t_i)^{n+1}/(n+1)! \max_{x \in [t_i, t_s]} ((op \circ p_s)^{(n+1)}(x))$$

L'extension primitive interne d'ordre n aux suites de bandes temporelles de l'opération réelle op est l'opération qui associe à la bande temporelle B , la bande temporelle (t_i, t_s, p_i', p_s') telle que :

- si un ou plusieurs minima m_i de op appartiennent toujours aux intervalles $[p_i(t_0), p_s(t_0)]$ ($t_0 \in [t_i, t_s[$) et que $\min_i op(m_i) < \min(op(p_i(t_0)), op(p_s(t_0)))$,
alors $p_i'(t) = \min_i op(m_i)$
sinon $p_i'(t) = \inf(op_i, op_s)$
- si un ou plusieurs maxima M_i de op appartiennent toujours aux intervalles $[p_i(t_0), p_s(t_0)]$ ($t_0 \in [t_i, t_s[$) et que $\max_i op(M_i) > \max(op(p_i(t_0)), op(p_s(t_0)))$,
alors $p_s'(t) = \max_i op(M_i)$
sinon $p_s'(t) = \sup(op_i, op_s)$

Cette définition est donnée pour des bandes temporelles respectant certaines contraintes. Comme pour une opération polynomiale sur une bande temporelle, il est toujours possible d'associer à une bande temporelle une suite dont toutes les bandes temporelles vérifient ces contraintes.

L'extension primitive interne d'une opération réelle op introduit de l'imprécision. Nous la quantifions de la manière suivante :

Si l'extension primitive associe la bande temporelle (t_i, t_s, p_i, p_s) et l'extension primitive interne la bande temporelle (t_i, t_s, p_i', p_s') , alors l'imprécision introduite à l'instant t_0 est :

$$I(t_0) = 1 - (p_i(t_0) - p_s(t_0)) / (p_i'(t_0) - p_s'(t_0))$$

Par exemple, la figure 5.2 illustre l'extension primitive et l'extension primitive interne d'ordre 1 de l'opération $op(x) = \exp(x)$ pour la bande temporelle $(0, 1, p_i, p_s)$ ($p_i(t) = t$ et $p_s(t) = 2t + 1$)

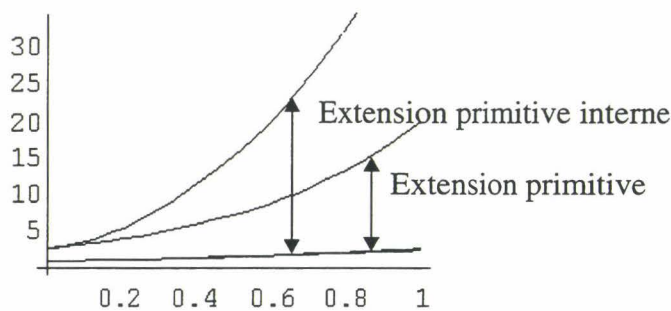


Figure 5.2 : Extension primitive d'une opération

L'imprécision introduite à l'instant $t_0 = 1$ est de l'ordre de 25%.

Pour diminuer l'imprécision introduite, deux solutions sont possibles :

- augmenter l'ordre de l'extension primitive interne (i.e. augmenter le degré des polynômes de la bande temporelle résultant de cette opération),
- découper l'extension temporelle de la bande temporelle.

Par exemple, la figure 5.3 illustre l'extension primitive et l'extension primitive d'ordre 1 de l'opération $op(x) = \exp(x)$ pour les bandes temporelles $(0, 0.5, p_i, p_s)$ et $(0.5, 1, p_i, p_s)$ (bande temporelle de la figure 5.2 découpée en une suite de bandes temporelles de taille 2)

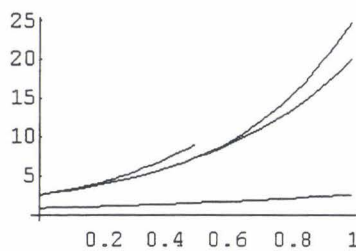


Figure 5.3 : Extension primitive d'une opération

L'imprécision introduite aux instants $t_0 = 0.5$ et $t_0 = 1$ est de l'ordre de 5%.

Multiplier par 2 le nombre de bandes temporelles a donc permis de diviser par 5 l'imprécision introduite.

Conclusion :

La notion de suites de bandes temporelles a été introduite pour modéliser des fonctions temporelles continues. La précision de la représentation d'une fonction dépend du nombre et du degré des bandes temporelles de la suite représentant cette fonction.

Les opérations sur les bandes temporelles ont été définies afin de représenter des fonctions temporelles résultant d'opérations invoquant une ou plusieurs fonctions temporelles. Ces opérations garantissent toujours que la représentation est correcte. Cette garantie provient du fait que toutes les opérations sur les suites de bandes temporelles conservent la propriété de représentation.

Trois opérations ont été particulièrement étudiées : l'addition, la multiplication et l'intégration. Les deux premières ont une propriété très intéressante : la non introduction d'imprécision ; par exemple, pour la multiplication, cette propriété permet de garantir que toute fonction représentée par le produit de deux suites de bandes temporelles $SB_1 SB_2$ est nécessairement le produit de deux fonctions représentées par SB_1 et SB_2 .

Toutefois, seules ces opérations ont cette propriété. Ainsi, en définissant l'addition et la multiplication d'une suite de bandes temporelles par un intervalle à partir de l'addition et de la multiplication de deux suites de bandes temporelles, la propriété de non introduction d'imprécision est perdue. De même, l'intégration d'une suite de bandes temporelles ne possède pas cette propriété. Par contre, il est possible de savoir si une fonction temporelle est réellement représentable par une suite de bandes temporelles résultant de ces opérations.

Enfin, toutes les opérations peuvent être étendue aux suites de bandes temporelles à partir des opérations d'addition et de multiplication. Ces extensions sont définies par analogie avec les extensions aux intervalles des opérations réelles. La propriété fondamentale de ces opérations est qu'elles conservent la propriété de représentation. Par contre, pour conserver la propriété de non introduction d'imprécision, il est indispensable de redéfinir ces opérations. Toutefois, ceci semble augmenter de manière significative le nombre de bandes temporelles des suites.

PARTIE II

Consistance des Modèles

Introduction

Le principe du diagnostic à base de modèles est de vérifier la cohérence des observations d'un système et du comportement théorique de ce système. Typiquement, le comportement réel d'un système est caractérisé par les grandeurs de ce système. Les observations sont les informations obtenues sur les grandeurs à partir d'une procédure de mesure. Le comportement théorique d'un système est un ensemble de relations liant les grandeurs du système, chaque relation décrivant le comportement d'un composant ou d'un bloc de composants.

Cette partie est consacrée à l'étude d'une procédure de comparaison des observations et du comportement théorique d'un système, lorsque les observations sont modélisées par des suites de bandes temporelles.

Cette procédure, permettant de détecter les ensembles de conflits, est le coeur des systèmes de diagnostic. Elle représente une contrainte très forte liant la modélisation des observations et le formalisme de description du comportement des systèmes. En effet, ces données sont modélisées de manière à pouvoir vérifier leur cohérence. Une modélisation simple des observations permet d'obtenir une procédure de comparaison simple, mais limite le formalisme de description des systèmes. A notre connaissance, tous les systèmes de diagnostic déjà réalisés sont basés sur une procédure de comparaison simple et immédiate. Ainsi, la procédure des systèmes basés sur une modélisation qualitative des grandeurs consiste à vérifier que le signe et la variation de deux grandeurs égales sont égaux et celle basée sur une modélisation quantitative consiste à vérifier que les intervalles encadrant deux grandeurs égales s'intersectent. Toutefois, ces systèmes requièrent les observations des variations des grandeurs, observations difficiles à obtenir.

Nous allons montrer dans cette partie que modéliser les observations par des suites de bandes temporelles permet de surmonter ces problèmes. Nous proposons de décrire le comportement d'un système par un ensemble d'équations différentielles de la forme $f(t) + g'(t) + h''(t) + \dots = 0$. Le principal résultat est qu'il existe une procédure permettant de vérifier la cohérence du comportement d'un système et des observations, lorsque le comportement du système est décrit par une équation différentielle de la forme précédemment proposée et les observations sont les SBT représentant les fonctions $f(t)$, $g(t)$, $h(t)$, ...

Dans le premier chapitre, nous proposons un formalisme de description du comportement des systèmes.

Le deuxième chapitre est consacré à la formalisation de la procédure de comparaison. Afin de définir la complétude de cette procédure, nous introduisons les notions d'inconsistance et de C-inconsistance d'un modèle pour un ensemble de SBT.

Une famille d'algorithmes de détection d'inconsistance des modèles pour un ensemble de SBT est présentée dans le troisième chapitre.

Chapitre 6

Description du système

Dans la théorie du diagnostic à partir des principes premiers [Reiter, 87], un système est décrit par un triplet (SD, COMP, OBS) où

- SD (System Description) est la description du comportement théorique du système,
- COMP est l'ensemble des composants,
- OBS est l'ensemble des observations.

Nous proposons dans ce chapitre de définir ces ensembles.

L'ensemble des observations est un ensemble composé de SBT et d'intervalles. Cette description permet de différencier les grandeurs constantes au cours du temps de celles qui ne le sont pas.

Le comportement d'un ensemble de composants est décrit par une équation différentielle et une fenêtre temporelle. L'avantage de cette modélisation est de permettre la description du comportement de bon nombre de composants directement à partir des modèles physiques de ces composants. Toutefois, la modélisation des composants en elle même n'a pas été étudiée.

Les exemples de modèles de composants¹ que nous présentons sont issus des travaux sur CATS [Dague et al, 91].

6.1 Les observations

Le comportement réel d'un système est décrit par les différentes grandeurs de ce système. Par exemple, les grandeurs du système illustré figure 6.1 sont les tensions et les courants, c'est-à-dire les grandeurs $v_g(t)$, $v_{r_1}(t)$, $v_{r_2}(t)$, $i_g(t)$, $i_{r_1}(t)$, $i_{r_2}(t)$. Les paramètres des composants seront aussi considérés comme des grandeurs. Ainsi, les valeurs des résistances r_1 et r_2 sont les grandeurs $r_1(t)$ et $r_2(t)$.

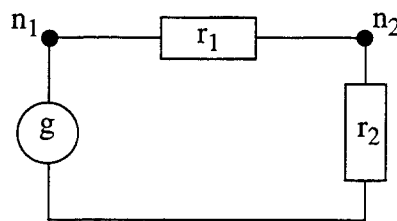


Figure 6.1 : Exemple de système

Le modèle mathématique d'une grandeur est une fonction continue de \mathbf{R} dans \mathbf{R} (où \mathbf{R} est l'ensemble des réels). Nous noterons \mathbf{F} l'ensemble des fonctions continues de \mathbf{R} dans \mathbf{R} et $\mathbf{F}(i)$ l'ensemble des fonctions i fois continûment dérivables.

1. Les composants dont les modèles sont présentés dans ce chapitre sont tous issus du domaine de l'électronique analogique. Toutefois, le formalisme proposé permet d'envisager la modélisation de composants issus d'autres domaines telle que la modélisation d'une usine chimique [Alarcon et al, 94].

Toute composée de fonctions temporelles est une fonction temporelle. Par conséquent, la composée de grandeurs est une grandeur. Par exemple, $r_1(t) * i_{r_1}(t)$ est une grandeur. Toute grandeur qui n'est pas la composée de grandeurs sera appelée grandeur élémentaire et on notera \mathbf{G} l'ensemble des grandeurs élémentaires. Pour le système illustré figure 6.1, $\mathbf{G} = \{v_g(t), v_{r_1}(t), v_{r_2}(t), i_g(t), i_{r_1}(t), i_{r_2}(t), r_1(t), r_2(t)\}$.

La valeur exacte d'une grandeur est inaccessible. A partir d'une procédure de mesure, seule une approximation de la valeur d'une grandeur (i.e la fonction temporelle) peut être obtenue. Notre intention est d'encadrer les grandeurs par des Suites de Bandes Temporelles. Toutefois, les SBT ne permettent pas de conserver certaines propriétés temporelles des fonctions telles que le fait qu'une fonction soit constante au cours du temps. C'est pourquoi nous proposons de différencier la représentation des grandeurs constantes au cours du temps de celles qui ne le sont pas. Une fonction constante $f(t)$ sera représentée par une constante réelle c . Ceci traduit le fait que, pour toute fonction constante $f(t)$, il existe un réel c tel que $f(t) = c$. Nous proposons d'encadrer les valeurs des constantes par des intervalles.

D'où la définition de l'ensemble des observations :

Définition 6.1: “Ensemble des observations”

L'ensemble des observations est un ensemble de couples

- (constante, intervalle) tel que la constante appartienne à l'intervalle,
- (fonction temporelle, SBT) tel que la fonction temporelle soit représentée par la SBT.

Par exemple, si on considère que les valeurs des résistances sont constantes au cours du temps (i.e. $r_1(t) = r_1$ et $r_2(t) = r_2$), alors un ensemble d'observations du circuit illustré figure 6.1 peut être $OBS = \{(v_g(t), SB_{v_g}), (v_{r_1}(t), SB_{v_1}), (v_{r_2}(t), SB_{v_2}), (r_1, R_1), (r_2, R_2)\}$ où SB_{v_g}, SB_{v_1} et SB_{v_2} sont des SBT et R_1 et R_2 sont des intervalles.

Remarque :

Si une constante réelle c est encadrée par un intervalle $[a, b]$, alors la fonction $f(t) = c$ est représentée sur l'intervalle temporel $[t_i, t_s]$ par la SBT $\{(t_i, t_s, a, b)\}$. Toutefois, la représentation par les SBT ne traduit pas le fait que $f(t)$ ne peut être égale qu'aux fonctions constantes représentées par $\{(t_i, t_s, a, b)\}$.

Nous présentons maintenant le formalisme de description du comportement théorique d'un système.

6.2 Description du comportement théorique

Nous introduisons d'abord la notion de composant.

Définition 6.2: “Ensemble de composants”

L'ensemble des composants, noté COMP, est un ensemble de constantes correspondant aux noms des composants.

A chaque composant, on associe le prédicat *correct* tel que *correct(c)* signifie que le composant c fonctionne normalement.

Nous introduisons maintenant la notion de modèle permettant de décrire le comportement d'un ensemble de composants.

Définition 6.3: "Modèle"

Un modèle est un triplet $(C, T, (f_0(t), \dots, f_n(t)))$ où

C est un ensemble de composants,

T est un ensemble d'intervalles temporels disjoints,

$f_0(t), \dots, f_n(t)$ sont des fonctions temporelles continues telles que $f_i(t)$ soit i fois continûment dérivable sur T (i.e. $f_i(t) \in \mathbf{F}(i)$).

Le comportement des composants C est décrit par l'équation différentielle $f_0(t) + f_1'(t) + \dots + f_n^{(n)}(t) = 0$ sur la fenêtre temporelle T .

D'une manière formelle :

Soit $(\{c_1, \dots, c_n\}, T, (f_0(t), \dots, f_n(t)))$ un modèle.

On a $correct(c_1) \wedge \dots \wedge correct(c_n) \Rightarrow \forall t_0 \in T, f_0(t_0) + f_1'(t_0) + \dots + f_n^{(n)}(t_0) = 0$

Remarque :

Le formalisme de description du comportement des composants (i.e. les modèles) permet uniquement de décrire le comportement normal des composants. Ce formalisme s'étend naturellement aux modèles de fautes [de Kleer et al, 92b] : il suffit de remplacer l'ensemble des composants du modèle par un ensemble de couples (composant, mode de fonctionnement). Cette extension n'apporte aucune contrainte supplémentaire sur la procédure de comparaison. C'est pourquoi, pour faciliter sa description, nous nous limitons aux modèles de bon fonctionnement.

Pour un modèle $(C, T, (f_0(t), \dots, f_n(t)))$, on appellera :

- C l'ensemble des composants du modèle,
- T l'extension temporelle du modèle,
- $f_0(t) + f_1'(t) + \dots + f_n^{(n)}(t) = 0$ l'équation différentielle du modèle,
- n le degré ou ordre du modèle (n est aussi le degré de l'équation différentielle),
- $f_0(t), \dots, f_n(t)$ les fonctions du modèle.

L'extension temporelle T d'un modèle représente une précondition sur la validité des modèles. L'idée est qu'un composant peut avoir plusieurs modes de fonctionnement possible. Par exemple, un transistor peut en avoir 3 : bloqué, passant, saturé. L'équation d'un modèle décrit le comportement d'un seul mode de fonctionnement, son extension temporelle correspond aux périodes pendant lesquelles le composant est dans ce mode de fonctionnement. Ces périodes sont définies dynamiquement pendant la procédure de diagnostic. C'est pourquoi l'extension temporelle d'un modèle n'est pas définie numériquement, sauf dans le cas des composants ayant un unique mode de fonctionnement : l'extension temporelle de ces modèles sera égale à l'ensemble des réels (i.e. \mathbf{R}).

Un modèle peut se réécrire de différentes manières car une équation peut avoir plusieurs expressions. Ces modèles ont tous la même signification. Par exemple, les deux modèles ci-

dessous décrivent le comportement d'une résistance :

$$(\{r\}, \mathbf{R}, (v(t) - r i(t))) \text{ (i.e. } v(t) = r i(t))$$

$$(\{r\}, \mathbf{R}, (v(t) / r - i(t)))$$

Afin de surmonter cette ambiguïté, nous introduisons la notion d'équivalence de deux modèles.

Définition 6.4: "Equivalence de deux modèles"

Deux modèles sont équivalents si et seulement si

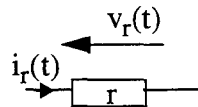
- leurs ensembles de composants sont identiques,
- leurs extensions temporelles sont identiques,
- les grandeurs invoquées dans les équations de ces modèles sont identiques,
- les ensembles de tuples de grandeurs qui sont solutions des équations sont identiques.

Nous proposons maintenant d'illustrer la définition du modèle à l'aide de quelques exemples.

6.3 Exemple de modèles

Nous noterons $v_c(t)$ la tension aux bornes du composant c , $i_c(t)$ l'intensité du courant y entrant et c_v la valeur de ce composant.

Modèle d'une résistance r :



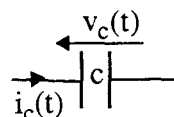
Le modèle d'une résistance est directement issu de la loi d'Ohm : la tension aux bornes d'une résistance est proportionnelle à l'intensité du courant y passant. Soit le modèle :

$$(\{r\}, \mathbf{R}, (v_r(t) - r_v i_r(t)))$$

En général, la valeur exacte d'une résistance (i.e. r_v) n'est pas connue. Par contre, un encadrement de cette valeur est toujours déterminable. Par exemple, la valeur d'une résistance dite de $10k\Omega$ à 1% appartient à l'intervalle $[9900, 10100]$. D'où l'observation $(r_v, [9900, 10100])$. On remarque que cette observation ne provient pas d'une mesure, mais est directement issue de la description du modèle.

Nous présentons maintenant un modèle d'ordre 1.

- Modèle d'une capacité c :

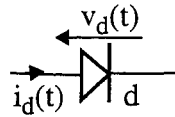


Le modèle d'une capacité est directement issu de la loi physique représentant son comportement (i.e. $i_c(t) = c_v v_c'(t)$). D'où le modèle :

$$(\{c\}, \mathbf{R}, (i_c(t), -c_v v_c(t)))$$

Les modèles de la résistance et de la capacité sont issus des modèles physiques de ces composants. Nous proposons maintenant un modèle d'un composant plus complexe : la diode. Le modèle proposé n'est pas issu du modèle physique de ce composant, mais a été décrit par un expert.

- Modèle d'une diode d :



Le comportement d'une diode est caractérisé par une équation de la forme $i_d(t) = p_1 \exp(p_2 v_d(t))$

Modéliser le comportement d'une diode directement à partir de cette relation semble difficile car les valeurs des paramètres p_1 et p_2 ne sont pas connues. Par contre, à partir de la caractéristique d'une diode, il est possible d'établir des relations entre les valeurs du courant et de la tension. Le modèle que nous proposons est une transcription dans notre formalisme du modèle de CATS.

Modèle de CATS (partiel) :

$$\begin{aligned} \text{correct}(d) &\Rightarrow [v_d(t) \geq 1 \Rightarrow i_d(t) \in [0.2, 0.5]] \\ \text{correct}(d) &\Rightarrow [v_d(t) \geq 0.92 \Rightarrow i_d(t) \in [0.1, 0.5]] \\ &\dots \\ \text{correct}(d) &\Rightarrow [v_d(t) \geq 0.0 \Rightarrow i_d(t) \in [0.0, 0.5]] \\ \text{correct}(d) &\Rightarrow [v_d(t) \leq 1.0 \Rightarrow i_d(t) \in [-1.0E-7, 0.5]] \\ \text{correct}(d) &\Rightarrow [v_d(t) \leq 0.87 \Rightarrow i_d(t) \in [-1.0E-7, 0.2]] \\ &\dots \\ \text{correct}(d) &\Rightarrow [v_d(t) \leq 0.0 \Rightarrow i_d(t) \in [-1.0E-7, 0.0]] \end{aligned}$$

Transcription :

$$\begin{aligned} &(\{d\}, \{t \mid v_d(t) \geq 1\}, (i_d(t) - I_{s0}(t))) \\ &(\{d\}, \{t \mid v_d(t) \geq 0.92\}, (i_d(t) - I_{s1}(t))) \\ &\dots \\ &(\{d\}, \{t \mid v_d(t) \geq 0.0\}, (i_d(t) - I_{s8}(t))) \\ &(\{d\}, \{t \mid v_d(t) \leq 1.0\}, (i_d(t) - I_{i0}(t))) \\ &(\{d\}, \{t \mid v_d(t) \leq 0.87\}, (i_d(t) - I_{i1}(t))) \\ &\dots \\ &(\{d\}, \{t \mid v_d(t) \leq 0.0\}, (i_d(t) - I_{i7}(t))) \end{aligned}$$

où $I_{s0}(t)$, ..., $I_{s8}(t)$ sont des grandeurs (fictives) respectivement représentées par les SBT $\{(-\infty, +\infty, 0.2, 0.5)\}$, ..., $\{(-\infty, +\infty, 0.1, 0.5)\}$. De la même façon que pour la résistance, ces SBT sont des observations.

Le comportement d'une diode est donc décrit par un ensemble de modèles, chaque modèle correspondant à une partie du comportement de la diode. L'extension temporelle des modèles permet de décrire une condition sur la validité de l'équation différentielle, condition appelée précondition par [Dague et al, 90a].

Un modèle permet de décrire le comportement d'un ou d'un ensemble de composants. Nous présentons maintenant la description d'un système.

6.4 Description du système

Définition 6.5: "Description du système"

La description du système, notée SD, est un ensemble de modèles.

Exemple :

Décrire le comportement d'un système consiste à décrire le comportement des composants et la structure de ce système. Par exemple, décrire le comportement du système illustré figure 6.2 consiste à décrire les modèles des résistances r_1 , r_2 , r_3 , ... ainsi que les connexions entre ces composants.

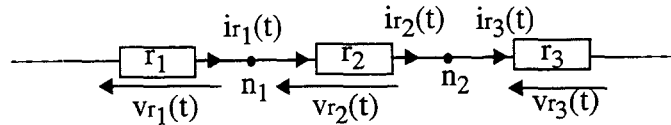


Figure 6.2 : Exemple de système

La définition de SD que nous avons proposée sous-entend que la structure du système (i.e. les connexions entre les composants) est décrite par un ensemble de modèles. Nous proposons de considérer les noeuds comme des composants et de décrire la structure des systèmes par l'intermédiaire de leurs modèles dont les équations sont issues de la loi de Kirchhoff. Par exemple, le modèle du noeud n illustré figure 6.3 est $(\{n\}, \mathbf{R}, (i_1(t) + i_2(t) - i_3(t)))$.

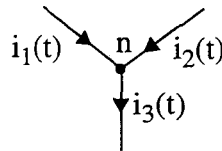


Figure 6.3 : Exemple de noeud

La description du système illustré figure 6.2 comprendra les modèles des résistances r_1 , r_2 et r_3 ainsi que les modèles des noeuds n_1 et n_2 .

Il est possible de décrire des modèles génériques pour les composants. Par exemple, le modèle générique d'une résistance r de valeur r_v est modèle(résistance, r , r_v) = $(\{r\}, (v(r ; t) - r_v i(r ; t)))$ où $v(r ; t)$ et $i(r ; t)$ sont les grandeurs représentant respectivement la tension et le courant. Décrire un système consiste à instancier les modèles génériques. Par exemple, un système composé de deux résistances r_1 et r_2 sera décrit par les modèles modèle(résistance, r_1 , r_{v1}) et modèle(résistance, r_2 , r_{v2}).

Chapitre 7

Inconsistance d'un modèle

Ce chapitre présente la formalisation de la procédure permettant de vérifier la cohérence entre le comportement théorique d'un système, dont la description est fournie par un modèle, et les observations de ce système, décrites par un ensemble de SBT.

Le but de cette formalisation est de définir la complétude de cette procédure.

Nous supposons dans un premier temps que les observations sont les SBT représentant les fonctions d'un modèle. Ceci permet d'aboutir à la définition de la consistance. Cette définition est ensuite étendue à la C-consistance. Les liens entre la consistance et la C-consistance sont étudiés dans le dernier paragraphe.

7.1 Inconsistance d'un modèle

Soit $(\{c_1, \dots, c_n\}, T, (f_0(t), \dots, f_n(t)))$ un modèle.

Nous supposons dans ce paragraphe que l'ensemble des observations est composé des SBT représentant les fonctions de ce modèle. Soit $OBS = \{(f_0(t), SB_0), \dots, (f_n(t), SB_n)\}$.

La définition d'un modèle nous assure que :

$$correct(c_1) \wedge \dots \wedge correct(c_n) \Rightarrow \forall t_0 \in T, f_0(t_0) + f_1'(t_0) + \dots + f_n^{(n)}(t_0) = 0$$

La contraposée de cette définition est :

$$\exists t_0 \in T, f_0(t_0) + f_1'(t_0) + \dots + f_n^{(n)}(t_0) \neq 0 \Rightarrow \neg correct(c_1) \vee \dots \vee \neg correct(c_n)$$

Détecter un composant défectueux consiste donc à montrer qu'il existe un instant où l'équation différentielle du modèle n'est pas satisfaite. En terme de comportement, ceci revient à assurer que les comportements réel et théorique d'un système sont différents.

- Le comportement réel du système est décrit par le tuple de fonctions $(f_0(t), \dots, f_n(t))$. Ce comportement est unique.

- Le comportement théorique est décrit par l'équation du modèle, soit $f_0(t) + f_1'(t) + \dots + f_n^{(n)}(t) = 0$.

Le comportement théorique d'un système représente un ensemble de comportements possibles, l'ensemble des solutions de l'équation différentielle du modèle (i.e. $f_0(t) + \dots + f_n^{(n)}(t) = 0$), c'est-à-dire l'ensemble $\{(f_0(t), \dots, f_n(t)) \text{ tel que } f_0(t) + \dots + f_n^{(n)}(t) = 0\}^1$. Par exemple, le modèle de la capacité représente l'ensemble des comportements $\{(i(t), -c_v v(t)) \text{ tel que } i(t) - c_v v'(t) = 0\}$. Comparer les comportements consiste à vérifier si le comportement réel appartient à cet ensemble.

1. L'ensemble de comportements possibles associé à un modèle est la définition de la description du comportement d'un ensemble de composants proposée dans [Struss, 89]. Toutefois, cette écriture n'est pas adaptée à notre formalisation de la procédure de comparaison des comportements.

Toutefois, le comportement réel d'un système n'est pas connu. Seule une approximation de ce comportement est fournie, c'est l'ensemble des observations. L'ensemble des observations représente l'ensemble des comportements réels possibles, c'est à dire que le comportement réel du système (i.e le tuple $(f_0(t), \dots, f_n(t))$) appartient à l'ensemble $\{(g_0(t), \dots, g_n(t)) \text{ tel que } g_0(t) \text{ Rep } SB_0, \dots, g_n(t) \text{ Rep } SB_n\}$.

Comparer les comportements réel et théorique d'un système revient donc à comparer les observations et le comportement théorique de ce système. Nous nous intéressons d'abord à toute différence entre ces comportements. On peut montrer que les comportements réel et théorique sont différents lorsque les ensembles de comportements possibles associés aux observations et au modèle ne s'intersectent pas (cf. figure 7.1).

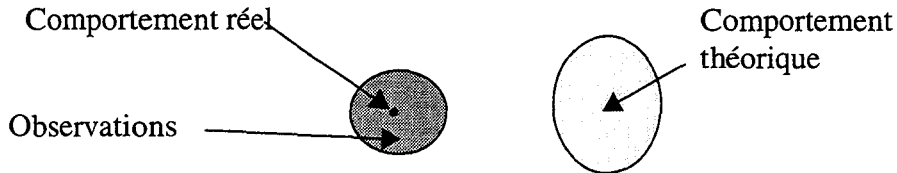


Figure 7.1 : Comportements réel et théorique différents

Nous introduisons la notion d'inconsistance pour décrire la différence entre les comportements.

Définition 7.1: "Inconsistance d'un modèle"

Un modèle $(C, T, (f_0(t), \dots, f_n(t)))$ est inconsistant pour un n-uplet de suites de bandes temporelles (SB_0, \dots, SB_n) si et seulement si l'équation différentielle du modèle (i.e. $f_0(t) + \dots + f_n^{(n)}(t) = 0$) n'a pas de solution dans l'espace des solutions que représentent les suites de bandes temporelles, c'est à dire l'ensemble $\{(g_0(t), \dots, g_n(t)) \text{ tel que } g_0(t) \text{ Rep } SB_0, \dots, g_n(t) \text{ Rep } SB_n\}$.

D'une manière formelle :

$(C, T, (f_0(t), \dots, f_n(t))) \text{ inconsistant pour } (SB_0, \dots, SB_n)$ $\Leftrightarrow \forall g_0(t) \text{ Rep } SB_0, \dots, \forall g_n(t) \text{ Rep } SB_n, \exists t_0 \in T, g_0(t_0) + g_1'(t_0) + \dots + g_n^{(n)}(t_0) \neq 0$

où les fonctions g_i sont de classe C^i

Nous dirons qu'un modèle est consistant pour un tuple de SBT si et seulement si il n'est pas inconsistant pour ce tuple de SBT.

Nous proposons une interprétation de l'inconsistance en terme de composants défectueux.

Propriété 7.1: "Composants d'un modèle inconsistant"

Si un modèle est inconsistant pour un tuple de SBT, alors au moins un des composants de ce modèle est défectueux.

D'une manière formelle :

$$(\{c_1, \dots, c_n\}, T, (f_0(t), \dots, f_n(t))) \text{ inconsistant pour } (SB_0, \dots, SB_n)$$

$$\Rightarrow \neg \text{correct}(c_1) \vee \dots \vee \neg \text{correct}(c_n)$$

Preuve :

Si le modèle $(\{c_1, \dots, c_n\}, T, (f_0(t), \dots, f_n(t)))$ est inconsistant pour (SB_0, \dots, SB_n) , alors il existe un instant $t_0 \in T$ tel que $f_0(t_0) + f_1'(t_0) + \dots + f_n^{(n)}(t_0) \neq 0$. Il découle alors directement de la définition d'un modèle qu'un des composants c_1, \dots, c_n est défectueux (i.e. $\neg correct(c_1) \vee \dots \vee \neg correct(c_n)$).

□

D'après cette propriété, détecter un composant défectueux consiste à montrer qu'un modèle est inconsistant pour un tuple de SBT. La question qui se pose alors est : si un modèle n'est pas inconsistant pour un tuple de SBT, est-il possible de détecter un composant défectueux? Si un modèle n'est pas inconsistant pour un tuple de SBT alors les ensembles de comportements associés au modèle et aux observations (i.e. les SBT) s'intersectent. D'après la définition d'un modèle, détecter un composant défectueux consiste à montrer que les comportements réel et théorique sont différents. Or, assurer l'intersection de ces ensembles ne permet pas de savoir si les comportements réel et théorique sont identiques ou différents (cf. figure 7.2).

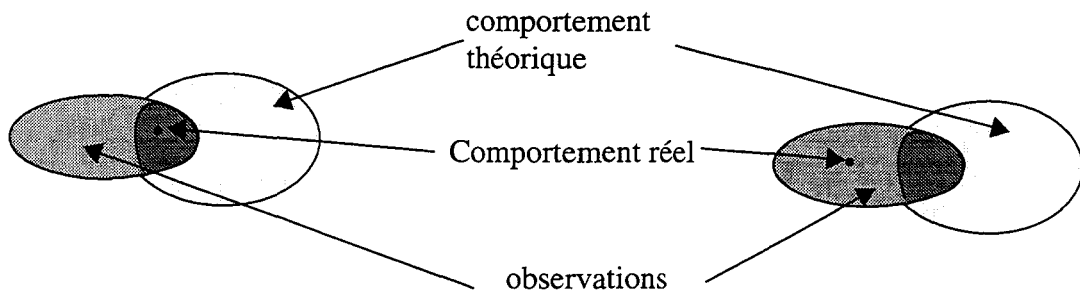


Figure 7.2a : Comportements identiques

Figure 7.2b : Comportements différents

Par conséquent, la notion d'inconsistance peut se résumer à :

- Si un modèle est inconsistant pour un tuple de SBT alors les comportements réel et théorique sont différents. Au moins un des composants du modèle est défectueux.
- Si un modèle n'est pas inconsistant pour un tuple de SBT alors l'imprécision introduite par les SBT ne permet pas de comparer les comportements réel et théorique. Aucune information sur l'état de fonctionnement des composants du modèle ne peut être déduite.

Il est aussi possible d'assurer que les comportements réel et théorique d'un système sont identiques. En effet, si l'ensemble des comportements réels possibles (i.e. ensemble des comportements associé à OBS) est inclus dans celui associé au modèle, alors les comportements sont identiques (cf figure 7.3).

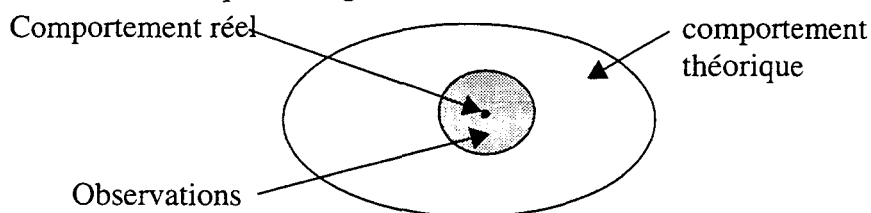


Figure 7.3 : Comportements réel et théorique identique

Toutefois, cette relation est peu intéressante. En effet, elle n'apporte aucune information sur le comportement des composants : la définition d'un modèle faisant intervenir une implication,

assurer que les comportements réel et théorique sont identiques (i.e. $f_0(t) + \dots + f_n^{(n)}(t) = 0$) ne permet pas d'assurer que les composants fonctionnent. De plus, obtenir une telle relation entre les ensembles de comportements nécessite que les observations représentent un unique comportement, c'est à dire que chaque SBT doit représenter une seule fonction. Ceci revient à supposer que le comportement réel du système est connue, hypothèse que nous ne faisons pas.

Nous étendons maintenant la définition de l'inconsistance.

7.2 C-inconsistance d'un modèle

Soit $(C, T, (f_0(t), \dots, f_n(t)))$ un modèle. En général, les fonctions $f_0(t), \dots, f_n(t)$ ne sont pas des grandeurs élémentaires du système, mais sont des composées de grandeurs élémentaires. Par exemple, le modèle de la capacité est $(\{c\}, \mathbf{R}, (i(t), -c_v v(t)))$. La fonction $f_1(t) = -c_v v(t)$ est composée des grandeurs élémentaires c_v et $v(t)$.

Nous supposons dans ce paragraphe que l'ensemble des observations est composé des observations des grandeurs élémentaires du système. Soient l'ensemble des grandeurs élémentaires $G = \{h_0(t), \dots, h_p(t), c_0(t), \dots, c_m(t)\}$ et l'ensemble des observations $OBS = \{(h_0(t), SB_0), \dots, (h_p(t), SB_p), (c_0, I_0), \dots, (c_m, I_m)\}$ (les grandeurs $c_0(t), \dots, c_m(t)$ sont constantes au cours du temps et on notera $c_i(t) = c_i$).

Afin d'exprimer les fonctions du modèle à partir des grandeurs élémentaires du système, nous introduisons les opérations op_0, \dots, op_n telles que :

$$f_0(t) = op_0(h_0(t), \dots, h_p(t) ; c_0, \dots, c_m),$$

...

$$f_n(t) = op_n(h_0(t), \dots, h_p(t) ; c_0, \dots, c_m)$$

Par exemple, pour le modèle de la capacité $(\{c, \mathbf{R}, (i(t), -c_v v(t))\})$, on a :

$$G = \{i(t), v(t), c_v(t)\} \text{ avec } c_v(t) = c_v,$$

$$f_0(t) = op_0(i(t), v(t) ; c_v) = i(t)$$

$$f_1(t) = op_1(i(t), v(t) ; c_v) = -c_v v(t)$$

Les dérivées des fonctions du modèle seront notées :

$$op_1^{(j)}(t) = op_1^{(j)}(h_0(t), \dots, h_p(t) ; c_0, \dots, c_m)$$

Nous introduisons maintenant la notion de C-inconsistance d'un modèle pour un ensemble d'observation.

Définition 7.2: "C-inconsistance"

Un modèle $(C, T, (f_0(t), \dots, f_n(t)))$ est C-inconsistant pour l'ensemble des observations $\{(h_0(t), SB_0), \dots, (h_p(t), SB_p), (c_0, I_0), \dots, (c_m, I_m)\}$ si et seulement si l'équation différentielle du modèle (i.e $f_0(t) + \dots + f_n^{(n)}(t) = 0$) n'a pas de solution dans l'espace des solutions que représente l'ensemble des observations, c'est à dire l'ensemble $\{(f_0(t), \dots, f_n(t)) \text{ tel que } h_0(t) \text{ Rep } SB_0, \dots, h_p(t) \text{ Rep } SB_p, c_0 \in I_0, \dots, c_m \in I_m\}$.

D'une manière formelle :

$$\begin{aligned}
 & (C, T, (op_0(h_0(t), \dots, h_p(t); c_0, \dots, c_m), \dots, op_n(h_0(t), \dots, h_p(t); c_0, \dots, c_m))) \\
 & C\text{-inconsistant pour } \{(h_0(t), SB_0), \dots, (h_p(t), SB_p), (c_0, I_0), \dots, (c_m, I_m)\} \\
 \Leftrightarrow & \forall g_0(t) \text{ Rep } SB_0, \dots, \forall g_p(t) \text{ Rep } SB_p, \forall b_0 \in I_0, \dots, \forall b_m \in I_m, \\
 & \exists t_0 \in T, op_0(\{g_0(t_0), \dots, g_p(t_0)\}; \{b_0, \dots, b_m\}) + \dots + op_n^{(n)}(\{g_0(t_0), \dots, g_p(t_0)\}; \{b_0, \dots, b_m\}) \neq 0
 \end{aligned}$$

Nous dirons qu'un modèle est C-consistant pour un ensemble d'observation OBS si et seulement si il n'est pas C-inconsistant pour OBS.

La consistance est un cas particulier de la C-consistance : les fonctions des modèles sont des grandeurs élémentaires. Soit :

$$\begin{aligned}
 & M(C, T, (f_0(t), \dots, f_n(t))) \text{ consistant pour } (SB_0, \dots, SB_n) \\
 \Leftrightarrow & M(C, T, (f_0(t), \dots, f_n(t))) \text{ C-consistant pour } \{(f_0(t), SB_0), \dots, (f_n(t), SB_n)\}
 \end{aligned}$$

La C-inconsistance a la même signification que l'inconsistance : un modèle est C-inconsistant pour un ensemble d'observations si et seulement si les comportements décrits par le modèle et les observations sont différents. La propriété suivante est l'interprétation de la C-inconsistance en terme de mode de composants défectueux.

Propriété 7.2: “Composants d'un modèle inconsistant pour un tuple de SBT”

Si un modèle est C-inconsistant pour un ensemble d'observations alors au moins un des composants de ce modèle est défectueux.

D'une manière formelle :

$$\begin{aligned}
 & (\{c_1, \dots, c_n\}, T, (f_0(t), \dots, f_n(t))) \text{ C-inconsistant pour OBS} \\
 \Rightarrow & \neg correct(c_1) \vee \dots \vee \neg correct(c_n)
 \end{aligned}$$

Les propriétés des opérations réelles étant différentes des propriétés des opérations sur les SBT, les notions de consistance et de C-consistance ne sont pas équivalentes. Nous établissons maintenant les liens existant entre la consistance et la C-consistance.

7.3 Liens entre la consistance et la C-consistance

Soient un modèle $(C, T, (f_0(t), \dots, f_n(t)))$ et un ensemble d'observations OBS. Nous appellerons OBS-extension de la fonction $f_i(t)$ la SBT obtenue en remplaçant dans l'expression de la fonction du modèle

- les grandeurs constantes par les intervalles les encadrant,
- les grandeurs variant au cours du temps par les SBT les représentant,
- les opérations par leurs extensions aux SBT.

Par exemple, pour le modèle $(\{c\}, \mathbf{R}, (i(t), -c_v v(t)))$ et l'ensemble des observations $OBS = \{(i(t), SB_i), (c(t), SB_v), (c_v, C)\}$, l'OBS-extension de $f_1(t) = -c_v v(t)$ est la $SBT - C SB_v$.

D'une manière générale, l'OBS-extension de la fonction du modèle est l'extension aux SBT de la fonction du modèle instanciée par les valeurs obtenues dans OBS. Pour un ensemble d'observations $OBS = \{(h_0(t), SB_0), \dots, (h_p(t), SB_p), (c_0, I_0), \dots, (c_m, I_m)\}$ et une fonction $f_i(t) = op_i(h_0(t), \dots, h_p(t); c_0, \dots, c_m)$, on notera $OP_i(SB_0, \dots, SB_p; I_0, \dots, I_m)$ l'OBS-extension de $f_i(t)$. L'opération OP_i est l'extension aux SBT de op_i .

Nous établissons maintenant le premier lien existant entre l'inconsistance et la C-inconsistance.

Théorème 7.2: "L'inconsistance implique la C-inconsistance"

Si un modèle est inconsistant pour les OBS-extensions des fonctions du modèle, alors il est C-inconsistant pour l'ensemble des observations.

D'une manière formelle :

$$(C, T, (f_0(t), \dots, f_n(t))) \text{ inconsistant pour } (OP_0(SB_0, \dots, SB_p; I_0, \dots, I_m), \dots, OP_n(SB_0, \dots, SB_p; I_0, \dots, I_m))$$

$$\Rightarrow (C, T, (op_0(h_0(t), \dots, h_p(t); c_0, \dots, c_m), \dots, op_n(h_0(t), \dots, h_p(t); c_0, \dots, c_m)))$$

$$C\text{-inconsistant pour } \{(h_0(t), SB_0), \dots, (h_p(t), SB_p), (c_0, I_0), \dots, (c_m, I_m)\}$$

Preuve :

Nous montrons que la C-consistance implique la consistance.

$$(C, T, (op_0(h_0(t), \dots, h_p(t); c_0, \dots, c_m), \dots, op_n(h_0(t), \dots, h_p(t); c_0, \dots, c_m)))$$

$$C\text{-consistant pour } \{(h_0(t), SB_0), \dots, (h_p(t), SB_p), (c_0, I_0), \dots, (c_m, I_m)\}$$

$$\Leftrightarrow \exists g_0(t) \text{ Rep } SB_0, \dots, \exists g_p(t) \text{ Rep } SB_p, \exists b_0 \in I_0, \dots, \exists b_m \in I_m,$$

$$\forall t_0 \in T, op_0(g_0(t_0), \dots, g_p(t_0); b_0, \dots, b_m) + \dots + op_n^{(n)}(g_0(t_0), \dots, g_p(t_0); b_0, \dots, b_m) = 0$$

Or, les opérations sur les SBT conservent la propriété de représentation :

$$g_0(t) \text{ Rep } SB_0, \dots, g_p(t) \text{ Rep } SB_p, b_0 \in I_0, \dots, b_m \in I_m$$

$$\Rightarrow op_i(g_0(t_0), \dots, g_p(t_0); b_0, \dots, b_m) \text{ Rep } OP_i(SB_0, \dots, SB_p; I_0, \dots, I_m)$$

D'où

$$(C, T, (op_0(h_0(t), \dots, h_p(t); c_0, \dots, c_m), \dots, op_n(h_0(t), \dots, h_p(t); c_0, \dots, c_m)))$$

$$C\text{-consistant pour } \{(h_0(t), SB_0), \dots, (h_p(t), SB_p), (c_0, I_0), \dots, (c_m, I_m)\}$$

$$\Rightarrow \exists k_0(t) \text{ Rep } OP_0(SB_0, \dots, SB_p; I_0, \dots, I_m),$$

$$\dots$$

$$\exists k_n(t) \text{ Rep } OP_n(SB_0, \dots, SB_p; I_0, \dots, I_m),$$

$$\forall t_0 \in T, k_0(t_0) + \dots + k_n^{(n)}(t_0) = 0$$

(il suffit de choisir k_i tel que $k_i(t) = op_i(g_0(t), \dots, g_p(t); b_0, \dots, b_m)$ pour $i = 1 \dots n$)

$$\Leftrightarrow (C, T, (op_0(\{h_0(t), \dots, h_p(t)\}; \{a_0, \dots, a_m\}), \dots, op_n(\{h_0(t), \dots, h_p(t)\}; \{a_0, \dots, a_m\})))$$

$$\text{consistant pour } (OP_0(\{SB_0, \dots, SB_m\}, \{I_0, \dots, I_p\}), \dots, OP_n(\{SB_0, \dots, SB_m\}, \{I_0, \dots, I_p\}))$$

□

Exemple :

Soit le modèle d'une capacité : $(\{c\}, \mathbf{R}, (i_c(t), -c_v v_c(t)))$.

$(\{c\}, \mathbf{R}, (i_c(t), -c_v v_c(t)))$ C-consistant pour $\{(i_c(t), SB_i), (v_c(t), SB_v), (c_v, C)\}$

$\Rightarrow (\{c\}, \mathbf{R}, (i_c(t), -c_v v_c(t)))$ consistant pour $(SB_i, -C SB_v)$

Nous établissons maintenant un cas où ces notions sont équivalentes.

Théorème 7.3: "Equivalence de la C-inconsistance et de l'inconsistance"

La C-inconsistance d'un modèle $(C, T, (f_0(t), \dots, f_n(t)))$ pour un ensemble d'observation OBS est équivalente à l'inconsistance de ce modèle pour les OBS-extensions des fonctions $f_0(t), \dots, f_n(t)$ lorsque les extensions aux SBT de ces fonctions n'introduisent pas d'imprécision.

D'une manière formelle :

Soient OP_0, \dots, OP_n les extensions aux SBT des opérations op_0, \dots, op_n .

Si les opérations OP_0, \dots, OP_n n'introduisent pas d'imprécision alors

$(C, T, (op_0(f_{00}(t), \dots, f_{0p}(t); c_{00}, \dots, c_{0m}), \dots, op_n(f_{n0}(t), \dots, f_{np}(t); c_{n0}, \dots, c_{nm})))$

C-inconsistant pour $\{(f_{00}(t), SB_{00}), \dots, (f_{0p}(t), SB_{0p}), \dots, (f_{n0}(t), SB_{n0}), \dots, (f_{np}(t), SB_{np}), (a_{00}, I_{00}), \dots, (a_{0m}, I_{0m}), \dots, (a_{n0}, I_{n0}), \dots, (a_{nm}, I_{nm})\}$

$\Leftrightarrow M(C, T, (f_0(t), \dots, f_n(t)))$ inconsistant pour

$(OP_0(SB_{00}, \dots, SB_{0p}; I_{00}, \dots, I_{0m}), \dots, OP_n(SB_{n0}, \dots, SB_{np}; I_{n0}, \dots, I_{nm}))$

La preuve de ce théorème découle directement de la définition de non introduction d'imprécision d'une opération sur les suites de bandes temporelles. Soit :

$\exists g_{00}(t) \text{ Rep } SB_{00} \wedge \dots \wedge \exists g_{0p}(t) \text{ Rep } SB_{0p} \wedge \exists b_{00} \in I_{00} \wedge \dots \wedge \exists b_{0m} \in I_{0m}$

$\Leftrightarrow \exists ho(t) \text{ Rep } OP_0(\{SB_{00}, \dots, SB_{0p}; I_{00}, \dots, I_{0m}\}, ho(t) = op_0(g_{00}(t), \dots, g_{0p}(t); b_{00}, \dots, b_{0m}))$

Exemple (suite) :

Soit le modèle d'une capacité $(\{c\}, \mathbf{R}, (i_c(t), -c_v v_c(t)))$ et l'ensemble des observations $OBS = \{(i_c(t), SB_i), (v_c(t), SB_v)\}$. La valeur de la capacité est supposée connue.

On a :

$(\{c\}, \mathbf{R}, (i_c(t), -c_v v_c(t)))$ C-consistant pour $\{(v_c(t), SB_v), (i_c(t), SB_i)\}$

$\Leftrightarrow (\{c\}, \mathbf{R}, (i_c(t), -c_v v_c(t)))$ C-consistant pour $\{(v_c(t), SB_v), (i_c(t), SB_i)\}$

$\Leftrightarrow (\{c\}, \mathbf{R}, (i_c(t), -c_v v_c(t)))$ consistant pour $(SB_i, -c_v SB_v)$

Remarque :

Le fait que l'inconsistance implique la C-inconsistance traduit le fait que l'ensemble des comportements associé à l'ensemble des observations est inclus dans celui associé aux OBS-extensions des fonctions du modèle. L'inconsistance et la C-inconsistance sont équivalentes

lorsque ces ensembles de comportements sont identiques (cf. figure 7.4).

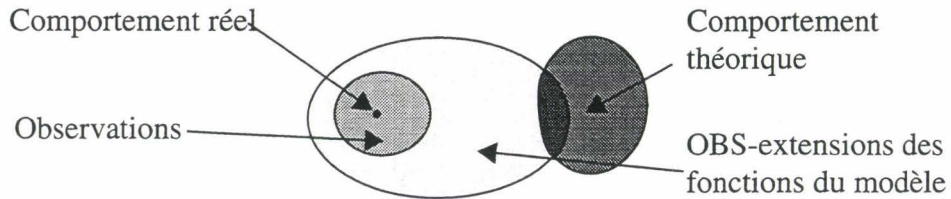


Figure 7.4 : C-inconsistance et consistance

Lorsque les extensions aux SBT des fonctions d'un modèle introduisent de l'imprécision, un modèle peut être à la fois C-inconsistant pour un ensemble d'observations OBS et consistant pour les OBS-extensions des fonctions de ce modèle. Dans ce cas, un des composants est défectueux (C-inconsistance du modèle) et cette panne ne peut pas être détectée lorsque l'on calcule les OBS-extensions des fonctions de ce modèle (consistance du modèle).

7.4 Conclusion

Détecter un composant défectueux (i.e. un ensemble de conflits) consiste à détecter l'inconsistance ou la C-inconsistance d'un modèle. En général, il faut détecter la C-inconsistance car l'ensemble des observations est composé des observations des grandeurs élémentaires et les fonctions du modèle ne sont pas des grandeurs élémentaires. Par contre, il est plus simple de détecter l'inconsistance d'un modèle.

La propriété fondamentale est que l'inconsistance implique la C-inconsistance. Cette propriété nous assure qu'aucun ensemble de conflits ne sera détecté à tort lorsque l'on vérifie uniquement l'inconsistance d'un modèle pour les OBS-extensions des fonctions de ce modèle. Par contre, l'inverse de la propriété fondamentale n'est pas toujours vrai. L'inconsistance et la C-inconsistance sont équivalentes uniquement lorsque les extensions aux SBT des fonctions des modèles n'introduisent pas d'imprécision. Cette propriété nous assure que, lorsque les extensions aux SBT des fonctions des modèles n'introduisent pas d'imprécision, tous les ensembles de conflits sont détectés en vérifiant uniquement l'inconsistance des modèles pour les OBS-extensions des fonctions de ces modèles. D'après les propriétés des opérations sur les SBT définies dans la deuxième partie, pour que la détection d'inconsistance soit suffisante, il suffit que :

- seules les opérations d'addition et de multiplication interviennent dans les fonctions des modèles,
- toute grandeur n'apparaisse qu'une seule fois dans les expressions des fonctions des modèles,
- l'ensemble des observations soit composée uniquement de SBT continues et d'intervalles dégénérés.

Chapitre 8

Famille d'algorithmes de détection d'inconsistance

Nous présentons dans ce chapitre les algorithmes de détection d'inconsistance des modèles. Après avoir introduit quelques notations facilitant l'écriture de ces algorithmes, nous proposons un algorithme de calcul de l'extension temporelle d'un modèle lorsque celle-ci est un ensemble d'instant de la forme $\{t \mid f(t) \geq 0\}$ et lorsque la fonction $f(t)$ est représentée par une SBT.

Une famille d'algorithmes de détection d'inconsistance est alors proposée, chaque algorithme correspondant à l'ordre d'un modèle. Nous présentons d'abord deux algorithmes complets, ceux pour les modèles d'ordre 0 et 1. Puis, nous présentons une extension de l'algorithme de détection d'inconsistance des modèles d'ordre 1 afin de prendre en compte les modèles d'ordre supérieur.

8.1 Préliminaires

Les procédures de détection d'inconsistance nécessitent de calculer les zéros de polynômes. Nous supposons l'existence d'un algorithme calculant l'opération Sol définie ci-dessous.

Définition 8.1: "Zéros de polynômes : opération Sol "

Nous noterons Sol l'opération qui, à un polynôme p et un intervalle T , associe l'ensemble ordonné des solutions appartenant à l'intervalle T de l'équation $p(t) = 0$.

D'une manière formelle :

Soit $p(t)$ un polynôme et T un intervalle.

$$Sol(p(t), T) = \{t_1, \dots, t_n \mid \forall i=1..n, p(t_i) = 0 \wedge t_i \in T, \\ \forall i=1..n-1, t_i < t_{i+1}\}$$

Des algorithmes numériques, tel que celui de Newton [Dahlquist et al, 74], permettent de déterminer les zéros d'un polynôme d'ordre quelconque. Ces zéros peuvent aussi être déterminés analytiquement pour les polynômes de degré inférieur ou égal à 4. L'avantage des solutions analytiques est que leur calcul est immédiat. Toutefois, étant donnée l'imprécision introduite par les ordinateurs lors du calcul d'expressions complexes, les zéros des polynômes de degré 3 et 4 obtenus par un algorithme numérique sont souvent de meilleure qualité que ceux provenant du calcul des solutions analytiques. C'est pourquoi, nous considérerons que seuls les zéros des polynômes de degré 0, 1 et 2 sont déterminés analytiquement.

Afin de faciliter l'écriture des algorithmes, nous introduisons la notion de fonction associée à une SBT.

Définition 8.2: "Fonctions associées à une suite de bandes temporelles"

Les fonctions inférieure et supérieure associées à une SBT $\{(t_1, t_2, p_{i1}, p_{s1}), \dots, (t_n, t_{n+1}, p_{in}, p_{sn})\}$, notées $F_i(SB ; t)$ et $F_s(SB ; t)$, sont des fonctions polynomiales par morceaux définies sur l'extension temporelle de SB de la manière suivante :

$$\forall t \in]t_k, t_{k+1}[, F_i(SB ; t) = p_{ik}(t) \quad \text{pour } k = 1 \dots n$$

$$F_i(SB ; t_1) = p_{i1}(t_1),$$

$$F_i(SB ; t_k) = \max(p_{ik-1}(t_k), p_{ik}(t_k)) \quad \text{pour } k = 2 \dots n$$

$$F_i(SB ; t_{n+1}) = p_{in}(t_{n+1}),$$

et

$$\forall t \in]t_k, t_{k+1}[, F_s(SB ; t) = p_{sk}(t) \quad \text{pour } k = 1 \dots n$$

$$F_s(SB ; t_1) = p_{s1}(t_1),$$

$$F_s(SB ; t_k) = \min(p_{sk-1}(t_k), p_{sk}(t_k)) \quad \text{pour } k = 2 \dots n$$

$$F_s(SB ; t_{n+1}) = p_{sn}(t_{n+1}),$$

Nous exprimons la propriété de représentation d'une fonction continue par une SBT à partir des fonctions associées à une SBT.

Lemme 8.1: "Représentation d'une fonction continue par une suite de bandes temporelles"

Une fonction temporelle continue est représentée par une suite de bandes temporelles SB si et seulement si elle est toujours comprise entre les fonctions inférieure et supérieure associées à SB.

D'une manière formelle :

Soit $f(t)$ une fonction temporelle continue et SB une suite de bandes temporelles.

$$f(t) \text{ Rep SB} \Leftrightarrow \forall t \in [t_i(SB), t_s(SB)], F_i(SB ; t) \leq f(t) \leq F_s(SB ; t)$$

Preuve :

Soit $SB = \{(t_1, t_2, p_{i1}, p_{s1}), \dots, (t_n, t_{n+1}, p_{in}, p_{sn})\}$ une suite de bandes temporelles.

Il découle directement de la propriété de représentation que :

$$f(t) \text{ Rep SB} \Leftrightarrow \forall k = 1 \dots n, \forall t \in [t_k, t_{k+1}[, p_{ik}(t) \leq f(t) \leq p_{sk}(t)$$

Montrer ce lemme consiste donc à montrer que :

$$f(t) \text{ Rep SB} \Rightarrow p_{ik-1}(t_k) \leq f(t_k) \leq p_{sk-1}(t_k)$$

Par définition de la représentation, on a :

$$f(t) \text{ Rep SB} \Rightarrow \forall t \in [t_{k-1}, t_k[, p_{ik-1}(t) \leq f(t) \leq p_{sk-1}(t)$$

$$D'où f(t) \text{ Rep SB} \Rightarrow \lim_{\theta \rightarrow t_k^-} p_{ik-1}(\theta) \leq \lim_{\theta \rightarrow t_k^-} f(\theta) \leq \lim_{\theta \rightarrow t_k^-} p_{sk-1}(\theta)$$

Comme les fonctions $p_{ik-1}(t)$, $f(t)$ et $p_{sk-1}(t)$ sont continues, on a :

$$\lim_{\theta \rightarrow t_k^-} p_{ik-1}(\theta) = p_{ik-1}(t_k), \quad \lim_{\theta \rightarrow t_k^-} f(\theta) = f(t_k), \quad \lim_{\theta \rightarrow t_k^-} p_{sk-1}(\theta) = p_{sk-1}(t_k),$$

$$D'où f(t) \text{ Rep SB} \Rightarrow \forall t \in [t_{k-1}, t_k[, p_{ik-1}(t_k) \leq f(t_k) \leq p_{sk-1}(t_k)$$

□

Nous proposons maintenant de définir un algorithme de calcul d'un ensemble d'instant de la forme $\{t \mid f(t) \geq 0\}$. Cet ensemble d'instant correspond à une définition courante de l'extension temporelle d'un modèle. C'est par exemple le cas pour les modèles d'une diode qui sont de la forme $(\{d\}, \{t \mid v_d(t) \geq cte\}, (i_d(t) - I_{di}(t)))$

8.2 Calcul de l'extension temporelle d'un modèle

De nombreuses possibilités sont envisageables pour définir l'extension temporelle d'un modèle. Il est par exemple possible d'envisager l'utilisation de connaissances empiriques. Par exemple, le comportement d'un transistor (cf figure 8.1) en régime passant est décrit par l'équation $i_c(t) = \beta i_b(t)$. Le modèle correspondant est $(\{q\}, \{t \mid passant(q, t)\}, (i_c(t) - \beta i_b(t)))$.

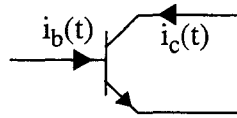


Figure 8.1 : Schéma d'un transistor

L'utilisation d'un tel modèle nécessite une procédure permettant de calculer l'ensemble des instants $\{t \mid passant(q, t)\}$.

Nous proposons dans ce paragraphe un algorithme de calcul de l'extension temporelle d'un modèle lorsque celle-ci est un ensemble d'instant de la forme $\{t \mid f(t) \geq 0\}$ où $f(t)$ est une fonction continue. Le fait que l'extension temporelle des modèles est souvent définie comme un ensemble d'instant de cette forme est la principale motivation de ce choix. L'algorithme est basé sur le lemme suivant.

Lemme 8.2: "Calcul des extensions temporelles"

L'ensemble des instants pour lesquels une fonction continue $f(t)$ représentée par une suite de bandes temporelles SB est positive contient l'ensemble des instants pour lesquels le polynôme inférieur associé à SB est positif.

D'une manière formelle :

$$\{t_0 \mid f(t_0) \geq 0 \wedge f(t) \text{ Rep SB}\} \supseteq \{t_0 \mid F_i(\text{SB} ; t_0) \geq 0\}$$

Preuve :

D'après la propriété de représentation, on a :

$$f(t) \text{ Rep SB} \Leftrightarrow \forall t_0 \in [t_i(\text{SB}), t_s(\text{SB})], F_i(\text{SB} ; t_0) \leq f(t_0) \leq F_s(\text{SB} ; t_0)$$

D'où $F_i(\text{SB} ; t_0) \geq 0 \Rightarrow f(t_0) \geq 0$

et $\{t_0 \mid f(t_0) \geq 0 \wedge f(t) \text{ Rep SB}\} \supseteq \{t_0 \mid F_i(\text{SB} ; t_0) \geq 0\}$

□

L'algorithme découle directement de ce lemme :

Calcul de $\{t \mid f(t) \geq 0\}$ lorsque $f(t)$ est représentée par la SBT $\{(t_1, t_2, p_{i1}, p_{s1}), \dots, (t_n, t_{n+1}, p_{in}, p_{sn})\}$
$T \leftarrow \{\}$ Pour $k = 1$ à n faire Calculer $Sol(p_{ik}, [t_k, t_{k+1}]) = \{x_1, \dots, x_p\}$ Si $p_{ik}((t_k + x_1)/2) \geq 0$ alors $T \leftarrow T \cup [t_k, x_1]$ Fin si Si $p_{ik}((x_p + t_{k+1})/2) \geq 0$ alors $T \leftarrow T \cup [x_p, t_{k+1}]$ Fin si Pour $j = 1$ à $p-1$ faire Si $p_{ik}((x_j + x_{j+1})/2) \geq 0$ alors $T \leftarrow T \cup [x_j, x_{j+1}]$ Fin si Fin pour Fin pour

D'autres ensembles d'instant peuvent être calculés à partir de cet algorithme. En effet :
 $\{t_0 \mid f(t_0) \leq 0 \wedge f(t) \text{ Rep SB}\} = \{t_0 \mid g(t_0) \geq 0 \wedge g(t) \text{ Rep -SB}\}$

$$\begin{aligned} & \{t_0 \mid f_1(t_0) \geq 0 \wedge f_2(t_0) \geq 0 \wedge f_1(t) \text{ Rep SB}_1 \wedge f_2(t) \text{ Rep SB}_2\} \\ & = \{t_0 \mid f_1(t_0) \geq 0 \wedge f_1(t) \text{ Rep SB}_1\} \cap \{t_0 \mid f_2(t_0) \geq 0 \wedge f_2(t) \text{ Rep SB}_2\} \end{aligned}$$

$$\begin{aligned} & \{t_0 \mid (f_1(t_0) \geq 0 \wedge f_1(t) \text{ Rep SB}_1) \vee (f_2(t_0) \geq 0 \wedge f_2(t) \text{ Rep SB}_2)\} \\ & = \{t_0 \mid f_1(t_0) \geq 0 \wedge f_1(t) \text{ Rep SB}_1\} \cup \{t_0 \mid f_2(t_0) \geq 0 \wedge f_2(t) \text{ Rep SB}_2\} \end{aligned}$$

D'un point de vue pratique, il serait utile de pouvoir calculer l'ensemble des instants pour lesquels la dérivée d'une fonction représentée par une SBT est positive. Toutefois, on peut montrer qu'un tel ensemble est toujours vide (i.e. $\{t_0 \mid f'(t_0) \geq 0 \wedge f(t) \text{ Rep SB}\} = \emptyset$). En effet, pour tout instant t_0 , une SBT non dégénérée représente toujours une fonction croissante et une fonction décroissante à l'instant t_0 . Par exemple, à l'instant t_0 , les signes des dérivées des fonctions $f(t)$ et $g(t)$ illustrées figure 8.2 sont opposés.

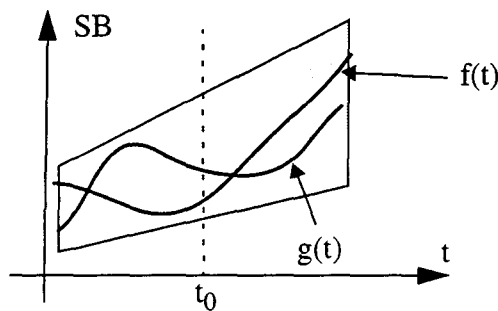


Figure 8.2 : Dérivée des fonctions représentées par une bande temporelle d'ordre 1

Nous proposons maintenant un algorithme de détection d'inconsistance d'un modèle d'ordre 0.

8.3 Inconsistance des modèles d'ordre 0

Afin de faciliter l'écriture, l'algorithme sera décrit pour un modèle et une SBT dont les extensions temporelles sont identiques. L'extension au cas général est immédiate, il suffit de découper l'extension temporelle de la SBT pour se ramener au cas traité dans ce chapitre.

Soient $(C, T, (f(t)))$ un modèle d'ordre 0 et SB une SBT dont l'extension temporelle est T.

Réécrivons la définition de l'inconsistance pour un modèle d'ordre 0 :

$$(C, T, (f(t))) \text{ inconsistant pour (SB)} \Leftrightarrow \forall g(t) \text{ Rep SB}, \exists t_0 \in T, g(t_0) \neq 0$$

Détecter l'inconsistance d'un modèle d'ordre 0 consiste à prouver l'existence d'un instant tel que toutes les fonctions représentées par SB soient différentes de 0.

Inversons cette équivalence :

$$(C, T, (f(t))) \text{ consistant pour (SB)} \Leftrightarrow \exists g(t) \text{ Rep SB}, \forall t_0 \in T, g(t_0) = 0$$

Un modèle d'ordre 0 est consistant pour une suite de bandes temporelles SB si et seulement si la fonction identiquement nulle est représentée par SB. Par exemple, tout modèle d'ordre 0 est respectivement consistant et inconsistant pour les suites de bandes temporelles SB₁ et SB₂ illustrées figures 8.3a et 8.3b.

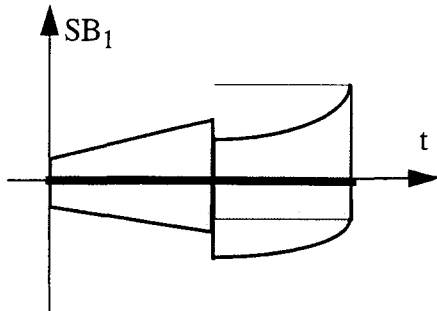


Figure 8.3a : Consistance

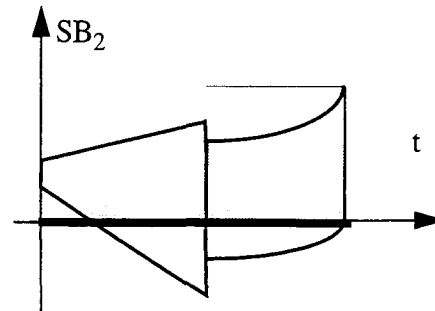


Figure 8.3b : Inconsistance

Nous proposons maintenant un algorithme permettant de détecter l'inconsistance d'un modèle.

8.3.1 Algorithme de détection d'inconsistance des modèles d'ordre 0

Cet algorithme est basé sur le lemme suivant.

Lemme 8.3: "Inconsistance d'un modèle d'ordre 0 pour une SBT"

Un modèle d'ordre 0 est inconsistant pour une suite de bandes temporelles SB si et seulement si

- le maximum de la fonction inférieure associée à SB est strictement positif
- ou
- le minimum de la fonction supérieure associée à SB est strictement négatif.

D'une manière formelle :

$$(C, T, (f(t))) \text{ inconsistant pour (SB)} \Leftrightarrow \max_{\theta \in T} F_i(\text{SB} ; \theta) > 0 \vee \min_{\theta \in T} F_s(\text{SB} ; \theta) < 0$$

Preuve :

$(C, T, (f(t)))$ inconsistant pour (SB)

$\Leftrightarrow \forall g(t) \text{ Rep SB}, \exists t_0 \in T, g(t_0) \neq 0$

$\Leftrightarrow \forall g(t) \text{ Rep SB}, \exists t_0 \in T, g(t_0) > 0 \vee g(t_0) < 0$

$\Leftrightarrow \exists t_0 \in T, F_i(\text{SB}; t_0) > 0 \vee F_s(\text{SB}; t_0) < 0$

(car $F_i(\text{SB}; t) \text{ Rep SB}$ et $F_s(\text{SB}; t) \text{ Rep SB}$ pour le sens \Rightarrow

et $\forall g(t) \text{ Rep SB}, F_i(\text{SB}; t_0) \leq g(t_0) \leq F_s(\text{SB}; t_0)$ pour le sens \Leftarrow)

$\Leftrightarrow \max_{\theta \in T} F_i(\text{SB}; \theta) > 0 \vee \min_{\theta \in T} F_s(\text{SB}; \theta) < 0$

(car $\max_{\theta \in T} F_i(\text{SB}; \theta) \geq F_i(\text{SB}; t_0)$ et $\min_{\theta \in T} F_s(\text{SB}; \theta) \leq F_s(\text{SB}; t_0)$ pour le sens \Rightarrow

et $\max_{\theta \in T} F_i(\text{SB}; \theta) = F_i(\text{SB}; t_1)$ et $\min_{\theta \in T} F_s(\text{SB}; \theta) = F_s(\text{SB}; t_2)$ pour le sens \Leftarrow)

□

Détecter l'inconsistance d'un modèle d'ordre 0 consiste donc à calculer les extrema des fonctions associées à une SBT, c'est à dire les extrema de fonctions polynomiales par morceaux.

D'où l'algorithme :

Algorithme de détection d'inconsistance d'un modèle d'ordre 0 pour la SBT $\{(t_1, t_2, p_{i1}, p_{s1}), \dots, (t_n, t_{n+1}, p_{in}, p_{sn})\}$
<p>inconsistance \leftarrow faux</p> <p>$k \leftarrow 1$</p> <p><u>faire</u></p> <p style="padding-left: 40px;">Calculer $\text{Sol}(p_{ik}(t), [t_k, t_{k+1}]) = \{x_1, \dots, x_p\}$</p> <p style="padding-left: 40px;"><u>Pour</u> $j = 1$ à p <u>faire</u></p> <p style="padding-left: 80px;"><u>Si</u> $p_{ik}(x_j) > 0$ <u>alors</u> inconsistance \leftarrow vrai</p> <p style="padding-left: 40px;"><u>Fin pour</u></p> <p style="padding-left: 40px;">Calculer $\text{Sol}(p_{sk}(t), [t_k, t_{k+1}]) = \{y_1, \dots, y_p\}$</p> <p style="padding-left: 40px;"><u>Pour</u> $j = 1$ à p <u>faire</u></p> <p style="padding-left: 80px;"><u>Si</u> $p_{sk}(y_j) < 0$ <u>alors</u> inconsistance \leftarrow vrai</p> <p style="padding-left: 40px;"><u>Fin pour</u></p> <p style="padding-left: 40px;"><u>Si</u> $p_{ik}(t_k) > 0$ <u>ou</u> $p_{ik}(t_{k+1}) > 0$ <u>ou</u> $p_{sk}(t_k) < 0$ <u>ou</u> $p_{sk}(t_{k+1}) < 0$</p> <p style="padding-left: 80px;"><u>alors</u> inconsistance \leftarrow vrai</p> <p style="padding-left: 40px;"><u>Fin si</u></p> <p style="padding-left: 40px;">$k \leftarrow k + 1$</p> <p><u>Jusqu'à</u> inconsistance = vrai <u>ou</u> $k = n + 1$</p>

La complexité de l'algorithme de détection d'inconsistance d'un modèle d'ordre 0 varie

linéairement en fonction du nombre de bandes temporelles de la SBT. Cet algorithme a des solutions analytiques pour des SBT de degré inférieur ou égal à 3.

Remarque :

Détecter l'inconsistance d'un modèle d'ordre 0 consiste à prouver l'existence d'un instant tel que toutes les fonctions représentées par SB soient différentes de 0. Avec l'algorithme proposé, pour démontrer l'existence de cet instant, nous le calculons. L'instant calculé est celui pour lequel une des fonctions associées à SB est extremum.

Nous présentons maintenant un exemple de détection d'inconsistance d'un modèle d'ordre 0.

8.3.2 Exemple

Soit le circuit illustré figure 8.4.

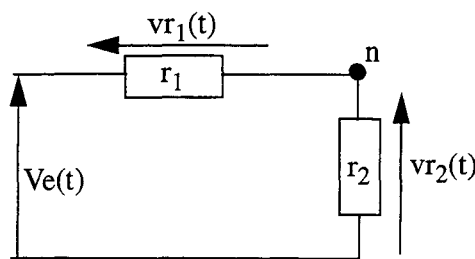


Figure 8.4 : Diviseur de tension

Ce circuit, composé de deux résistances et d'un générateur de tension représente un diviseur de tension. Lorsque tous les composants fonctionnent correctement, la tension aux bornes de la résistance r_2 est égale à la tension aux bornes du générateur multipliée par le rapport $\frac{R_2}{R_1 + R_2}$ ¹.

Le modèle représentant le comportement de ce système est

$$(\{r_1, r_2, n\}, \mathbf{R}, (v_e(t) - \frac{R_2}{R_1 + R_2} v_{r_2}(t)))$$

Soit l'ensemble des observations

$$OBS = \{(R_1, [10, 10]), (R_2, [10, 10]), (v_e(t), SB_e), (v_{r_2}(t), SB_2)\}$$

$$\text{avec } SB_e = \{(1, 3, 0.4 t, 0.4 t + 0.4)\} \text{ et } SB_2 = \{(1, 3, 0.2 t + 1.4, 0.2 t + 1.7)\}$$

L'ensemble des observations étant composé des observations des grandeurs élémentaires du système, nous proposons de détecter la C-inconsistance du modèle pour OBS.

N'ayant pas d'algorithme permettant de détecter la C-inconsistance, nous proposons de détecter l'inconsistance de ce modèle pour l'OBS-extension de la fonction de ce modèle, c'est-

1. Ce rapport étant toujours inférieur à 1, la tension aux bornes de r_2 est toujours inférieure à la tension aux bornes du générateur. D'où le nom de diviseur de tension.

à-dire détecter l'inconsistance du modèle $(\{r_1, r_2, n\}, \mathbf{R}, (v_e(t) - \frac{R_2}{R_1 + R_2} vr_2(t)))$ pour la SBT $SB_e - \frac{R_2}{R_1 + R_2} * SB_2 = SB_e - 0.5 * SB_2 = \{(1, 3, 0.3 t - 0.85, 0.3 t - 0.3)\}$. Cette SBT est illustrée figure 8.5.

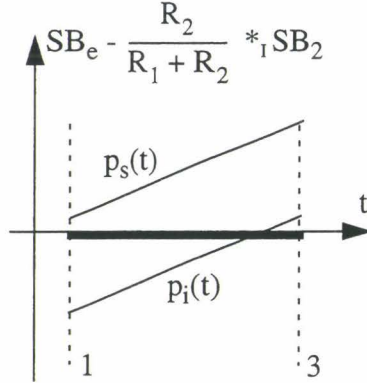


Figure 8.5 : Suite de bandes temporelles représentant la fonction du modèle

Comme la fonction identiquement nulle n'est pas représentée par $SB_e - 0.5 * SB_2$, le modèle est inconsistant pour cette SBT. Le modèle est donc C-inconsistant pour OBS car l'inconsistance implique la C-inconsistance (Théorème 7.2). Par conséquent, un des composants r_1, r_2 ou n est défectueux (i.e. $\{r_1, r_2, n\}$ est un ensemble de conflits).

L'algorithme de détection d'inconsistance consiste à calculer les extrema des fonctions associées à la suite de bandes temporelles. L'inconsistance est détectée à l'instant $t = 3$ car $p_i(3) = 0.05 > 0$.

Nous avons présenté un algorithme complet de détection d'inconsistance d'un modèle d'ordre 0. Toutefois, lorsque les OBS-extensions des fonctions des modèles introduisent de l'imprécision, cet algorithme n'est plus complet pour la C-inconsistance des modèles. Nous présentons maintenant quelques extensions de cet algorithme afin de se rapprocher de la C-inconsistance des modèles.

8.3.3 Extension à la C-consistance

Il n'est pas question dans ce paragraphe de présenter un algorithme complet de détection de la C-inconsistance des modèles d'ordre 0. Seules trois extensions de l'algorithme proposé précédemment sont présentées.

Lorsqu'un modèle est à la fois C-inconsistant pour un ensemble d'observation OBS et consistant pour les OBS-extensions des fonctions de ce modèle, la non détection d'un ensemble de conflits (car consistance) est due à l'imprécision introduite par les opérations sur les SBT. Nous proposons d'étudier trois cas d'imprécisions :

- celle due à la non continuité des SBT,
- celle due à la non conservation des dépendances entre les variables, c'est-à-dire celle due à la multiple apparition d'une grandeur dans l'expression de la fonction du modèle,
- celle due à la non conservation de certaines propriétés temporelles des fonctions, c'est-à-

dire celle due à l'apparition de grandeurs constantes au cours du temps dans l'expression de la fonction du modèle.

8.3.3.1 Non continuité des suites de bandes temporelles

Reprenons l'exemple présenté au paragraphe 8.3.2 (et illustré figure 8.4). Le modèle de ce

système est $(\{r_1, r_2, n\}, \mathbf{R}, (v_e(t) - \frac{R_2}{R_1 + R_2} vr_2(t)))$.

Soit l'ensemble des observations

$OBS = \{(R_1, [10, 10]), (R_2, [10, 10]), (v_e(t), SB_e), (vr_2(t), SB_2)\}$

avec $SB_e = \{(1, 2, 1, 2), (2, 3, 1, 3)\}$ et $SB_2 = \{(1, 2, 2, 6), (2, 3, 5, 6)\}$

Comme le montre la figure 8.6, le modèle est consistant pour la suite de bandes temporelles $SB = SB_e - (R_2 / (R_1 + R_2)) SB_2 = \{(1, 2, -2, 1), (2, 3, -2, 0.5)\}$.

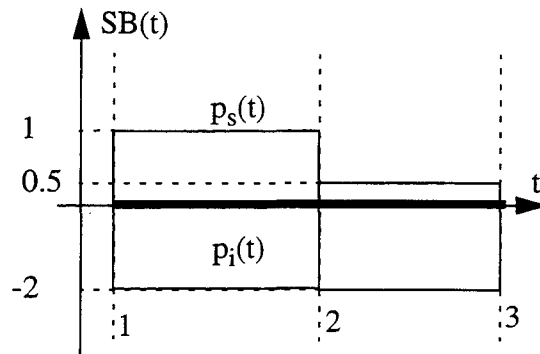


Figure 8.6 : SBT représentant la fonction du modèle

Montrons maintenant que le modèle $(\{r_1, r_2, n\}, \mathbf{R}, (v_e(t) - \frac{R_2}{R_1 + R_2} vr_2(t)))$ est C-inconsistant pour OBS.

Il découle directement de la propriété de représentation et du lemme 8.1 que :

$$v_e(t) \text{ Rep } SB_e \Rightarrow v_e(2) \leq F_s(SB_e ; 2) = \min(2, 3) = 2$$

et

$$vr_2(t) \text{ Rep } SB_2 \Rightarrow vr_2(2) \geq F_i(SB_2 ; 2) = \max(2, 5) = 5$$

$$\text{D'où } v_e(2) - \frac{R_2}{R_1 + R_2} vr_2(2) \leq 2 - 5/2 = -0.5 < 0$$

Comme la fonction du modèle est strictement négative à l'instant 2, ce modèle est C-inconsistant pour OBS.

D'une manière générale, le problème de la non continuité des SBT est considéré dans la définition des fonctions associées aux SBT, mais pas dans celle des opérations sur les SBT. On peut montrer qu'aux instants de discontinuité des SBT, les fonctions associées à l'addition (resp. la multiplication) de deux SBT ne sont pas toujours égales à l'addition (resp. la multiplication) des fonctions associées à ces SBT .

Par exemple,

$$F_s(SB_e ; 2) - 0.5 F_i(SB_2 ; 2) = -0.5$$

et

$$F_s(SB_e - 0.5 SB_2 ; 2) = +0.5$$

La C-inconsistance a été détectée en remarquant que

$$v_e(2) - \frac{R_2}{R_1 + R_2} vr_2(2) \leq F_s(SB_e ; 2) - 0.5 F_i(SB_2 ; 2) < 0,$$

alors que la C-inconsistance ne peut être détectée avec $F_s(SB_e - 0.5 SB_2 ; 2)$ car

$$F_s(SB_e - 0.5 SB_2 ; 2) > 0.$$

Nous proposons une méthode permettant de prendre en compte la discontinuité des SBT :

Elle consiste à calculer un encadrement de la fonction du modèle aux instants de discontinuité des SBT. Cet encadrement est calculé à partir des valeurs aux instants de discontinuité des fonctions associées aux différentes SBT.

Par exemple, $v_e(2) - \frac{R_2}{R_1 + R_2} vr_2(2) \in [F_i(SB_e ; 2), F_s(SB_e ; 2)] - 0.5 [F_i(SB_2 ; 2), F_s(SB_2 ; 2)]$

Nous nous intéressons maintenant aux expressions des fonctions des modèles.

8.3.3.2 Expression des fonctions des modèles

Reprenons l'exemple présenté au paragraphe 8.3.2 (et illustré figure 8.4). Le modèle de ce

système est $(\{r_1, r_2, n\}, \mathbf{R}, (v_e(t) - \frac{R_2}{R_1 + R_2} vr_2(t)))$.

Soit l'ensemble des observations

$$OBS = \{(R_1, [9, 11]), (R_2, [9, 11]), (v_e(t), SB_e), (vr_2(t), SB_2)\}$$

avec $SB_e = \{(1, 3, 2, 4)\}$ et $SB_2 = \{(1, 3, 2.5, 3.5)\}$

Etudions la consistance de ce modèle pour l'OBS-extension aux SBT de la fonction du

modèle, c'est à dire pour la SBT $SB_e - \frac{[9, 11]}{[9, 11] + [9, 11]} SB_2 = SB_e - [0.41, 0.61] SB_2$.

Comme le montre la figure 8.7, ce modèle est consistant pour cette SBT.

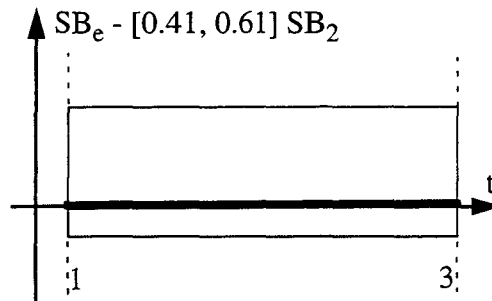


Figure 8.7 : SBT représentant la fonction du modèle

La grandeur R_2 apparaissant deux fois dans l'expression de la fonction du modèle (i.e $v_e(t) -$

$\frac{R_2}{R_1 + R_2} vr_2(t)$), l'extension aux SBT de la fonction du modèle (i.e $SB_e - \frac{[9, 11]}{[9, 11] + [9, 11]} SB_2$) introduit de l'imprécision. Par conséquent, assurer la consistance du modèle pour cette SBT ne permet pas d'assurer que le modèle est C-consistant pour OBS, c'est-à-dire que tous les ensembles de conflit sont détectés.

Afin de réduire l'imprécision introduite par les opérations sur les SBT, nous proposons de détecter l'inconsistance d'un modèle équivalent.

On remarque que les modèles

$$(\{r_1, r_2, n\}, \mathbf{R}, ((\frac{R_1}{R_2} + 1) v_e(t) - vr_2(t)))$$

$$\text{et } (\{r_1, r_2, n\}, \mathbf{R}, (v_e(t) - \frac{R_2}{R_1 + R_2} vr_2(t)))$$

sont équivalents car $R_2 \neq 0$ (i.e. $0 \notin [9, 11]$).

Vérifions l'inconsistance du modèle $(\{r_1, r_2, n\}, \mathbf{R}, ((\frac{R_1}{R_2} + 1) v_e(t) - vr_2(t)))$ pour l'OBS-extension de sa fonction, c'est à dire pour $(\frac{[9, 11]}{[9, 11]} + 1)SB_e - SB_2 = [1.82, 2.22] SB_e - SB_2$.

Comme le montre la figure 8.8, ce modèle est inconsistant pour cette SBT.

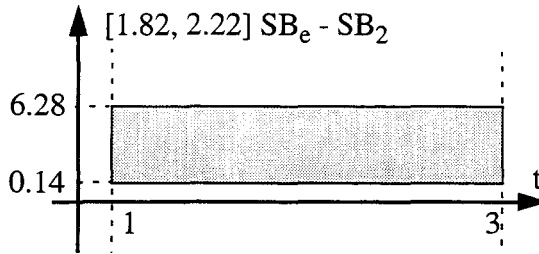


Figure 8.8 : Extension aux SBT de $(\frac{R_1}{R_2} + 1) v_e(t) - vr_2(t)$

Comme le calcul sur les intervalles, le calcul sur les SBT ne tient pas compte des dépendances entre les variables. Ainsi, la complétude de l'algorithme de détection d'inconsistance dépend de l'expression de la fonction du modèle. La grandeur R_2 n'apparaissant plus qu'une seule fois dans l'expression de la fonction du modèle équivalent, l'OBS-extension de cette fonction est une SBT de "meilleure qualité" (propriété 5.3). Cette SBT permet de détecter l'inconsistance du modèle. D'une manière générale, pour éviter le problème de la non conservation des dépendances entre les variables, il faut et il suffit de déterminer un modèle équivalent dont l'expression de la fonction ne fait intervenir qu'une seule fois les grandeurs. Pour déterminer ce modèle équivalent, les opérations autorisées sont les suivantes :

- la factorisation par une grandeur,
- la multiplication ou la division de la fonction du modèle par une grandeur constante non nulle (i.e. l'intervalle encadrant cette grandeur constante ne contient pas 0),
- la multiplication de la fonction du modèle par une grandeur dont la valeur est toujours différente de 0 (i.e. la SBT représentant cette fonction est positive ou négative).

On peut montrer que si un modèle à occurrences simples est consistant pour l'OBS-extension

de sa fonction, alors tous les modèles équivalents sont consistants pour les OBS-extensions de leurs fonctions. Ceci découle directement de la propriété sur les occurrences simples.

Toutefois, cette méthode a certaines limites :

- Pour un modèle donné, il n'existe pas toujours de modèle équivalent dont l'expression de la fonction ne fait intervenir qu'une seule fois chaque constante et chaque grandeur. Par exemple, il est impossible de déterminer un tel modèle à partir du modèle $(C, T, ((r_1+r_2) v_1(t) + (r_2+r_3) v_2(t) + v_3(t)))$.
- L'addition et la multiplication d'une SBT par un intervalle introduisent de l'imprécision. Par conséquent, si des grandeurs constantes apparaissent dans l'expression de la fonction d'un modèle, alors vérifier la consistance de ce modèle pour l'OBS-extension de sa fonction ne permet pas d'assurer la C-consistance de ce modèle pour OBS, c'est à dire ne permet pas d'assurer que tous les ensembles de conflits sont détectés. C'est cette propriété que nous étudions maintenant.

8.3.3.3 Imprécision introduite par les constantes

Reprenons l'exemple présenté au paragraphe 8.3.2 (et illustré figure 8.4). Le modèle de ce système est $(\{r_1, r_2, n\}, \mathbf{R}, (v_e(t) - \frac{R_2}{R_1 + R_2} v_{r_2}(t)))$.

Soit l'ensemble des observations

$$OBS = \{(R_1, [9, 11]), (R_2, [9, 11]), (v_e(t), SB_e), (v_{r_2}(t), SB_2)\}$$

$$\text{avec } SB_e = \{(1, 3, 0.4 t, 0.4 t + 0.4)\} \text{ et } SB_2 = \{(1, 3, 0.2 t + 1.4, 0.2 t + 1.7)\}$$

Pour éviter le problème de la non conservation des dépendances entre les variables, nous proposons de vérifier la consistance du modèle $(\{r_1, r_2, n\}, \mathbf{R}, ((\frac{R_1}{R_2} + 1) v_e(t) - v_{r_2}(t)))$ pour l'OBS-extension de sa fonction. Comme le montre la figure 8.9, ce modèle est consistant pour $(\frac{[9, 11]}{[9, 11]} + 1) SB_e - SB_2$. De plus, la fonction $(\frac{R_1}{R_2} + 1) v_e(t) - v_{r_2}(t)$ ne faisant intervenir qu'une seule fois chaque grandeur, aucun ensemble de conflits ne peut être détecté en vérifiant la consistance de ce modèle.

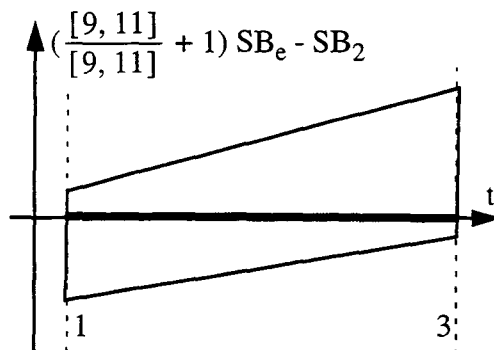


Figure 8.9 : Extension aux SBT de la fonction $(\frac{R_1}{R_2} + 1) v_e(t) - v_{r_2}(t)$

Chapitre 8: Famille d'algorithmes de détection d'inconsistance

Montrons maintenant que le modèle $(\{r_1, r_2, n\}, \mathbf{R}, (v_e(t) - \frac{R_2}{R_1 + R_2} vr_2(t)))$ est C-inconsistant pour OBS.

D'après l'équation du modèle, on a $\frac{R_2}{R_1 + R_2} = \frac{v_e(t)}{vr_2(t)}$.

Il découle de la propriété de représentation d'une fonction par une SBT que :

$$v_e(t) \text{ Rep } SB_e \Rightarrow v_e(1) \in SB_e(1) = [0.4, 0.8] \text{ et } v_e(3) \in SB_e(3) = [1.2, 1.6]$$

et

$$vr_2(t) \text{ Rep } SB_2 \Rightarrow vr_2(1) \in SB_2(1) = [1.6, 1.9] \text{ et } vr_2(3) \in SB_2(3) = [2, 2.3]$$

$$\text{D'où : } \frac{R_2}{R_1 + R_2} = \frac{v_e(1)}{vr_2(1)} \in [0.4, 0.8] / [1.6, 1.9] = [0.21, 0.50]$$

$$\text{et } \frac{R_2}{R_1 + R_2} = \frac{v_e(3)}{vr_2(3)} \in [1.2, 1.6] / [2, 2.3] = [0.52, 0.80]$$

Comme $\frac{R_2}{R_1 + R_2}$ ne peut pas appartenir aux deux intervalles $[0.21, 0.50]$ et $[0.52, 0.80]$, le

modèle est C-inconsistant pour OBS. La non intersection des intervalles signifie que la fonction $f(t) = \frac{R_2}{R_1 + R_2}$ ne peut pas être constante au cours du temps pendant l'intervalle

temporelle $[1, 3]$. Comme les opérations sur les SBT ne conservent pas les propriétés temporelles des fonctions, dès lors que le calcul de l'OBS-extension de la fonction du modèle

est réalisé, on perd la possibilité de montrer que la fonction $f(t) = \frac{R_2}{R_1 + R_2}$ ne peut pas être

constante au cours du temps. C'est pourquoi, il est impossible de déterminer un modèle qui est inconsistant pour l'OBS-extension de sa fonction. Cette propriété peut s'exprimer autrement. En effet, on peut montrer qu'aucun instant pour lequel l'équation du modèle est différente de 0 ne peut être calculé. Seules l'existence de cet instant et son appartenance à l'intervalle temporelle $[1, 3]$ peuvent être déduits. Il est donc logique dans ce cas de ne pouvoir détecter l'inconsistance du modèle car la détection d'inconsistance nécessite de pouvoir calculer un instant pour lequel l'équation du modèle est différente de 0. D'une manière générale, la C-inconsistance d'un modèle pour un ensemble d'observation OBS peut être détectée en vérifiant l'inconsistance de ce modèle pour l'OBS-extension de la fonction du modèle uniquement lorsqu'il est possible de calculer un instant pour lequel l'équation du modèle est différente de 0.

Nous proposons maintenant une méthode permettant de considérer les propriétés temporelles des grandeurs. Cette méthode consiste à calculer les encadrements des grandeurs constantes à partir des encadrements des valeurs à différents instants des grandeurs variant au cours du temps. Ceci revient à vérifier la consistance du système de contraintes contenant

- les intervalles encadrant les grandeurs constantes,
- les intervalles encadrant les valeurs à différents instants des grandeurs variant au cours du temps,
- les équations du modèle correspondant aux différents instants.

Par exemple, soit le modèle $(\{r_1, r_2, n\}, \mathbf{R}, (v_e(t) - \frac{R_2}{R_1 + R_2} vr_2(t)))$

et l'ensemble des observations

OBS = $\{ (v_e(t), \{(1, 3, 0.4 t, 0.4 t + 0.4)\}), (vr_2(t), \{(1, 3, 0.2 t + 1.4, 0.2 t + 1.7)\}), (R_1, [9, 11]), (R_2, [9, 11]) \}$.

Pour vérifier la C-consistance de ce modèle pour OBS, nous proposons de vérifier la consistance du système de contraintes :

$$\{R_1 \in [9, 11], R_2 \in [9, 11],$$

$$v_e(t_1) \in [0.4 t_1, 0.4 t_1 + 0.4], v_e(t_2) \in [0.4 t_2, 0.4 t_2 + 0.4], \dots,$$

$$vr_2(t_1) \in [0.2 t_1 + 1.4, 0.2 t_1 + 1.7], vr_2(t_2) \in [0.2 t_2 + 1.4, 0.2 t_2 + 1.7], \dots,$$

$$v_e(t_1) - \frac{R_2}{R_1 + R_2} vr_2(t_1) = 0, v_e(t_2) - \frac{R_2}{R_1 + R_2} vr_2(t_2) = 0, \dots \}$$

Cette méthode pose essentiellement deux problèmes :

- Le premier concerne le choix des instants t_1, t_2, \dots . Il semble cohérent de choisir les instants définies par la procédure de consistance car ces instants correspondent aux bornes inférieure et supérieure de l'intervalle encadrant l'équation du modèle à tout instant. Pour l'exemple proposé, les instants définis par la procédure de consistance sont les instants 1 et 3. Ces instants ont permis de trouver l'inconsistance du système de contraintes.
- Le deuxième concerne la résolution de ce système de contraintes. Nous avons proposé de calculer un encadrement d'une fonction constante au cours du temps à partir de l'encadrement de la valeur à différents instants d'une fonction variant au cours du temps. Toutefois, il n'est pas toujours possible d'isoler une fonction constante au cours du temps. Une autre solution consiste à utiliser un algorithme de propagation d'intervalles [Lhomme, 94], algorithme permettant d'obtenir de meilleurs encadrements pour chaque grandeur. Pour l'exemple proposé, cet algorithme consiste à :

Calculer un encadrement de R_1 à partir des encadrements des autres grandeurs par l'intermédiaire des équations $R_1 = R_2 (vr_2(1) / v_e(1) - 1) = R_2 (vr_2(3) / v_e(3) - 1)$

$$R_1 \in [9, 11] ([1.6, 1.9] / [0.4, 0.8] - 1) = [9, 41.25] \text{ (à l'instant } t = 1)$$

$$\text{et } R_1 \in [9, 11] ([2, 2.3] / [1.2, 1.6] - 1) = [2.25, 10.12] \text{ (à l'instant } t = 3)$$

$$\text{et } R_1 \in [9, 11] \text{ (valeur donnée dans OBS)}$$

D'où, $R_1 \in [9, 10.12]$ (intersection des trois intervalles calculés précédemment).

Calculer un encadrement de R_2 à partir des encadrements des autres grandeurs par l'intermédiaire de l'équation $R_2 = R_1 * v_e(1) / (vr_2(1) - v_e(1)) = R_1 * v_e(3) / (vr_2(3) - v_e(3))$
Après intersection des intervalles, on trouve $R_2 \in [9.81, 10.12]$.

Recalculer un encadrement de R_1 car un nouvel encadrement de R_2 a été trouvé.

$$R_1 \in [9.81, 10.12].$$

Recalculer un encadrement de R_2 avec la nouvelle valeur de R_1

$$R_2 \in [10.69, 10.12].$$

L'inconsistance du système de contrainte est détectée car $[10.69, 10.12]$ n'est pas un

intervalle.

Il est clair que le coût d'un algorithme de propagation d'intervalles est important. D'autres méthodes permettant de détecter la consistance d'un système de contraintes peuvent être envisagées. Toutefois, l'avantage de la propagation est de toujours fournir une information utile : soit la C-inconsistance du modèle est détectée, soit des encadrements de meilleure qualité pour les grandeurs constantes sont obtenues. Ces encadrements peuvent être très utiles dans le cas où les encadrements initiaux des grandeurs constantes sont de mauvaise qualité. Par exemple, ceci est le cas du gain d'un transistor.

8.3.3.4 Conclusion

Détecter la C-inconsistance d'un modèle consiste à montrer, à partir des observations, l'existence d'un instant t_0 tel que l'équation du modèle soit différente de 0. Nous distinguons deux cas.

- Soit, à partir des observations, l'instant t_0 est calculable. Dans ce cas, il est possible de calculer une SBT encadrant la fonction du modèle pour laquelle le modèle est inconsistant. Limiter l'imprécision consiste alors à améliorer le calcul de la SBT représentant la fonction du modèle. Cette amélioration peut provenir

- pour la discontinuité des SBT :
 - d'un recalcul des fonctions aux instants de discontinuité,
- pour la multiple apparition d'une grandeur :
 - d'un calcul d'un modèle dont la fonction est plus adaptée au calcul sur les SBT.

- Soit, à partir des observations, l'instant t_0 est incalculable, seul l'existence de cet instant peut être démontrée. Dans ce cas, il est impossible de calculer une SBT pour laquelle le modèle est inconsistant. Pour détecter la C-inconsistance du modèle pour les observations, il est nécessaire de faire appel à d'autres algorithmes. Ce cas correspond à l'apparition de grandeur(s) constante(s) dans l'expression de la fonction du modèle.

Nous nous intéressons maintenant à la consistance des modèles d'ordre 1.

8.4 Algorithme de consistance des modèles d'ordre 1

Nous présentons d'abord un exemple dont le but est d'expliquer pourquoi les SBT permettent de diagnostiquer des équations différentielles d'ordre 1 sans observer les dérivées des grandeurs. Nous présentons ensuite les problèmes à résoudre pour établir un algorithme de détection d'inconsistance. La dernière partie est consacrée à l'algorithme que nous proposons.

8.4.1 Exemple

Soit le système illustré figure 8.10.

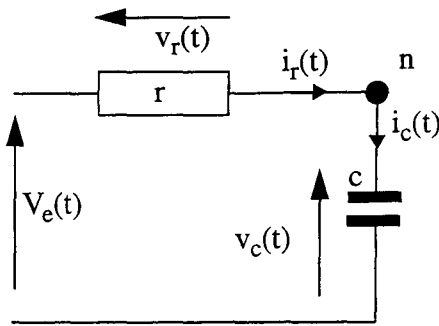


Figure 8.10 : Circuit RC

Le comportement de ce circuit est décrit par l'équation $v_r(t) = RC v_c'(t)$

où R et C sont respectivement les valeurs de la résistance r et de la capacité c .

Afin de simplifier cet exemple, nous supposons que ces valeurs sont connues. Nous prendrons $RC = 1$.

Lorsque ce système fonctionne correctement, une montée rapide de la tension aux bornes du générateur provoque une montée progressive de la tension aux bornes de la capacité et une diminution progressive de la tension aux bornes de la résistance. La forme de ces signaux est illustrée figure 8.11.

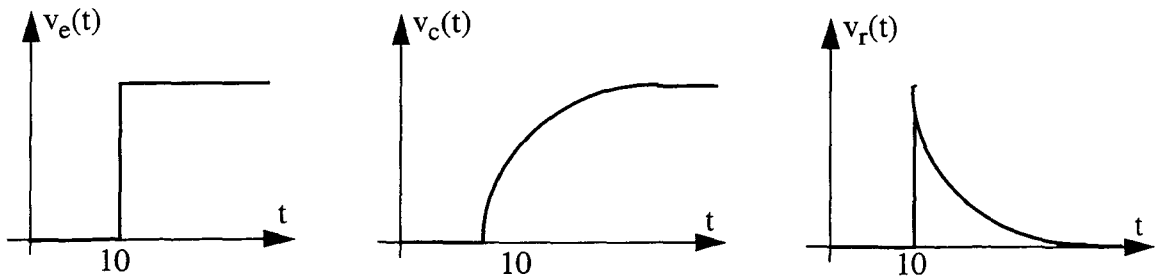


Figure 8.11 : Signaux du circuit RC fonctionnant correctement

Supposons la coupure de la capacité. Dans ce cas de figure, aucun courant ne traverse le circuit ($i_c(t) = i_r(t) = 0$). Le courant traversant la résistance étant nul, la tension aux bornes de ce composant est nulle. Par conséquent, la tension aux bornes de la capacité est identique à celle aux bornes du générateur. Les suites de bandes temporelles représentant ces tensions sont illustrées figure 8.12.

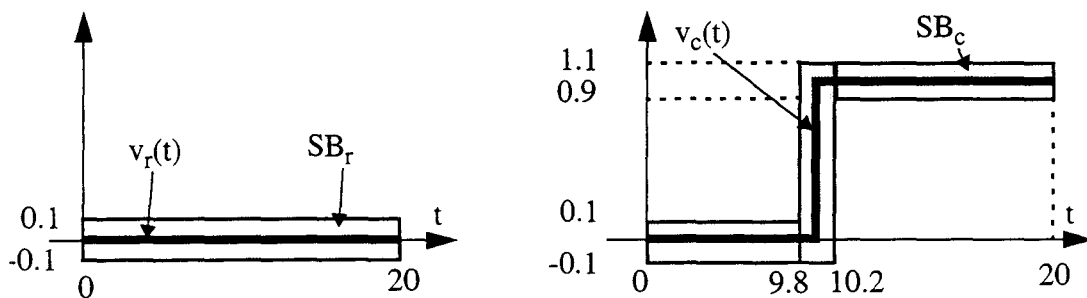


Figure 8.12 : Signaux du circuit RC en panne

L'ensemble des observations correspondant est :

$$OBS = \{(v_r(t), SB_r), (v_c(t), SB_c)\}$$

Chapitre 8: Famille d'algorithmes de détection d'inconsistance

où $SB_r = \{(0, 20, -0.1, 0.1)\}$ et $SB_c = \{(0, 9.8, -0.1, 0.1), (9.8, 10.2, -0.1, 1.1), (10.2, 20, 0.9, 1.1)\}$

Notre but est de montrer que ces observations sont suffisantes pour détecter que le circuit RC est en panne. Raisonnons par l'absurde et supposons que ce circuit fonctionne correctement. On a alors $v_r(t) = v_c'(t)$. Il découle de cette relation que si le circuit fonctionne correctement alors les grandeurs $v_c(t)$ et $v_c'(t)$ sont respectivement représentées par les suites de bandes temporelles SB_c et SB_r (car $v_c'(t) = v_r(t)$ et $v_r(t)$ Rep SB_r). Afin de détecter la panne, il est nécessaire d'établir les liens existant entre les encadrements d'une fonction et de la dérivée de cette fonction. Nous proposons deux liens.

Le premier lien est obtenu en encadrant **la dérivée d'une fonction à partir de l'encadrement de la fonction**. Ce lien est établi à partir du théorème des accroissements finis :

$$\exists c \in]t_1, t_2[\text{ tel que } v_c'(c) = \frac{v_c(t_2) - v_c(t_1)}{t_2 - t_1}$$

Ne connaissant pas l'instant c , seul un encadrement de $v_c'(c)$ peut être obtenu. Soit

$$\min_{\theta \in [t_1, t_2]} v_c'(\theta) \leq v_c'(c) \leq \max_{\theta \in [t_1, t_2]} v_c'(\theta)$$

Pour détecter la panne, nous proposons de choisir les instants $t_1 = 9.8$ et $t_2 = 10.2$. On obtient :

$$\begin{aligned} v_c(t) \text{ Rep } SB_c &\Rightarrow v_c(9.8) \in [F_i(SB_c ; 9.8), F_s(SB_c ; 9.8)] = [-0.1, 0.1] \\ &\wedge v_c(10.2) \in [F_i(SB_c ; 10.2), F_s(SB_c ; 10.2)] = [0.9, 1.1] \\ \Rightarrow v_c'(c) &= \frac{v_c(t_2) - v_c(t_1)}{t_2 - t_1} \in [2, 3] \end{aligned}$$

$$v_c'(t) \text{ Rep } SB_r \Rightarrow v_c'(c) \in \left[\min_{\theta \in [t_1, t_2]} v_c'(\theta), \max_{\theta \in [t_1, t_2]} v_c'(\theta) \right] = [-0.1, 0.1]$$

Ceci est absurde car $v_c'(c)$ ne peut appartenir aux intervalles $[2, 3]$ et $[-0.1, 0.1]$. Donc, le système est en panne.

Graphiquement, comme $v_c(t)$ est représentée par SB_c , il existe un instant c appartenant à $[9.8, 10.2]$ tel que $v_c'(c)$ est supérieur à la pente de la droite (D) passant par les points A et B de coordonnées $(9.8, 0.1)$ et $(10.2, 0.9)$ (cf. figure 8.13). Or, ceci est absurde car la pente de cette droite est 2, alors que l'encadrement de $v_c'(t)$ par SB_r nous assure que cette pente est inférieure à 0.1.

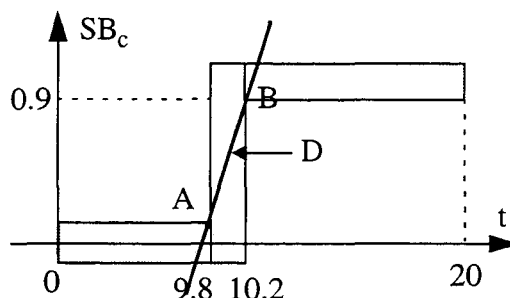


Figure 8.13 : Encadrement de la dérivée en un instant

A partir de la SBT représentant une fonction, il est possible d'encadrer la valeur de la dérivée

de cette fonction en un instant. Toutefois, cet instant n'est pas connu.

Le deuxième lien entre les encadrements d'une fonction et de sa dérivée est obtenu en encadrant la **dérivée moyenne d'une fonction à partir de l'encadrement de la fonction**. Ce lien est établi à partir de l'opération d'intégration :

$$\int_{t_1}^{t_2} v_c'(x) dx = v_c(t_2) - v_c(t_1)$$

L'information fournie par l'opération d'intégration est de meilleure qualité que celle fournie par le théorème des accroissements finis. En effet, les propriétés de l'intégration nous assurent que $(t_2 - t_1) \min_{\theta \in [t_1, t_2]} v_c'(\theta) \leq \int_{t_1}^{t_2} v_c'(x) dx \leq (t_2 - t_1) \max_{\theta \in [t_1, t_2]} v_c'(\theta)$ ¹.

L'avantage des SBT est de permettre l'utilisation de cette information car l'intégration d'une SBT est définie. En choisissant les mêmes instants que précédemment, il est donc logique que la panne du système soit détectée. Ainsi :

$$v_c(t) \text{ Rep } SB_c \Rightarrow v_c(10.2) - v_c(9.8) \in [F_i(SB_c; 10.2), F_s(SB_c; 10.2)] - [F_i(SB_c; 9.8), F_s(SB_c; 9.8)] = [0.8, 1.2]$$

et

$$v_c'(t) \text{ Rep } SB_r \Rightarrow \int_{9.8}^{10.2} v_c'(x) dx = v_c(10.2) - v_c(9.8) \in \int_{9.8}^{10.2} SB_r = [-0.04, 0.04]$$

Ceci est absurde car $v_c(10.2) - v_c(9.8)$ ne peut appartenir aux deux intervalles $[0.8, 1.2]$ et $[-0.04, 0.04]$. Donc, la panne du système est détectée.

Graphiquement, nous avons calculé la dérivée moyenne de $v_c(t)$ sur l'intervalle $[9.8, 10.2]$ (i.e.

$(\int_{9.8}^{10.2} v_c'(x) dx) / (10.2 - 9.8)$). La SBT représentant $v_c(t)$ (i.e. SB_c) nous assure que la dérivée moyenne est supérieure à la pente de la droite D illustrée figure 8.13, c'est-à-dire 2. La SBT représentant $v_c'(t)$ (i.e. SB_r) nous assure que la dérivée moyenne est inférieure à la surface S, c'est-à-dire 0.1 (cf. figure 8.14). Comme ceci est absurde, le circuit est en panne.

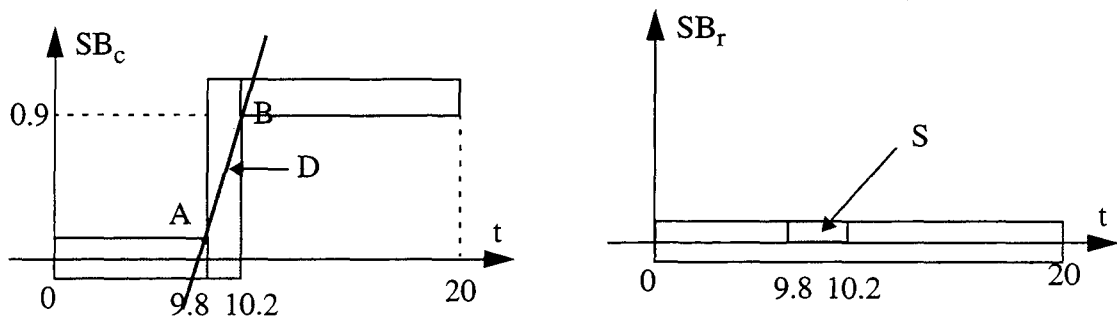


Figure 8.14 : Explication de la dérivée moyenne

En conclusion, la modélisation par des SBT permet de diagnostiquer un circuit, dont le comportement est décrit par une équation différentielle d'ordre 1, sans nécessiter les observations des dérivées des grandeurs. **Ce diagnostic est possible car le lien entre les**

1. Cette relation nous assure aussi que les deux informations sont équivalentes lorsque la fonction $v_c(t)$ est constante au cours du temps. Etant donnée la modélisation par des SBT, les deux informations sont identiques lorsque les fonctions associées à la SBT représentant $v_c'(t)$ sont constantes.

encadrements d'une fonction et de sa dérivée est conservé.

Toutefois, l'encadrement de la dérivée moyenne d'une grandeur dépend de l'intervalle pour lequel ce calcul est réalisé. Par conséquent, la détection de la panne dépend de l'intervalle choisi (i.e. des instants d'intégration). Par exemple, en choisissant les instants $t_1 = 0$ et $t_2 = 10.2$, la panne n'est pas détectée car les encadrements de la dérivée moyenne s'intersectent. En effet,

$$v_c(t) \text{ Rep } SB_c \Rightarrow v_c(10.2) - v_c(0) \in [F_i(SB_c ; 10.2), F_s(SB_c ; 10.2)] - SB_c(0) = [0.8, 1.2]$$

et

$$v_c'(t) \text{ Rep } SB_r \Rightarrow \int_0^{10.2} v_c'(x) dx = v_c(10.2) - v_c(0) \in \int_0^{10.2} SB_r = [-1.02, 1.02]$$

Nous proposons maintenant une explication formelle de ce phénomène en utilisant les notations précédemment définies.

8.4.2 Explication

Soient le modèle $(C, T, (f_0(t), f_1(t)))$

et l'ensemble des observations $OBS = \{(f_0(t), SB_0), (f_1(t), SB_1)\}$

Montrons que tout modèle d'ordre 1 est équivalent à une infinité de modèles d'ordre 0 dont les équations sont obtenues par intégration. Etant donné un instant θ de T , on a :

$$\forall t \in T, g_0(t) + g_1'(t) = 0 \Leftrightarrow \forall t \in T, \int_{\theta}^t g_0(x) dx + g_1(t) - g_1(\theta) = 0$$

D'où

$$(C, T, (f_0(t), f_1(t))) \Leftrightarrow (C, T, (\int_{\theta}^t f_0(x) dx + f_1(t) - f_1(\theta)))$$

Il découle de cette équivalence que détecter l'inconsistance d'un modèle d'ordre 1 est équivalent à détecter la C-inconsistance d'un modèle d'ordre 0. En effet :

$$(C, T, (f_0(t), f_1(t))) \text{ inconsistant pour } (SB_0, SB_1)$$

$$\Leftrightarrow (C, T, (\int_{\theta}^t f_0(x) dx + f_1(t) - f_1(\theta))) \text{ C-inconsistant pour } \{(f_0(t), SB_0), (f_1(t), SB_1)\}$$

Aucune procédure de détection de la C-inconsistance d'un modèle d'ordre 0 n'a été établie. Toutefois, ce problème peut être ramené à un problème de détection d'inconsistance car l'inconsistance implique la C-inconsistance (théorème 7.2) :

$$(C, T, (\int_{\theta}^t f_0(x) dx + f_1(t) - f_1(\theta))) \text{ inconsistant pour } (\int_{\theta} SB_0 + SB_1 - SB_1(\theta))$$

$$\Rightarrow (C, T, (f_0(t), f_1(t))) \text{ inconsistant pour } (SB_0, SB_1)$$

Cette implication permet d'assurer que

la vérification de l'inconsistance d'un modèle d'ordre 0 peut permettre la détection de l'inconsistance d'un modèle d'ordre 1.

Par contre, l'implication inverse n'est pas vraie car l'extension aux SBT de la fonction du

modèle (i.e. $\int_{\theta} SB_0 + SB_1 - SB_1(\theta)$) introduit de l'imprécision car cette fonction fait intervenir deux opérations introduisant de l'imprécision :

- l'intégration d'une SBT,
- l'addition d'un intervalle et d'une SBT (i.e. $SB_1(\theta)$ est un intervalle).

D'où

Vérifier l'inconsistance d'un (et un seul) modèle d'ordre 0 est insuffisant pour assurer la consistance d'un modèle d'ordre 1.

Nous proposons d'illustrer ces deux propositions en reprenant l'exemple du paragraphe précédent. Soient les SBT SB_0 et SB_1 (cf figure 8.15) représentant respectivement les fonctions $f_0(t)$ et $f_1(t)$ (ces SBT correspondent aux SBT représentant les tensions du circuit RC en panne).

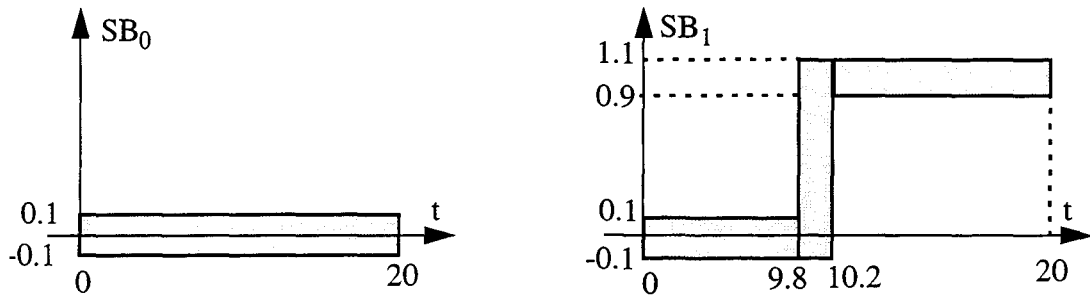


Figure 8.15 : Exemple de SBT

Pour vérifier la consistance du modèle d'ordre 1 ($C, T, (f_0(t), f_1(t))$), nous proposons de vérifier la consistance des modèles d'ordre 0 ($C, T, (\int_{\theta} f_0(x)dx + f_1(t) - f_1(\theta))$) pour les instants $\theta = 0$ et $\theta = 9.8$. Les OBS-extensions des fonctions de ces modèles sont illustrées figure 8.16.

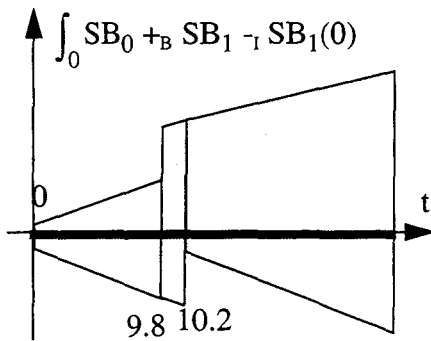


Figure 8.16a : $\theta = 0$

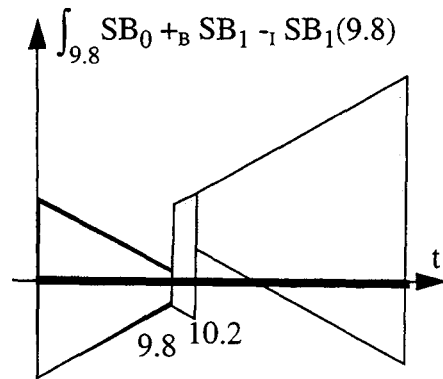


Figure 8.16b : $\theta = 9.8$

Le modèle d'ordre 0 obtenu pour $\theta = 9.8$ est inconsistant : ceci permet de détecter l'inconsistance du modèle d'ordre 1 (1^{ère} proposition).

Le modèle d'ordre 0 obtenu pour $\theta = 0$ est consistant : vérifier uniquement la consistance de ce modèle aurait engendré la non détection de l'inconsistance du modèle d'ordre 1 (2^{ème} proposition).

Les algorithmes de détection d'inconsistance des modèles d'ordre 1 proposés dans le paragraphe suivant sont directement issus de ces remarques :

- les algorithmes sont basés sur l'intégration de l'équation du modèle,
- ils doivent contenir une procédure de choix de l'instant d'intégration.

8.4.3 Choix d'un algorithme

Trois algorithmes sont proposés. Chacun correspond à une procédure de choix des instants d'intégration différente :

1 - La première possibilité consiste à déterminer une heuristique permettant de choisir un ou plusieurs instants d'intégration, puis vérifier la consistance des modèles d'ordre 0 correspondant à ces différents instants. Cette voie de recherche n'a pas été investiguée car il semble difficile de déterminer des critères simples et surtout pertinents pour choisir les différents instants d'intégration.

2 - La deuxième possibilité consiste à ne pas choisir d'instant d'intégration, c'est-à-dire à vérifier la consistance de tous les modèles d'ordre 0 équivalents au modèle d'ordre 1. Un algorithme reposant sur ce principe et fonctionnant pour les SBT d'ordre 1 a été proposé dans [Loiez et al, 96b]. Le principe est le suivant.

Détecter la consistance du modèle d'ordre 1 $(C, T, (f_0(t), f_1(t)))$ pour $\{(f_0(t), SB_0), (f_1(t), SB_1)\}$ consiste à :

- Intégrer l'équation du modèle. Soit :

$$\forall t_1, t_2, \int_{t_1}^{t_2} f_0(x)dx + f_1(t_2) - f_1(t_1) = 0$$

- Encadrer la fonction ainsi obtenue par les fonctions associées aux SBT. Soit :

$$\forall t_1 < t_2, \int_{t_1}^{t_2} f_0(x)dx + f_1(t_2) - f_1(t_1) \geq P_{i0}(t_2) - P_{i0}(t_1) + p_{i1}(t_2) - p_{s1}(t_1)$$

$$\text{et } \forall t_1 < t_2, \int_{t_1}^{t_2} f_0(x)dx + f_1(t_2) - f_1(t_1) \leq P_{s0}(t_2) - P_{s0}(t_1) + p_{s1}(t_2) - p_{i1}(t_1)$$

où $P_{i0}(t)$, $P_{s0}(t)$, $p_{i1}(t)$ et $p_{s1}(t)$ sont les fonctions associées aux SBT représentant $\int_{t_i} SB_0$ et SB_1 (t_i étant le début de l'extension temporelle de ces SBT).

- Vérifier que $\int_{t_1}^{t_2} f_0(x)dx + f_1(t_2) - f_1(t_1) = 0$, c'est-à-dire que :

$$\forall t_1 < t_2, 0 \geq P_{i0}(t_2) - P_{i0}(t_1) + p_{i1}(t_2) - p_{s1}(t_1)$$

$$\text{et } \forall t_1 < t_2, 0 \leq P_{s0}(t_2) - P_{s0}(t_1) + p_{s1}(t_2) - p_{i1}(t_1)$$

L'algorithme proposé dans [Loiez et al, 96a] est une méthode efficace pour vérifier ces inégalités. Il peut être étendu aux SBT d'ordre supérieur.

3- Le troisième algorithme possible consiste à déterminer les instants d'intégration les plus propices pour détecter l'incohérence des modèles d'ordre 1. L'avantage d'un tel algorithme est de prendre en compte l'imprécision introduite par l'addition d'un intervalle à une SBT. Nous proposons maintenant cet algorithme.

8.4.4 Algorithme de détection d'inconsistance d'un modèle d'ordre 1

Nous présentons dans un premier temps une condition nécessaire pour assurer la consistance d'un modèle d'ordre 1. Cette condition permettra d'obtenir l'algorithme initial. Celui ci sera ensuite optimisé. Il semble que cette condition soit aussi suffisante. Toutefois, la preuve n'a pu être achevée.

8.4.4.1 Algorithme initial

Proposition 8.1: "Condition nécessaire pour la consistance des modèles d'ordre 1"

Soient SB_0, SB_1 deux SBT d'extension temporelle T .

Si un modèle d'ordre 1 $(C, T, (f_0(t), f_1(t)))$ est consistant pour le couple (SB_0, SB_1) alors les suites $\int_{\theta} SB_0 - SB_1(\theta)$, pour θ variant dans T , s'intersectent.

D'une manière formelle :

$$(C, T, (f_0(t), f_1(t))) \text{ consistant pour } (SB_0, SB_1) \Rightarrow \bigcap_{\theta \in T} \int_{\theta} SB_0 - SB_1(\theta) \neq \emptyset$$

Preuve :

Soient SB_0, SB_1 deux SBT d'extension temporelle T .

$(C, T, (f_0(t), f_1(t)))$ consistant pour (SB_0, SB_1)

$$\Leftrightarrow \exists g_0(t) \in \mathbf{F}, \exists g_1(t) \in \mathbf{F}(\mathbf{1}), g_0(t) \text{ Rep } SB_0 \wedge g_1(t) \text{ Rep } SB_1 \wedge \forall t \in T, g_0(t) + g_1'(t) = 0$$

$$\Leftrightarrow \exists g_1(t) \in \mathbf{F}(\mathbf{1}), -g_1'(t) \text{ Rep } SB_0 \wedge g_1(t) \text{ Rep } SB_1 \quad (\text{car } g_0(t) = -g_1'(t))$$

$$\Leftrightarrow \exists g(t) \in \mathbf{F}(\mathbf{1}), g'(t) \text{ Rep } SB_0 \wedge g(t) \text{ Rep } -SB_1 \quad (\text{remplacer } g_1(t) \text{ par } -g(t))$$

Or, on a :

$$\forall t \in T, \forall \theta \in T, \int_{\theta}^t g'(x)dx = g(t) - g(\theta) \Leftrightarrow \forall t \in T, \forall \theta \in T, \int_{\theta}^t g'(x)dx + g(\theta) = g(t)$$

Comme les opérations sur les SBT conservent la propriété de représentation, on a :

$$g'(t) \text{ Rep } SB_0 \wedge g(t) \text{ Rep } -SB_1 \Rightarrow \forall \theta \in T, \int_{\theta}^t g'(x)dx + g(\theta) \text{ Rep } \int_{\theta} SB_0 - SB_1(\theta)$$

D'où

$(C, T, (f_0(t), f_1(t)))$ consistant pour (SB_0, SB_1)

$$\Leftrightarrow \exists g(t) \in \mathbf{F}(\mathbf{1}), g'(t) \text{ Rep } SB_0 \wedge g(t) \text{ Rep } -SB_1$$

$$\Rightarrow \exists g(t) \in \mathbf{F}(\mathbf{1}), \forall \theta \in T, \int_{\theta}^t g'(x)dx + g(\theta) \text{ Rep } \int_{\theta} SB_0 - SB_1(\theta)$$

$$\Rightarrow \exists g(t) \in \mathbf{F}(\mathbf{1}), g(t) \text{ Rep } \bigcap_{\theta \in T} \int_{\theta} SB_0 - SB_1(\theta)$$

$$\Rightarrow \bigcap_{\theta \in T} \int_{\theta} SB_0 - SB_1(\theta) \neq \emptyset$$

L'intersection de ces SBT n'est pas vide car elles doivent représenter au moins une fonction commune.

□

Il découle de cette proposition que vérifier l'inconsistance d'un modèle d'ordre 1 revient à vérifier qu'une infinité de SBT s'intersectent. Nous allons maintenant montrer que ceci peut être ramené à un nombre fini de SBT.

L'intuition est que les fonctions associées aux suites de bandes temporelles $\int_{\theta_1} SB_0 - SB_1(\theta_1)$ et $\int_{\theta_2} SB_0 - SB_1(\theta_2)$ sont parallèles sur les intervalles $[t_i(SB_1), \min(\theta_1, \theta_2)]$ et $[\max(\theta_1, \theta_2), t_s(SB_1)]$. Ceci découle directement du lemme 4.3. Ces fonctions sont décalées par rapport à l'axe des abscisses. Un exemple est illustré figure 8.17.

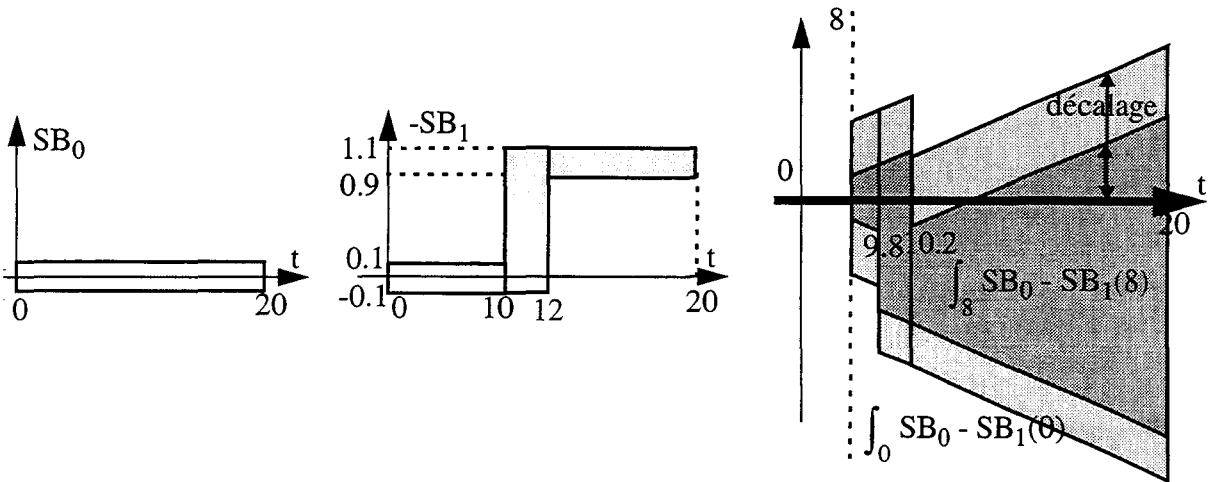


Figure 8.17: Décalage des fonctions associées à $\int_{\theta_1} SB_0 - SB_1(\theta_1)$ et $\int_{\theta_2} SB_0 - SB_1(\theta_2)$

Supposons que $\theta_1 < \theta_2$. Le décalage

- pour les fonctions inférieures sur l'intervalle $[\max(\theta_1, \theta_2), t_s(SB_1)]$ est $P_{i0}(\theta_2) - P_{i0}(\theta_1) - p_{s1}(\theta_1) - (P_{i0}(\theta_2) - P_{i0}(\theta_2) - p_{s1}(\theta_2)) = P_{i0}(\theta_2) - P_{i0}(\theta_1) - p_{s1}(\theta_1) + p_{s1}(\theta_2)$
- pour les fonctions supérieures sur l'intervalle $[\max(\theta_1, \theta_2), t_s(SB_1)]$ est $P_{s0}(\theta_2) - P_{s0}(\theta_1) - p_{i1}(\theta_1) + p_{i1}(\theta_2)$
- pour les fonctions inférieures sur l'intervalle $[t_i(SB_1), \min(\theta_1, \theta_2)]$ est $P_{s0}(\theta_2) - P_{s0}(\theta_1) - p_{s1}(\theta_1) + p_{s1}(\theta_2)$
- pour les fonctions inférieures sur l'intervalle $[\max(\theta_1, \theta_2), t_s(SB_1)]$ est $P_{i0}(\theta_2) - P_{i0}(\theta_1) - p_{s1}(\theta_1) + p_{i1}(\theta_2)$

Nous allons montrer que l'intersection de l'infinité de SBT est égal à l'intersection des SBT correspondant aux instants pour lesquels ce décalage est extremum, c'est-à-dire pour lesquels les dérivées partielles des décalages s'annulent.

Afin de faciliter l'écriture des propositions suivantes, nous introduisons les ensembles d'instant correspondant aux décalages extrema.

Définition 8.3: “ $T(SB_0, SB_1)$, $TD(SB_0, SB_1)$, $TF(SB_0, SB_1)$ ”

Soient SB_0, SB_1 deux suites de bandes temporelles synchrones de taille n , d'extension temporelle T et p_{i0}, p_{s0}, p_{i1} et p_{s1} les fonctions associées à ces suites.

On notera :

- T_1 l'ensemble ordonné des instants de coupure de SB_0 (et donc aussi de SB_1),
 $T_1 = \{t_i(SB_0[1]), t_i(SB_0[2]), \dots, t_i(SB_0[n]), t_s(SB_0[n])\}$
- T_2 l'ensemble ordonné des solutions de l'équation $p_{i0}(t) + p_{s1}'(t)$ appartenant à l'extension temporelle de SB_0 ,
 $T_2 = \{x \in T - T_1 \mid p_{i0}(t) + p_{s1}'(t) = 0\}$
- T_3 l'ensemble ordonné des solutions de l'équation $p_{s0}(t) + p_{i1}'(t)$ appartenant à l'extension temporelle de SB_0 ,
 $T_3 = \{x \in T - T_1 \mid p_{s0}(t) + p_{i1}'(t) = 0\}$
- T_4 l'ensemble ordonné des solutions de l'équation $p_{s0}(t) + p_{s1}'(t)$ appartenant à l'extension temporelle de SB_0 ,
 $T_4 = \{x \in T - T_1 \mid p_{s0}(t) + p_{s1}'(t) = 0\}$
- T_5 l'ensemble ordonné des solutions de l'équation $p_{i0}(t) + p_{i1}'(t)$ appartenant à l'extension temporelle de SB_0 ,
 $T_5 = \{x \in T - T_1 \mid p_{i0}(t) + p_{i1}'(t) = 0\}$
- $TD(SB_0, SB_1)$ l'ensemble ordonné contenant les éléments de T_1, T_2 et T_3 .
- $TF(SB_0, SB_1)$ l'ensemble ordonné contenant les éléments de T_1, T_4 et T_5 .
- $T(SB_0, SB_1)$ l'ensemble ordonné contenant les éléments de $TD(SB_0, SB_1)$ et de $TF(SB_0, SB_1)$.

Nous nous intéressons maintenant au calcul de $\bigcap_{\theta \in T} \int_{\theta} SB_0 - SB_1(\theta)$.

Proposition 8.2:

L'intersection des suites $\int_{\theta} SB_0 - SB_1(\theta)$ pour tout instant θ de T est égale à l'intersection de $-SB_1$ et des suites $\int_{\theta} SB_0 - SB_1(\theta)$ pour tout instant θ de $T(SB_0, SB_1)$.

D'une manière formelle :

$$\bigcap_{\theta \in T} \int_{\theta} SB_0 - SB_1(\theta) = -SB_1 \cap \bigcap_{\theta \in T(SB_0, SB_1)} \int_{\theta} SB_0 - SB_1(\theta)$$

Preuve :

Soient SB_0, SB_1 deux SBT d'extension temporelle $T = [t_i, t_s[$,

$p_{i0}(t), p_{s0}(t)$ les fonctions associées à SB_0 ,

$p_{i1}(t), p_{s1}(t)$ les fonctions associées à SB_1 ,

et $P_{i0}(t), P_{s0}(t)$ les fonctions associées à $\int_{t_i} SB_0$.

La fonction inférieure associée à $\int_{\theta} SB_0 - SB_1(\theta)$ est

Chapitre 8: Famille d'algorithmes de détection d'inconsistance

$$\begin{aligned} f_i(t) &= P_{s0}(t) - P_{s0}(\theta) - ps1(\theta) \quad \text{pour } t \leq \theta \\ &= P_{i0}(t) - P_{i0}(\theta) - ps1(\theta) \quad \text{pour } t \geq \theta \end{aligned}$$

D'après la définition de l'intersection de 2 SBT, la fonction inférieure associée à l'intersection de 2 SBT est égale au maximum des fonctions inférieures associées à ces SBT. Comme l'intersection est associative, ceci s'étend à n SBT.

D'où, la fonction inférieure associée à $\bigcap_{\theta \in T} \int_{\theta} SB_0 - SB_1(\theta)$ est égale à

$$\begin{aligned} F_i(t) &= \max\left(\max_{\theta \in [t, t_s]} (P_{s0}(t) - P_{s0}(\theta) - ps1(\theta)), \max_{\theta \in [t_i, t]} (P_{i0}(t) - P_{i0}(\theta) - ps1(\theta))\right) \\ &= \max(P_{s0}(t) + \max_{\theta \in [t, t_s]} (-P_{s0}(\theta) - ps1(\theta)), P_{i0}(t) + \max_{\theta \in [t_i, t]} (-P_{i0}(\theta) - ps1(\theta))) \end{aligned}$$

Supposons que $F_i(t) = P_{s0}(t) + \max_{\theta \in [t, t_s]} (-P_{s0}(\theta) - ps1(\theta))$. Le maximum de $-P_{s0}(\theta) - ps1(\theta)$ pour $\theta \in [t, t_s]$ est obtenu

- soit en t. On a alors $F_i(t) = -ps1(t)$.
- soit en un des instants où la fonction $-P_{s0}(t) - ps1(t)$ est discontinue, c'est à dire les instants de coupure de SB_1 (i.e. en un instant de T_1).
- soit en un instant tel que la dérivée de $-P_{s0}(t) - ps1(t)$ s'annule, c'est à dire en un instant de T_4 .

D'où, si $F_i(t) = P_{s0}(t) + \max_{\theta \in [t, t_s]} (-P_{s0}(\theta) - ps1(\theta))$ alors

- soient les fonctions inférieures de $\bigcap_{\theta \in T} \int_{\theta} SB_0 - SB_1(\theta)$ et $-SB_1$ sont égales en t,
- soient les fonctions inférieures de $\bigcap_{\theta \in T} \int_{\theta} SB_0 - SB_1(\theta)$ et $\int_x SB_0 - SB_1(x)$ sont égales en t avec x appartenant à T_1 ou à T_4 .

De même, si $F_i(t) = P_{i0}(t) + \max_{\theta \in [t_i, t]} (-P_{i0}(\theta) - ps1(\theta))$ alors

- soient les fonctions inférieures de $\bigcap_{\theta \in T} \int_{\theta} SB_0 - SB_1(\theta)$ et $-SB_1$ sont égales en t,
- soient les fonctions inférieures de $\bigcap_{\theta \in T} \int_{\theta} SB_0 - SB_1(\theta)$ et $\int_x SB_0 - SB_1(x)$ sont égales en t avec x appartenant à T_1 ou à T_2 .

De la même manière, on montre que la fonction supérieure associée à $\bigcap_{\theta \in T} \int_{\theta} SB_0 - SB_1(\theta)$ est égale à la fonction supérieure de $-SB_1$ ou d'une des SBT $\int_{\theta} SB_0 - SB_1(\theta)$ pour θ appartenant à T_1, T_2, T_3, T_4 ou T_5 .

Comme les fonctions associées à $\bigcap_{\theta \in T} \int_{\theta} SB_0 - SB_1(\theta)$ sont égales aux fonctions associées à $-SB_1$ ou à une des SBT $\int_{\theta} SB_0 - SB_1(\theta)$ pour θ appartenant à T_1, T_3 ou T_5 , on a

$$\bigcap_{\theta \in T} \int_{\theta} SB_0 - SB_1(\theta) = -SB_1 \cap \bigcap_{\theta \in T(SB_0, SB_1)} \int_{\theta} SB_0 - SB_1(\theta)$$

□

Cette proposition nous permet d'assurer que la condition nécessaire pour assurer la consistance d'un modèle d'ordre 1 consiste à vérifier qu'un nombre fini de SBT s'intersectent. Un algorithme immédiat consiste à calculer l'intersection des SBT. Toutefois, un tel algorithme serait inefficace car seul le fait que ces SBT s'intersectent est intéressant. Nous montrons maintenant comment vérifier que n SBT s'intersectent sans passer par le calcul de l'intersection.

Proposition 8.3: “*intersection de n SBT*”

L'intersection de n SBT est vide ssi l'intersection de 2 de ces SBT est vide.

D'une manière formelle :

$$SB_1 \cap \dots \cap SB_n = \emptyset \Leftrightarrow \exists i, j \in [1, n] \text{ tel que } SB_i \cap SB_j = \emptyset$$

Preuve :

Par définition de l'intersection, on a :

$$SB_1 \cap SB_2 \cap \dots \cap SB_k = \emptyset$$

$$\Leftrightarrow \exists t_0 \in [t_i, t_s], SB_1(t_0) \cap SB_2(t_0) \cap \dots \cap SB_k(t_0) = \emptyset$$

Or, la valeur d'une SBT en un instant (i.e. $SB_i(t_0)$) est un intervalle. Or, l'intersection de k intervalles est vide si et seulement si deux de ces intervalles ne s'intersectent pas. D'où :

$$SB_1 \cap SB_2 \cap \dots \cap SB_k = \emptyset$$

$$\Leftrightarrow \exists t_0 \in [t_i, t_s], SB_1(t_0) \cap SB_2(t_0) \cap \dots \cap SB_k(t_0) = \emptyset$$

$$\Leftrightarrow \exists t_0 \in [t_i, t_s], \exists i, j \in [1, k], SB_i(t_0) \cap SB_j(t_0) = \emptyset$$

$$\Leftrightarrow SB_i \cap SB_j = \emptyset$$

□

L'algorithme initial découle de cette proposition.

Vérifier le critère nécessaire de la consistance d'un modèle d'ordre 1 consiste à vérifier que :

- pour tout instant θ de $T(SB_0, SB_1)$, les SBT $\int_{\theta} SB_0 - SB_1(\theta)$ et $-SB_1$ s'intersectent,
- pour tout couple d'instants (θ_1, θ_2) de $T(SB_0, SB_1)$,
les SBT $\int_{\theta_1} SB_0 - SB_1(\theta_1)$ et $\int_{\theta_2} SB_0 - SB_1(\theta_2)$ s'intersectent.

Un certain nombre de ces vérifications est inutile. Nous proposons maintenant une optimisation de cet algorithme.

8.4.4.2 Optimisation de l'algorithme

L'optimisation découle de l'intuition suivante. La fonction inférieure (resp. supérieure) associée à $-SB_1 \cap_{\theta \in T(SB_0, SB_1)} \int_{\theta} SB_0 - SB_1(\theta)$ est égale

- soit à l'intégrale indéfini de la fonction inférieure associée à SB_0 plus une constante,
- soit à l'intégrale indéfini de la fonction supérieure associée à SB_0 plus une constante,
- soit à la fonction inférieure (resp. supérieure) associée à $-SB_1$,

où les constantes correspondent aux décalages.

Chapitre 8: Famille d'algorithmes de détection d'inconsistance

Nous introduisons d'abord la suite de bandes temporelles $D(SB_0, SB_1)$ (resp. $F(SB_0, SB_1)$), suite qui est égale à $-SB_1 \cap_{\theta \in T(SB_0, SB_1)} \int_{\theta} SB_0$ lorsque la fonction inférieure associée à cette suite est l'intégrale indéfini de la fonction inférieure (resp. supérieure) associée à SB_0 et sa fonction supérieure est est l'intégrale indéfini de la fonction supérieure (resp. inférieure) associée à SB_0 plus une constante. Nous montrerons alors que $-SB_1 \cap_{\theta \in T(SB_0, SB_1)} \int_{\theta} SB_0 - SB_1(\theta) = D(SB_0, SB_1) \cap F(SB_0, SB_1) \cap -SB_1$.

Définition 8.4: "Suite de bandes temporelles $D(SB_0, SB_1)$ "

Soit $TD(SB_0, SB_1) = \{x_1, \dots, x_p\}$,

soient $P_{i0}(t), P_{s0}(t)$ les fonctions associées à $\int_{t_i} SB_0$ et $p_{i1}(t), p_{s1}(t)$ les fonctions associées à SB_1 .

$D(SB_0, SB_1)$

$= \{(x_1, x_2, P_{i0}(t) + M(x_1), P_{s0}(t) + m(x_1)), \dots, (x_{p-1}, x_p, P_{i0}(t) + M(x_{p-1}), P_{s0}(t) + m(x_{p-1}))\}$

où $M(x_i) = \max(-P_{i0}(x_1) - p_{s1}(x_1), \dots, -P_{i0}(x_i) - p_{s1}(x_i))$

et $m(x_i) = \min(-P_{s0}(x_1) - p_{i1}(x_1), \dots, -P_{s0}(x_i) - p_{i1}(x_i))$.

Exemple :

La figure 8.18 illustre deux SBT SB_0 et SB_1 et la suite $D(SB_0, SB_1)$ en résultant.

Les dérivées des fonctions associées à SB_1 sont nulles. Elles n'intersectent pas les fonctions associées à SB_0 . Par conséquent, $TD(SB_0, SB_1)$ est composée des instants de coupure de ces SBT, c'est à dire $TD(SB_0, SB_1) = \{0, 10, 12, 20\}$

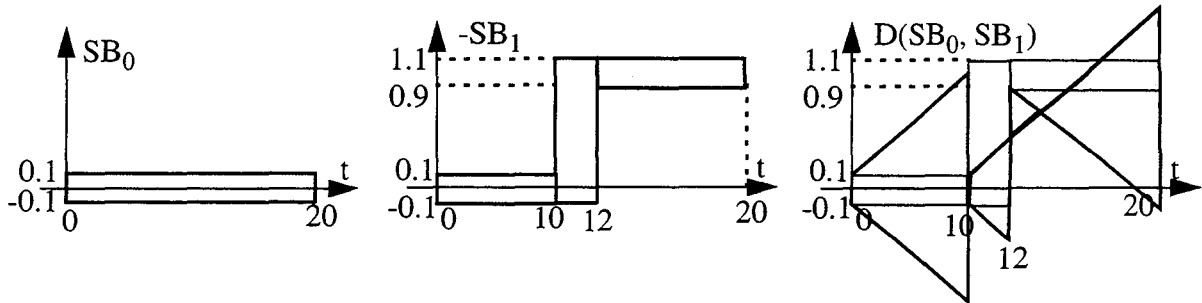


Figure 8.18 : Exemple de SBT $D(SB_0, SB_1)$

Comme le montre la figure précédente, la suite de bandes temporelles $D(SB_0, SB_1)$ n'existe pas nécessairement. Nous dirons dans ce cas qu'elle est vide. La proposition suivante est un moyen efficace pour détecter lorsque cette SBT est vide.

Proposition 8.4:

La SBT $D(SB_0, SB_1)$ est vide ssi la valeur de cette SBT à un des instants de coupure est vide.

D'une manière formelle :

Soit $T(SB_0, SB_1) = \{x_1, \dots, x_p\}$.

$SB = D(SB_0, SB_1) = \emptyset \Leftrightarrow \exists i = 1 \dots p$ tel que $SB(x_i) = \emptyset$

Preuve :

Il découle directement de la définition d'une SBT vide que

$$\exists i = 1 \dots p \text{ tel que } SB(x_i) = \emptyset \Rightarrow SB = D(SB_0, SB_1) = \emptyset$$

Nous montrons maintenant l'inverse.

Soit $TD(SB_0, SB_1) = \{x_1, \dots, x_p\}$.

Les fonctions inférieure et supérieure associées à $D(SB_0, SB_1)$ sont respectivement

$$f_i(t) = P_{i0}(t) + M(x_i) \text{ et } f_s(t) = P_{s0}(t) + m(x_i)$$

où $P_{i0}(t), P_{s0}(t)$ sont les fonctions associées à $\int_{x_1} SB_0$.

D'où

$$SB = D(SB_0, SB_1) = \emptyset$$

$$\Leftrightarrow \exists t_0 \in [x_1, x_p[\text{ tel que } SB(t_0) = [f_i(t_0), f_s(t_0)] = \emptyset$$

$$\Leftrightarrow \exists t_0 \in [x_1, x_p[\text{ tel que } f_i(t_0) > f_s(t_0)$$

$$\Leftrightarrow \exists i \in \{1, \dots, p\}, \exists t_0 \in [x_i, x_{i+1}[\text{ tel que } P_{i0}(t_0) + M(x_i) > P_{s0}(t_0) + m(x_i)$$

$$\Leftrightarrow \exists i \in \{1, \dots, p\}, \exists t_0 \in [x_i, x_{i+1}[\text{ tel que } P_{i0}(t_0) - P_{s0}(t_0) > m(x_i) - M(x_i)$$

Or, la fonction $P_{i0}(t) - P_{s0}(t)$ est décroissante car $P_{i0}'(t) - P_{s0}'(t) = p_{i0}(t) - p_{s0}(t) \leq 0$ (par définition des SBT).

D'où

$$D(SB_0, SB_1) = \emptyset$$

$$\Leftrightarrow \exists i \in \{1, \dots, p\}, \exists t_0 \in [x_i, x_{i+1}[\text{ tel que } P_{i0}(t_0) - P_{s0}(t_0) > m(x_i) - M(x_i)$$

$$\Rightarrow \exists i \in \{1, \dots, p\}, P_{i0}(x_i) - P_{s0}(x_i) > m(x_i) - M(x_i) \text{ (car } P_{i0}(x_i) - P_{s0}(x_i) \geq P_{i0}(t_0) - P_{s0}(t_0))$$

$$\Rightarrow \exists i \in \{1, \dots, p\}, SB(x_i) = [P_{i0}(x_i) + M(x_i), P_{i0}(x_i) + m(x_i)] = \emptyset$$

□

Un algorithme de calcul de $D(SB_0, SB_1)$ est proposé ci dessous.

Soient $SB_0 = \{(t_1, t_2, p_{i0_1}(t), p_{s0_1}(t)), (t_2, t_3, p_{i0_2}(t), p_{s0_2}(t)), \dots, (t_n, t_{n+1}, p_{i0_n}(t), p_{s0_n}(t))\}$,

$SB_1 = \{(t_1, t_2, p_{i1_1}(t), p_{s1_1}(t)), (t_2, t_3, p_{i1_2}(t), p_{s1_2}(t)), \dots, (t_n, t_{n+1}, p_{i1_n}(t), p_{s1_n}(t))\}$.

On note $P_{i0_i}(t)$ (resp. P_{s0_i}) l'intégrale indéfinie de p_{i0_i} (resp. p_{s0_i})

$$\text{sup} \leftarrow - p_{s1_1}(t_1)$$

$$\text{inf} \leftarrow - p_{i1_1}(t_1)$$

$$i \leftarrow 1$$

existence \leftarrow vrai

$$D(SB_0, SB_1) \leftarrow \emptyset$$

faire

Créer l'ensemble ordonné contenant t_i, t_{i+1} et les solutions des équations $-p_{i0_i}(t) - p_{s1_i}'(t) = 0$ et $-p_{s0_i}(t) - p_{i1_i}'(t) = 0$. Soit $T = \{x_1, \dots, x_p\}$ cet ensemble (NB : $x_1 = t_i, x_p = t_{i+1}$).

$$j \leftarrow 1$$

```

faire
  sup ← max(sup + Pi0i(xj), - ps1i(xj))
  inf ← min(inf + Ps0i(xj), - pi1i(xj))
  D(SB0, SB1) ← D(SB0, SB1) ∪ {(xj, xj+1, Pi0i(t) - Pi0i(xj) + sup, Ps0i(t) - Ps0i(xj) + inf)}
  si sup > inf alors
    existence ← faux
    D(SB0, SB1) ← ∅

  fsi
  j ← j + 1
jusqu'à j = p ou existence = faux
  sup ← max(sup + Pi0i(xp), - ps1i(xp))
  inf ← min(inf + Ps0i(xp), - pi1i(xp))1
  i ← i + 1
jusqu'à i = n + 1 ou existence = faux
  
```

Nous proposons maintenant de définir une deuxième SBT.

Définition 8.5: “Suite de bandes temporelles $F(SB_0, SB_1)$ ”

Soit $TF(SB_0, SB_1) = \{x_1, \dots, x_p\}$,

soient $Pi0(t), Ps0(t)$ les fonctions associées à $\int_{t_i} SB_0$ et $pi1(t), ps1(t)$ les fonctions associées à SB_1 .

$$F(SB_0, SB_1) = \{(x_1, x_2, Ps0(t) + M(x_2), Pi0(t) + m(x_2)), \dots, (x_{p-1}, x_p, Ps0(t) + M(x_p), Pi0(t) + m(x_p))\}$$

où $M(x_j) = \max(-Ps0(x_p) - ps1(x_p), \dots, -Ps0(x_j) - ps1(x_j))$

et $m(x_j) = \min(-Pi0(x_p) - pi1(x_p), \dots, -Pi0(x_j) - pi1(x_j))$.

Exemple :

Deux SBT SB_0 et SB_1 ainsi que la suite $F(SB_0, SB_1)$ sont illustrées figure 8.19.

On a $TF(SB_0, SB_1) = \{0, 10, 12, 20\}$.

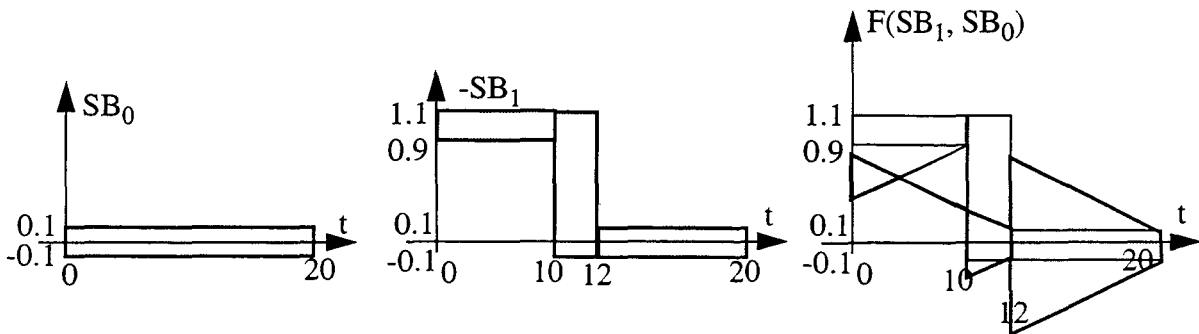


Figure 8.19 : Exemple de SBT $F(SB_1, SB_0)$

L'algorithmme permettant de créer $F(SB_0, SB_1)$ est similaire à celui de $D(SB_0, SB_1)$.

1. Ces 2 clauses permettent de prendre en considération la discontinuité de SB_1 .

Nous nous intéressons maintenant au calcul de $-SB_1 \cap_{\theta \in T(SB_0, SB_1)} \int_{\theta} SB_0 - SB_1(\theta)$.

Proposition 8.5:

$$-SB_1 \cap \cap_{\theta \in T(SB_0, SB_1)} \int_{\theta} SB_0 - SB_1(\theta) = -SB_1 \cap D(SB_0, SB_1) \cap F(SB_0, SB_1)$$

Preuve :

Soit $T(SB_0, SB_1) = \{x_1, \dots, x_p\}$.

Pour deux éléments consécutifs x_i, x_{i+1} de $T(SB_0, SB_1)$, la fonction inférieure associée à

$-SB_1 \cap \cap_{\theta \in T(SB_0, SB_1)} \int_{\theta} SB_0 - SB_1(\theta)$ est le maximum des fonctions inférieures associées à ces $p + 1$ SBT, c'est à dire $\forall t \in [x_i, x_{i+1}]$,

$$\begin{aligned} f_i(t) &= \max(-ps1(t), Pi0(t) - Pi0(x_1) - ps1(x_1), \dots, Pi0(t) - Pi0(x_i) - ps1(x_i), \\ &\quad Ps0(t) - Ps0(x_{i+1}) - ps1(x_{i+1}), \dots, Ps0(t) - Ps0(x_p) - ps1(x_p)) \\ &= \max(-ps1(t), Pi0(t) + \max(-Pi0(x_1) - ps1(x_1), \dots, -Pi0(x_i) - ps1(x_i)), \\ &\quad Ps0(t) + \max(-Ps0(x_{i+1}) - ps1(x_{i+1})), \dots, -Ps0(x_p) - ps1(x_p)) \end{aligned}$$

Sur l'intervalle $[x_i, x_{i+1}]$, la fonction inférieure associée à $-SB_1 \cap D(SB_0, SB_1) \cap F(SB_0, SB_1)$ est

$$F_i(t) = \max(-ps1(t), Pi0(t) + \max(-Pi0(y_1) - ps1(y_1), \dots, -Pi0(y_j) - ps1(y_j)), \\ Ps0(t) + \max(-Ps0(z_k) - ps1(z_k)), \dots, -Ps0(z_n) - ps1(z_n))$$

avec $y_j \leq x_i < y_{j+1}$ et $y_i \in TD(SB_0, SB_1) \subseteq T(SB_0, SB_1)$ ($i = 1 \dots j$).

et $z_{k-1} \leq x_i < z_k$ et $z_i \in TF(SB_0, SB_1) \subseteq T(SB_0, SB_1)$ ($i = k \dots n$)

Donc, les fonctions inférieures associées à $-SB_1 \cap \cap_{\theta \in T(SB_0, SB_1)} \int_{\theta} SB_0 - SB_1(\theta)$ et à $-SB_1 \cap D(SB_0, SB_1) \cap F(SB_0, SB_1)$ sont différentes ssi $\exists i \in \{1, \dots, p\}, \exists t_0 \in [x_i, x_{i+1}]$ tel que $f_i(t_0) = Pi0(t_0) + \max(-Pi0(x_1) - ps1(x_1), \dots, -Pi0(x_i) - ps1(x_i)) = Pi0(t_0) - Pi0(x_k) - ps1(x_k)$ ($1 \leq k \leq i$) avec $x_k \notin TD(SB_0, SB_1)$ et $f_i(t_0) \neq -ps1(t_0)$.

ou

$$f_i(t_0) = Ps0(t_0) + \max(-Ps0(x_{i+1}) - ps1(x_{i+1})), \dots, -Ps0(x_p) - ps1(x_p)) = Ps0(t_0) - Ps0(x_k) - ps1(x_k)$$

($i+1 \leq k \leq p$) avec $x_k \notin TF(SB_0, SB_1)$ et $f_i(t_0) \neq -ps1(t_0)$.

Montrons que ces deux cas sont impossibles.

Supposons que $\exists t_0 \in [x_i, x_{i+1}], f_i(t_0) = Pi0(t_0) - Pi0(x_k) - ps1(x_k)$ avec $x_k \notin TD(SB_0, SB_1)$ et $x_k \leq x_i$ et $f_i(t_0) \neq -ps1(t_0)$.

Par définition, $TD(SB_0, SB_1)$ contient les instants pour lesquels $-Pi0(t) - ps1(t)$ est maximum. Comme $x_k \notin TD(SB_0, SB_1)$, $-Pi0(x_k) - ps1(x_k)$ n'est pas un maximum local de $-Pi0(t) - ps1(t)$.

Comme $-Pi0(x_k) - ps1(x_k) = \max(-Pi0(x_1) - ps1(x_1), \dots, -Pi0(x_i) - ps1(x_i))$ avec $x_j \in T(SB_0, SB_1)$ ($j = 1 \dots i$), la fonction $-Pi0(t) - ps1(t)$ est croissante sur $[x_k, x_{i+1}]$ car $-Pi0(x_k) - ps1(x_k)$ est supérieur à tous les maxima locaux de $-Pi0(t) - ps1(t)$ sur $[x_k, x_{i+1}[$.

Donc, $\forall t \in [x_k, x_{i+1}[, f_i(t) = Pi0(t) - Pi0(x_k) - ps1(x_k) \leq Pi0(t) - Pi0(t) - ps1(t) = -ps1(t)$

Chapitre 8: Famille d'algorithmes de détection d'inconsistance

Or, par définition, $f_i(t) \geq -p_{s1}(t)$. Donc, $f_i(t) = -p_{s1}(t)$. Ce qui est absurde car $f_i(t) \neq -p_{s1}(t)$. Donc, $f_i(t)$ ne peut pas être égal à $P_{i0}(t) - P_{i0}(x_k) - p_{s1}(x_k)$ avec $x_k \notin TD(SB_0, SB_1)$.

De la même manière, on peut montrer que $f_i(t)$ ne peut pas être égal à $P_{s0}(t) - P_{s0}(x_k) - p_{s1}(x_k)$ avec $x_k \notin TF(SB_0, SB_1)$ car dans ce cas, la fonction $-P_{s0}(t) - p_{s1}(t)$ est décroissante.

Donc, les fonctions inférieures associées à $-SB_1 \cap \bigcap_{\theta \in T(SB_0, SB_1)} \int_{\theta} SB_0 - SB_1(\theta)$ et à $-SB_1 \cap D(SB_0, SB_1) \cap F(SB_0, SB_1)$ sont égales. De la même manière, on peut montrer que les fonctions supérieures sont aussi égales.

Comme les fonctions inférieures et supérieures de ces SBT sont égales, ces SBT sont égales.

□

Cette proposition permet de simplifier l'algorithme initial. En effet, Vérifier le critère nécessaire de consistance d'un modèle d'ordre 1 pour un couple de SBT (SB_0, SB_1) consiste à vérifier que :

- $D(SB_0, SB_1)$ et $F(SB_0, SB_1)$ s'intersectent,
- $-SB_1$ et $F(SB_0, SB_1)$ s'intersectent,
- $-SB_1$ et $D(SB_0, SB_1)$ s'intersectent.

Par rapport à l'algorithme initial, cet algorithme permet non seulement d'éviter certaines vérifications mais surtout permet de limiter la taille des fenêtres temporelles sur lesquelles ces vérifications doivent être réalisées.

Nous allons maintenant montrer que deux des trois vérifications sont inutiles.

Proposition 8.6:

Les trois propositions suivantes sont équivalentes.

- L'intersection de $-SB_1$, $D(SB_0, SB_1)$ et $F(SB_0, SB_1)$ n'est pas vide,
- l'intersection de $-SB_1$ et $D(SB_0, SB_1)$ n'est pas vide,
- l'intersection de $-SB_1$ et $F(SB_0, SB_1)$ n'est pas vide,

D'une manière formelle :

$$-SB_1 \cap D(SB_0, SB_1) \cap F(SB_0, SB_1) \neq \emptyset$$

$$\Leftrightarrow -SB_1 \cap D(SB_0, SB_1) \neq \emptyset$$

$$\Leftrightarrow -SB_1 \cap F(SB_0, SB_1) \neq \emptyset$$

Preuve :

Nous montrons uniquement que

$$-SB_1 \cap D(SB_0, SB_1) \cap F(SB_0, SB_1) \neq \emptyset \Leftrightarrow -SB_1 \cap D(SB_0, SB_1) \neq \emptyset$$

La preuve de la deuxième équivalence est obtenue par symétrie.

Soient SB_0, SB_1 deux SBT d'extensions temporelles T .

Montrer cette équivalence revient à montrer que

$$-SB_1 \cap D(SB_0, SB_1) \neq \emptyset \Rightarrow D(SB_0, SB_1) \cap F(SB_0, SB_1) \neq \emptyset$$

$$\text{et } -SB_1 \cap D(SB_0, SB_1) \neq \emptyset \Rightarrow -SB_1 \cap F(SB_0, SB_1) \neq \emptyset$$

Montrons que $-SB_1 \cap D(SB_0, SB_1) \neq \emptyset \Rightarrow D(SB_0, SB_1) \cap F(SB_0, SB_1) \neq \emptyset$

Raisonnons par l'absurde et supposons que $-SB_1 \cap D(SB_0, SB_1) \neq \emptyset$ et $D(SB_0, SB_1) \cap F(SB_0, SB_1) = \emptyset$.

L'intersection de 2 SBT est vide ssi le polynôme inférieur d'une des SBT est supérieur en un point au polynôme supérieur de l'autre, ou inversement. D'où :

$$D(SB_0, SB_1) \cap F(SB_0, SB_1) = \emptyset$$

$$\Rightarrow \exists x_i \in TD(SB_0, SB_1), \exists y_i \in TF(SB_0, SB_1), \exists x \in [x_i, y_i] \text{ tel que}$$

$$Pi0(x) - Pi0(x_i) - ps1(x_i) > Pi0(x) - Pi0(y_i) - pi1(y_i)$$

$$\vee Ps0(x) - Ps0(x_i) - pi1(x_i) < Ps0(x) - Ps0(y_i) - ps1(y_i)$$

$$\Rightarrow \exists x_i \in TD(SB_0, SB_1), \exists y_i \in TF(SB_0, SB_1) \text{ tel que } y_i \geq x_i \wedge$$

$$(Pi0(y_i) - Pi0(x_i) - ps1(x_i) > -pi1(y_i))$$

$$\vee Ps0(y_i) - Ps0(x_i) - pi1(x_i) < -ps1(y_i))$$

et

$$-SB_1 \cap D(SB_0, SB_1) \neq \emptyset$$

$$\Rightarrow \forall x_i \in TD(SB_0, SB_1), \forall x \geq x_i,$$

$$-ps1(x) \leq Ps0(x) - Ps0(x_i) - pi1(x_i)$$

$$\wedge -pi1(x) \geq Pi0(x) - Pi0(x_i) - ps1(x_i)$$

$$\Rightarrow \forall x_i \in TD(SB_0, SB_1), \forall y_i \in TF(SB_0, SB_1), y_i \geq x_i,$$

$$-ps1(y_i) \leq Ps0(y_i) - Ps0(x_i) - pi1(x_i)$$

$$\wedge -pi1(y_i) \geq Ps0(y_i) - Ps0(x_i) - pi1(x_i)$$

Il est donc absurde de supposer que $D(SB_0, SB_1) \cap F(SB_0, SB_1) = \emptyset$. D'où

$$-SB_1 \cap D(SB_0, SB_1) \neq \emptyset \Rightarrow D(SB_0, SB_1) \cap F(SB_0, SB_1) \neq \emptyset$$

Montrons que $-SB_1 \cap D(SB_0, SB_1) \neq \emptyset \Rightarrow -SB_1 \cap F(SB_0, SB_1) \neq \emptyset$

Raisonnons par l'absurde et supposons que $-SB_1 \cap D(SB_0, SB_1) \neq \emptyset$ et $-SB_1 \cap F(SB_0, SB_1) = \emptyset$.

$$-SB_1 \cap F(SB_0, SB_1) = \emptyset$$

$$\Rightarrow \exists y_i \in TF(SB_0, SB_1), \exists x \leq y_i,$$

$$-ps1(x) > Pi0(x) - Pi0(y_i) - pi1(y_i)$$

$$\vee -pi1(x) < Ps0(x) - Ps0(y_i) - ps1(y_i)$$

Or, $\forall x \in T, \exists x' \leq x$ tel que $x' \in TD(SB_0, SB_1)$ et $-Pi0(x') - ps1(x') \geq -Pi0(x) - ps1(x)$

car $TD(SB_0, SB_1)$ contient les instants pour lesquels la fonction $-Pi0(t) - ps1(t)$ est maximum.

Chapitre 8: Famille d'algorithmes de détection d'inconsistance

De même, $\forall x \in T, \exists x' \leq x$ tel que $x' \in TD(SB_0, SB_1)$ et $-Ps_0(x') - pi_1(x') \leq -Ps_0(x) - pi_1(x)$

D'où

$$\begin{aligned} & -SB_1 \cap F(SB_0, SB_1) = \emptyset \\ \Rightarrow & \exists y_i \in TF(SB_0, SB_1), \exists x' \in TD(SB_0, SB_1) \text{ tel que } x' \leq y_i \wedge \\ & (-ps_1(x') > Pi_0(x') - Pi_0(y_i) - pi_1(y_i) \\ & \vee -pi_1(x') < Ps_0(x') - Ps_0(y_i) - ps_1(y_i)) \end{aligned}$$

De plus,

$$\begin{aligned} & -SB_1 \cap D(SB_0, SB_1) \neq \emptyset \\ \Rightarrow & \forall x_i \in TD(SB_0, SB_1), \forall y_i \in TF(SB_0, SB_1), y_i \geq x_i, \\ & -ps_1(y_i) \leq Ps_0(y_i) - Ps_0(x_i) - pi_1(x_i) \\ & \wedge -pi_1(y_i) \geq Pi_0(y_i) - Pi_0(x_i) - pi_1(x_i) \end{aligned}$$

Il est donc absurde de supposer que $-SB_1 \cap D(SB_0, SB_1) \neq \emptyset$ et $-SB_1 \cap F(SB_0, SB_1) = \emptyset$.

D'où $-SB_1 \cap D(SB_0, SB_1) \neq \emptyset \Rightarrow -SB_1 \cap F(SB_0, SB_1) \neq \emptyset$

□

Il découle directement des propositions précédentes que :

$$\begin{aligned} & (C, T, (f_0(t), f_1(t))) \text{ consistant pour } (SB_0, SB_1) \\ \Rightarrow & -SB_1 \cap D(SB_0, SB_1) \neq \emptyset \end{aligned}$$

D'où l'algorithme :

Vérifier la condition nécessaire de consistance d'un modèle d'ordre 1 pour un couple de SBT (SB_0, SB_1) consiste à :

- **calculer (et montrer l'existence de) $D(SB_0, SB_1)$,**
- **vérifier que $-SB_1$ et $D(SB_0, SB_1)$ s'intersectent,**

Une inconsistance est détectée lorsqu'une des deux clauses n'est pas respectée.

Nous nous intéressons maintenant à la complétude de cet algorithme.

8.4.4.3 Complétude de l'algorithme

Démontrer la complétude de l'algorithme revient à montrer l'inverse de la proposition 8.1.

Proposition 8.7: *“condition suffisante pour la consistance des modèles d'ordre 1”*

Soient SB_0, SB_1 deux SBT d'extension temporelle T.

Un modèle d'ordre 1 $(C, T, (f_0(t), f_1(t)))$ est consistant pour le couple (SB_0, SB_1) ssi, pour tout instant θ de T, les suites $\int_{\theta} SB_0 - SB_1(\theta)$ s'intersectent.

D'une manière formelle :

$$(C, T, (f_0(t), f_1(t))) \text{ consistant pour } (SB_0, SB_1) \Leftrightarrow \bigcap_{\theta \in T} \int_{\theta} SB_0 - SB_1(\theta) \neq \emptyset$$

Preuve (incomplète) :

Nous avons montré dans la proposition 8.1 que

$$(C, T, (f_0(t), f_1(t))) \text{ consistant pour } (SB_0, SB_1) \Rightarrow \bigcap_{\theta \in T} \int_{\theta} SB_0 - SB_1(\theta) \neq \emptyset$$

Il reste à démontrer l'inverse.

Montrer qu'un modèle d'ordre 1 est consistant pour un couple de SBT (SB_0, SB_1) revient à montrer l'existence d'une fonction $f(t)$ continûment dérivable telle que $f(t) \text{ Rep } -SB_1$ et $f'(t) \text{ Rep } SB_0$. En effet :

$$(C, T, (f_0(t), f_1(t))) \text{ consistant pour } (SB_0, SB_1) \Leftrightarrow \exists g(t) \in \mathbf{F}(\mathbf{1}), g'(t) \text{ Rep } SB_0 \wedge g(t) \text{ Rep } -SB_1$$

Il faut donc montrer que

$$\bigcap_{\theta \in T} \int_{\theta} SB_0 - SB_1(\theta) \neq \emptyset \Rightarrow \exists g(t) \in \mathbf{F}(\mathbf{1}), g'(t) \text{ Rep } SB_0 \wedge g(t) \text{ Rep } -SB_1$$

D'après la proposition 8.2, on a

$$\bigcap_{\theta \in T} \int_{\theta} SB_0 - SB_1(\theta) = -SB_1 \cap \bigcap_{\theta \in T(SB_0, SB_1)} \int_{\theta} SB_0 - SB_1(\theta)$$

La fonction inférieure associée à $-SB_1 \cap \bigcap_{\theta \in T(SB_0, SB_1)} \int_{\theta} SB_0 - SB_1(\theta)$, $Fi(t)$, est représentée par $-SB_1$ car $-SB_1 \cap \bigcap_{\theta \in T(SB_0, SB_1)} \int_{\theta} SB_0 - SB_1(\theta) \subseteq -SB_1$.

Montrons que la dérivée de $Fi(t)$ est représentée par SB_0 .

$Fi(t)$ est le maximum des fonctions

$$-ps_1(t), Pi_0(t) - Pi_0(x) - ps_1(x), Ps_0(t) - Ps_0(y) - ps_1(y) \text{ avec } x \leq t \leq y \text{ et } x, y \in T(SB_0, SB_1)$$

$Fi'(t)$ est donc égale à $-ps_1'(t)$ ou $Pi_0'(t) = pi_0(t)$ ou $P_s0'(t) = ps_0(t)$.

Supposons que $\forall t \in [t_1, t_2], Fi(t) = -ps_1(t)$.

Montrons alors que $\forall t \in [t_1, t_2], pi_0(t) \leq -ps_1'(t) \leq ps_0(t)$.

On a $\forall t \in [t_1, t_2], Fi(t) = -ps_1(t) \geq Pi_0(t) - Pi_0(x) - ps_1(x), \forall x \in T(SB_0, SB_1)$ et $x \leq t$

D'où $\forall t \in [t_1, t_2], \forall x \in T(SB_0, SB_1), x \leq t, -Pi_0(t) - ps_1(t) \geq -Pi_0(x) - ps_1(x)$

Donc, la fonction $-Pi_0(t) - ps_1(t)$ est croissante, soit $-Pi_0'(t) - ps_1'(t) = -pi_0(t) - ps_1'(t) \geq 0$

De la même manière, on montre que $-Ps_0(t) - ps_1(t)$ est décroissante, soit $-ps_1'(t) \leq ps_0(t)$.

Donc, la fonction inférieure associée à $\bigcap_{\theta \in T} \int_{\theta} SB_0 - SB_1(\theta) \neq \emptyset$ est représentée par $-SB_1$ et sa dérivée est représentée par SB_0 . Il semble possible de montrer que cette fonction est continue. Toutefois, ceci n'est pas suffisant car cette fonction n'est pas nécessairement dérivable. Toutefois, elle admet toujours une dérivée à droite et à gauche.

Chapitre 8: Famille d'algorithmes de détection d'inconsistance

Il semble possible de construire à partir de $F_i(t)$ une fonction $g(t)$ continûment dérivable telle que $g(t) \text{ Rep } -SB_1$ et $g'(t) \text{ Rep } SB_0$.

Soient x_1, \dots, x_n les instants où la fonction $F_i'(t)$ est discontinue.

Soit $h_\varepsilon(t)$ la fonction telle que

$$\forall t \in [x_i + \varepsilon, x_{i+1} - \varepsilon], h_\varepsilon(t) = F_i'(t)$$

$\forall t \in [x_{i+1} - \varepsilon, x_{i+1} + \varepsilon]$, $h_\varepsilon(t)$ est égale à l'équation de la droite reliant les points de coordonnées $(x_{i+1} - \varepsilon, F_i'(x_{i+1} - \varepsilon))$ et $(x_{i+1} + \varepsilon, F_i'(x_{i+1} + \varepsilon))$.

La fonction $h(t)$ est représentée par SB_0 . Soit $g_\varepsilon(t) = \int_{t_i}^t h_\varepsilon(x) dx$. Il semble possible de montrer que, pour ε suffisamment petit, $g_\varepsilon(t)$ est représentée par $-SB_1$.

Donc, si $\bigcap_{\theta \in T} \int_{\theta} SB_0 - SB_1(\theta)$ n'est pas vide alors, à partir de la fonction inférieure associée à cette SBT, il semble possible de construire une fonction $g_\varepsilon(t)$ représentée par $-SB_1$ et dont la dérivée est représentée par SB_0 . L'existence de cette fonction permet d'assurer la consistance du modèle d'ordre 1.

□

Le théorème suivant est un résumé des propositions établies précédemment.

Théorème 8.1: "Consistance d'un modèle d'ordre 1"

Les quatre propositions suivantes sont équivalentes :

- Le modèle d'ordre 1 $(C, T, (f_0(t), f_1(t)))$ est consistant pour le couple (SB_0, SB_1) ,
- pour tout instant θ de $TD(SB_0, SB_1)$, les modèles d'ordre 0 $(C, T, (\int_{\theta}^t f_0(x) dx + f_1(t) - f_1(\theta)))$ sont consistants pour $(\int_{\theta} SB_0 + SB_1 - SB_1(\theta))$,
- pour tout instant θ de $TF(SB_0, SB_1)$, les modèles d'ordre 0 $(C, T, (\int_{\theta}^t f_0(x) dx + f_1(t) - f_1(\theta)))$ sont consistants pour $(\int_{\theta} SB_0 + SB_1 - SB_1(\theta))$,
- les SBT $D(SB_0, SB_1)$ et $-SB_1$ s'intersectent.

D'une manière formelle :

$(C, T, (f_0(t), f_1(t)))$ consistant pour (SB_0, SB_1)

$$\Leftrightarrow \forall \theta \in TD(SB_0, SB_1), (C, T, (\int_{\theta}^t f_0(x) dx + f_1(t) - f_1(\theta))) \text{ consistant pour } (\int_{\theta} SB_0 + SB_1 - SB_1(\theta))$$

$$\Leftrightarrow \forall \theta \in TF(SB_0, SB_1), (C, T, (\int_{\theta}^t f_0(x) dx + f_1(t) - f_1(\theta))) \text{ consistant pour } (\int_{\theta} SB_0 + SB_1 - SB_1(\theta))$$

$$\Leftrightarrow D(SB_0, SB_1) \cap -SB_1 \neq \emptyset$$

La preuve de ce théorème découle directement des propositions précédentes. Il signifie que, non seulement la vérification de la consistance de certains modèles d'ordre 0 est suffisante pour assurer la consistance d'un modèle d'ordre 1, mais aussi qu'il est inutile de réaliser ces

vérifications sur l'extension temporelle des SBT SB_0, SB_1 . Il est suffisant de vérifier la consistance du modèle $(C, T, (\int_{\theta}^t f_0(x)dx + f_1(t) - f_1(\theta)))$ sur la fenêtre temporelle $[\theta, \omega]$ où θ et ω sont deux instants de $TD(SB_0, SB_1)$ (resp. $TF(SB_0, SB_1)$) tels que $-Pi_0(\theta) - p_{s1}(\theta) < -Pi_0(\omega) - p_{s1}(\omega)$ ou $-P_{s0}(\theta) - pi_1(\theta) > -P_{s0}(\omega) - pi_1(\omega)$ (resp. $-P_{s0}(\theta) - p_{s1}(\theta) > -P_{s0}(\omega) - p_{s1}(\omega)$ ou $-Pi_0(\theta) - pi_1(\theta) > -Pi_0(\omega) - pi_1(\omega)$).

8.5 Extension aux modèles d'ordre n

Deux procédures de détection d'inconsistance ont été établies ; une pour les modèles d'ordre 0 et une pour ceux d'ordre 1. Nous avons montré que l'inconsistance d'un modèle d'ordre 1 peut être détectée en vérifiant l'inconsistance d'un modèle d'ordre 0. Toutefois, cet algorithme n'est pas complet. C'est pourquoi nous avons proposé un algorithme spécifique pour les modèles d'ordre 1. L'extension aux modèles d'ordre n est basée sur ce principe : l'inconsistance d'un modèle d'ordre n peut être détectée en vérifiant l'inconsistance d'un modèle d'ordre n-1. Ces algorithmes sont incomplets, mais nous n'avons pas développé d'algorithme spécifique. L'extension aux modèles d'ordre 2 est d'abord présentée. La généralisation est ensuite proposée.

8.5.1 Consistance des modèles d'ordre 2

8.5.1.1 Formalisation

Tout modèle d'ordre 2 est équivalent à une infinité de modèles d'ordre 1. En effet, étant donné un instant θ de T , on a :

$$\begin{aligned} \forall t \in T, f_0(t) + f_1'(t) + f_2''(t) &= 0 \\ \Leftrightarrow \forall t \in T, \int_{\theta}^t f_0(x)dx + f_1(t) - f_1(\theta) + f_2'(t) - f_2'(\theta) &= 0 \end{aligned}$$

et

$$(C, T, (f_0(t), f_1(t), f_2(t))) \Leftrightarrow (C, T, (\int_{\theta}^t f_0(x)dx + f_1(t) - f_1(\theta) - f_2'(\theta), f_2(t)))$$

Il découle de cette équivalence que l'inconsistance d'un modèle d'ordre 2 est équivalente à la C-inconsistance d'un modèle d'ordre 1. Soit :

$$\begin{aligned} (C, T, (f_0(t), f_1(t), f_2(t))) \text{ consistant pour } (SB_0, SB_1, SB_2) \\ \Leftrightarrow (C, T, (\int_{\theta}^t f_0(x)dx + f_1(t) - f_1(\theta) - f_2'(\theta), f_2(t))) \\ \text{C-consistant pour } \{(f_0(t), SB_0), (f_1(t), SB_1), (f_2(t), SB_2)\} \end{aligned}$$

Comme l'inconsistance implique la C-inconsistance (théorème 7.2), l'inconsistance d'un modèle d'ordre 2 peut être détectée en vérifiant l'inconsistance d'un modèle d'ordre 1 pour les OBS-extensions des fonctions du modèle, c'est-à-dire les fonctions $\int_{\theta}^t f_0(x)dx + f_1(t) - f_1(\theta) - f_2'(\theta)$ et $f_2(t)$. Le problème est que le calcul de l'OBS-extension nécessite un encadrement de

Chapitre 8: Famille d'algorithmes de détection d'inconsistance

$f_2'(\theta)$. Or, nous ne disposons pas d'un tel encadrement.

Nous supposons l'existence d'une procédure permettant de calculer un encadrement de la dérivée d'une fonction. Un tel encadrement est difficile à obtenir (les SBT ont d'ailleurs été introduites pour éviter de tels calculs). Toutefois, ce problème est atténué car il est possible de vérifier l'inconsistance d'un modèle d'ordre 2 dès que l'on dispose d'un encadrement de $f_2'(t)$ en un instant. De plus, cet instant peut être choisi librement.

L'algorithme découle alors directement de ces remarques.

Détecter l'inconsistance du modèle $(C, T, (f_0(t), f_1(t), f_2(t)))$ pour (SB_0, SB_1, SB_2) consiste à :

- Calculer un encadrement I de $f_2'(t)$ à un instant θ .
- Détecter l'inconsistance du modèle $(C, T, (g_0(t), g_1(t)))$ pour $(\int_{\theta} SB_0 + SB_1 - SB_1(\theta) - I, SB_2)$

Nous présentons maintenant un exemple.

8.5.1.2 Exemple : le circuit LC

Soit le système illustré figure 8.20.

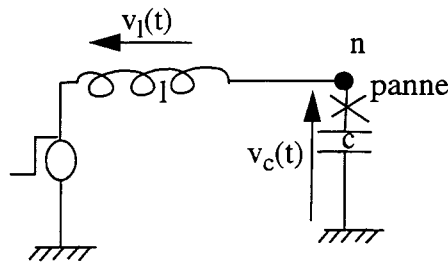


Figure 8.20 : Circuit LC

Le modèle de ce système est $(\{r, l, n\}, \mathbf{R}, (v_1(t), 0, -LC v_c(t)))$.

Afin de simplifier l'exemple, nous prendrons $LC = 1$.

Lorsque ce système fonctionne correctement, une montée rapide de la tension aux bornes du générateur provoque une montée progressive de la tension aux bornes de la capacité et une diminution progressive de la tension aux bornes de la self. Les tensions $v_1(t)$ et $v_c(t)$ sont illustrées figure 8.21.

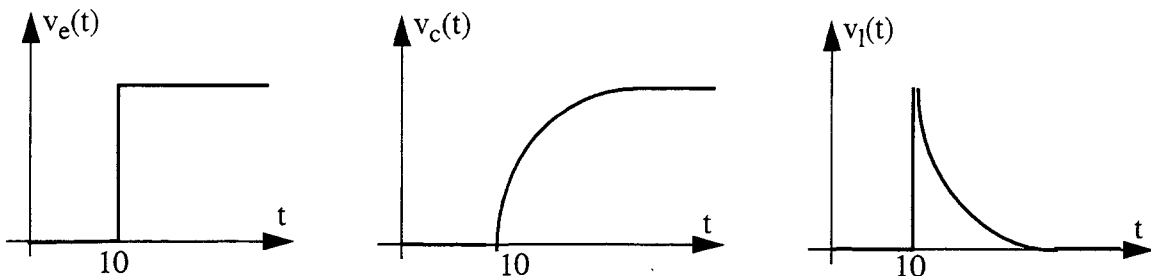


Figure 8.21 : Grandeurs lorsque les composants fonctionnent correctement

Par contre, lorsque le circuit est ouvert à l'endroit indiqué sur la figure 8.20, aucun courant ne circule dans le système. La tension aux bornes de la self reste alors toujours nulle. Les tensions

aux bornes de la capacité et du générateur sont identiques. Les signaux et les SBT correspondant à cette panne sont illustrés figure 8.22.

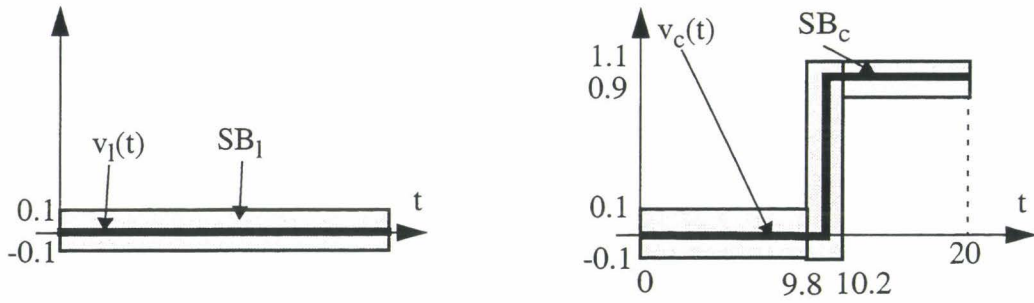


Figure 8.22 : Signaux lorsque le système est en panne

Soit l'ensemble des observations $OBS = \{(v_r(t), SB_r), (v_c(t), SB_c)\}$ avec $SB_r = \{(0, 20, -0.1, 0.1)\}$ et $SB_c = \{(0, 9.8, -0.1, 0.1), (9.8, 10.2, -0.1, 1.1), (10.2, 20, 0.9, 1.1)\}$.

Nous allons montrer que le modèle de ce système, c'est-à-dire $(\{r, l, n\}, \mathbf{R}, (v_1(t), 0, -v_c(t)))$, est inconsistant pour le triplet de SBT $(SB_1, 0, -SB_c)$.

Pour cela nous proposons de vérifier l'inconsistance du modèle obtenu par intégration, c'est-à-dire $(\{r, l, n\}, \mathbf{R}, (\int_{\theta}^t v_1(x)dx + v_c'(\theta), -v_c(t)))$.

Détecter l'inconsistance de ce modèle nécessite le calcul d'un encadrement de ces fonctions et, par conséquent, de disposer d'un encadrement de $v_c'(\theta)$ pour un instant θ quelconque.

Comme le système est stable pendant l'intervalle $[0, 9.8]$ (i.e. tous les signaux sont constants), la dérivée de $v_c(t)$ est nulle pour tous ces instants. Nous prendrons $v_c'(5) = 0$.

La SBT SB représentant la fonction $\int_5^t v_1(x)dx + v_c'(5)$ (i.e. $\int_5 SB_1$) est illustrée figure 8.23.

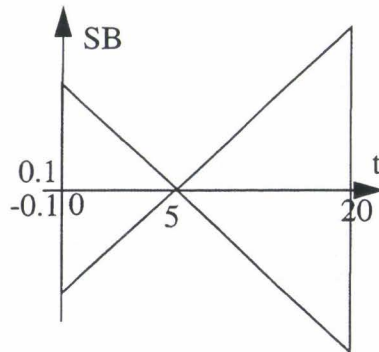


Figure 8.23 : SBT représentant la première fonction du modèle d'ordre 1

Il reste alors à détecter l'inconsistance du modèle $(\{r, l, n\}, \mathbf{R}, (\int_{\theta}^t v_1(x)dx + v_c'(\theta), -v_c(t)))$ pour le couple $(\int_5 SB_1, SB_c)$. Ceci est réalisé en calculant $D(\int_5 SB_1, SB_c)$ qui est vide (cf. figure 8.24).

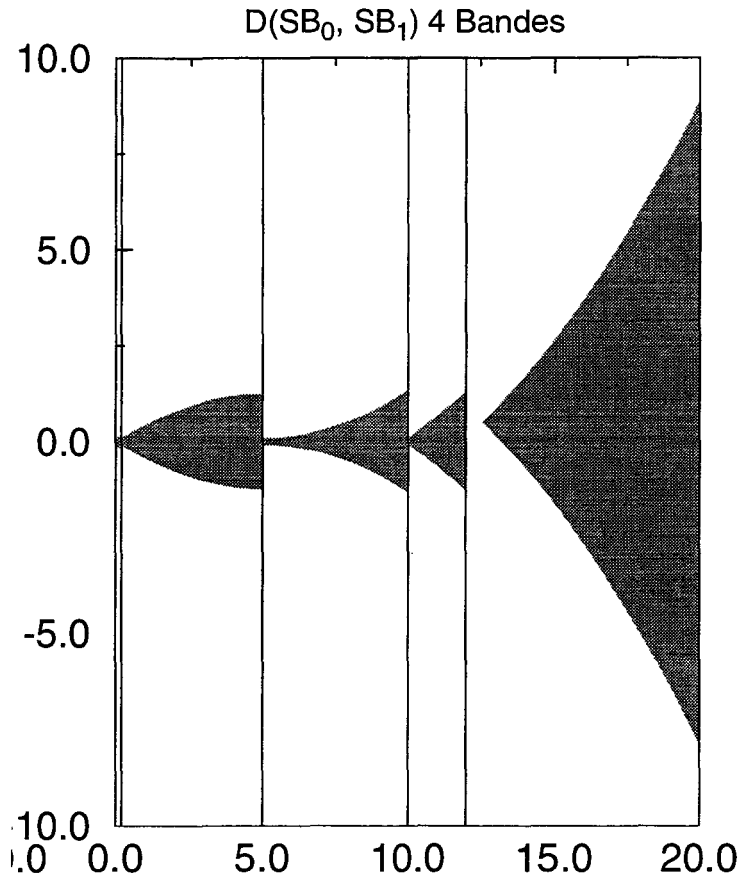


Figure 8.24 : Suite de bandes temporelles D

8.5.1.3 Complétude de l'algorithme

L'algorithme de détection d'inconsistance des modèles d'ordre 2 est incomplet car il nécessite le calcul d'une SBT représentant la fonction $\int_{\theta}^t f_0(x)dx + f_1(t) - f_1(\theta) - f_2'(\theta)$. Ce calcul est imprécis car il fait intervenir l'intégration de la SBT représentant $f_0(t)$. Une manière de limiter l'imprécision introduite est de calculer plusieurs SBT pour différents instants d'intégration, puis de vérifier la consistance du modèle pour l'intersection des SBT obtenues. Le problème est que les instants d'intégration sont imposés par la nécessaire connaissance d'un encadrement de $f_2'(t)$ à ces instants. Les résultats obtenus par CATS montrent qu'un encadrement de "bonne qualité" de la dérivée d'une fonction peut être obtenu lorsque cette fonction varie lentement. Pour l'exemple proposé précédemment, un encadrement de $v_c'(t)$ peut être obtenu pour un instant de l'intervalle $[10.2, 20]$ car $v_c(t)$ varie lentement durant cet intervalle. En prenant $v_c'(15) = 0$, le nouvel encadrement de la fonction $g(t) = \int_{\theta}^t v_1(x)dx + v_c'(\theta)$ est illustré figure 8.25.

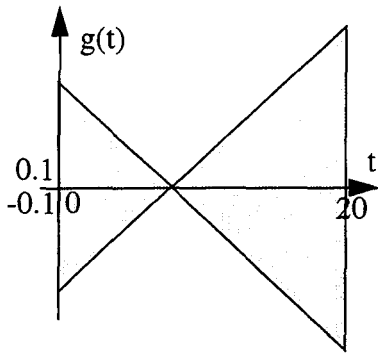


Figure 8.25a : Ancien encadrement

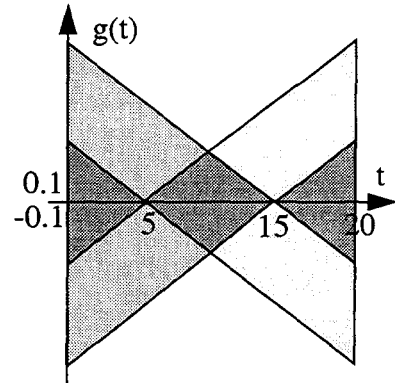


Figure 8.25b : Nouvel encadrement

Par contre, le calcul de l'encadrement de la dérivée d'une fonction variant rapidement est difficile (voire impossible). Pour ce cas de figure, il est alors intéressant de calculer un encadrement de la dérivée d'une fonction à partir de la SBT représentant cette fonction. Ce calcul est réalisé directement à partir du théorème des accroissements finis : il existe un instant θ appartenant à $[9.8, 10.2]$ tel que $v_c'(\theta)$ est supérieur à la pente de la droite (D) (i.e. $v_c'(\theta) > 2$) illustrée figure 8.25. A partir de cette information, un nouvel encadrement de $g(t)$ peut être obtenu en réalisant l'union des SBT obtenues pour $v_c'(\theta) = 2$ et θ appartenant à $[9.8, 10.2]$.

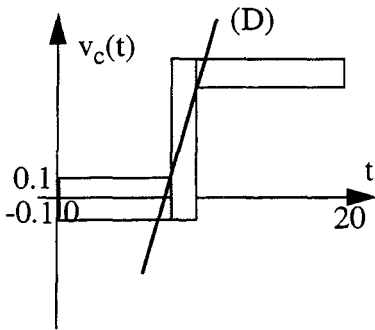


Figure 8.26a : Ancien encadrement

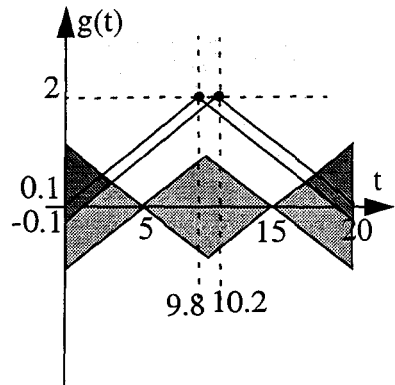


Figure 8.26b : Nouvel encadrement

On remarque que l'encadrement obtenu à partir de l'information $v_c'(\theta) > 2$ n'intersecte pas les encadrements obtenus précédemment. Ceci nous assure que le modèle est inconsistant. Le phénomène intéressant est que l'inconsistance du modèle d'ordre 2 est détectée avant d'avoir vérifié l'inconsistance du modèle d'ordre 1 équivalent.

En conclusion, améliorer la procédure de détection d'inconsistance des modèles d'ordre 2 (i.e. $(C, T, (f_0(t), f_1(t), f_2(t)))$) consiste à améliorer l'encadrement de la fonction obtenue après une première intégration (i.e. $\int_{\theta}^t f_0(x)dx + f_1(t) - f_1(\theta) - f_2'(\theta)$). Pour cela, il est nécessaire d'obtenir des encadrements de $f_2'(\theta)$ pour différents instants θ . Deux solutions sont possibles :

- créer une procédure ad hoc,
- déterminer ces encadrements à partir de la SBT représentant $f_2(t)$. Ces encadrements sont déterminés à partir du théorème des accroissements finis : lorsque $f(t)$ appartient à $[p_i(t), p_s(t)]$,

Chapitre 8: Famille d'algorithmes de détection d'inconsistance

pour tout intervalle $[t_1, t_2]$, il existe un instant c de cet intervalle tel que

$$f'(c) \in \left[\frac{P_i(t_2) - P_s(t_1)}{(t_2 - t_1)}, \frac{P_s(t_2) - P_i(t_1)}{t_2 - t_1} \right].$$

L'utilisation de cette information nécessite une procédure calculant l'union des SBT représentant $\int_{\theta}^t f_0(x)dx + f_1(t) - f_1(\theta) - f_2'(\theta)$ ($\theta \in [t_1, t_2]$).

Il n'est donc pas obligatoire d'avoir une procédure de calcul de la dérivée d'une fonction pour détecter l'inconsistance d'un modèle d'ordre 2. Toutefois, il semble que, lorsque la fonction $f_2(t)$ varie lentement, une procédure ad hoc soit plus efficace. Inversement, lorsque $f_2(t)$ varie rapidement, déterminer un encadrement de la dérivée d'une fonction à partir de la SBT représentant cette fonction semble plus efficace.

Une extension de l'algorithme aux modèles d'ordre n est maintenant proposée.

8.5.1.4 Extension à l'ordre n

L'extension à l'ordre n est basée sur le fait que tout modèle d'ordre n $(C, T, (f_0(t), \dots, f_n(t)))$ est inconsistant pour (SB_0, \dots, SB_n) lorsqu'un des modèles d'ordre $n-1$ équivalents (i.e.

$$(C, T, (\int_{\theta}^t f_0(x)dx + f_1(t) - \sum_{i=1}^n f_i^{(i-1)}(\theta), \dots, f_n(t)))$$

est inconsistant pour les OBS-extensions de ses fonctions. Ce calcul nécessite un encadrement des réels $f_i^{(i-1)}(\theta)$ ($i = 1 \dots n$).

L'algorithme est défini par récurrence :

Détecter l'inconsistance d'un modèle d'ordre n $(C, T, (f_0(t), \dots, f_n(t)))$ pour (SB_0, \dots, SB_n) consiste à :

- Déterminer des intervalles I_i encadrant $f_i^{(i-1)}(\theta)$ où θ est un instant librement choisi,
- Détecter l'inconsistance du modèle d'ordre $n-1$ $(C, T, (g_0(t), f_2(t), \dots, f_n(t)))$ pour

$$(\int_{\theta} SB_0 + SB_1 - \sum_{i=1}^n I_i, SB_2, \dots, SB_n).$$

8.6 Conclusion

Le but du diagnostic est de vérifier la C-consistance des modèles car, en général, l'ensemble des observations ne contient pas les SBT représentant les fonctions des modèles. Toutefois, seuls deux algorithmes de détection d'inconsistance ont été proposés ; celui pour les modèles d'ordre 0 et celui pour les modèles d'ordre 1. Pour vérifier la C-consistance des modèles, la première étape consiste à calculer les SBT représentant les fonctions des modèles, SBT que nous avons appelée OBS-extensions. La deuxième étape consiste à vérifier la consistance des modèles. Des algorithmes directs sont applicables lorsque ceux-ci sont d'ordre 0 ou 1. Une extension de ces

Chapitre 8: Famille d'algorithmes de détection d'inconsistance

algorithmes est nécessaire pour considérer les grandeurs constantes au cours du temps. Lorsque l'ordre du modèle est supérieur à 2, une étape intermédiaire consiste à intégrer l'équation du modèle jusqu'à l'obtention d'un modèle d'ordre 1. Cette étape nécessite le calcul des dérivées de certaines grandeurs à partir des mesures. Ces étapes sont résumées sur le diagramme de la figure 8.27.

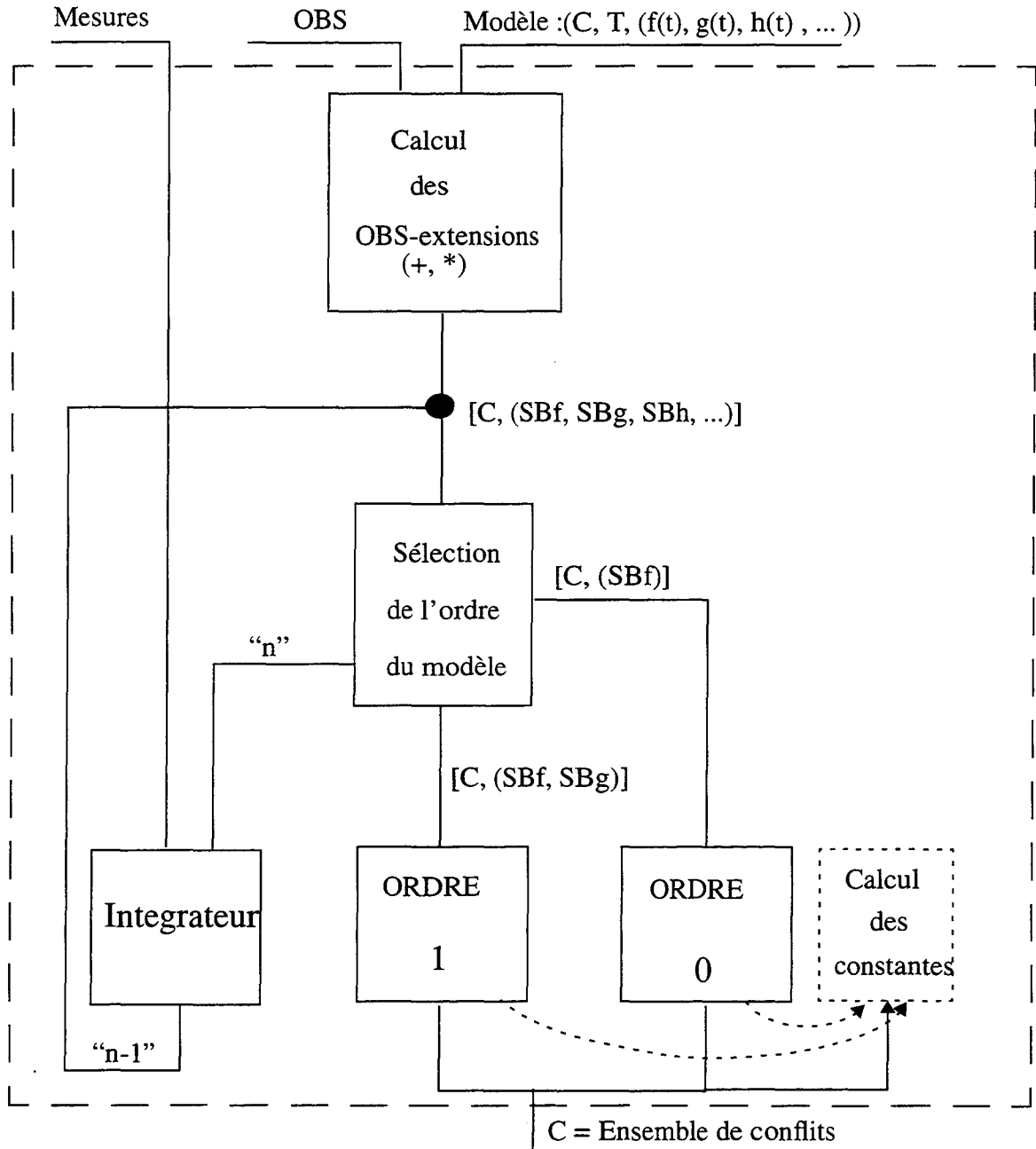


Figure 8.27 : Le détecteur d'inconsistance

Conclusion

Nous avons présenté dans cette partie une procédure de comparaison des observations et du comportement théorique d'un système. Les grandeurs observées sont des fonctions temporelles. Nous avons proposé de différencier les grandeurs constantes au cours du temps de celles qui ne le sont pas. Les observations de ces grandeurs sont respectivement des intervalles et des suites de bandes temporelles. Ce choix est motivé par le fait que les opérations invoquant ces différentes grandeurs ont des propriétés différentes. Par exemple, l'addition d'un intervalle et d'une SBT introduit de l'imprécision alors que l'addition de deux SBT n'en introduit pas. Le comportement théorique du système est décrit par une équation différentielle. Pour formaliser la description de ce comportement, la notion de modèle a été introduite.

Etant donnée notre description du système, il est impossible de diagnostiquer directement les systèmes, c'est-à-dire de déterminer tous les composants défectueux. C'est pourquoi le diagnostic est réalisé en deux étapes : la première consiste à déterminer tous les ensembles de conflits, c'est-à-dire tous les ensembles de composants comportant au moins un composant défectueux, la deuxième consiste à déterminer les diagnostics à partir des ensembles de conflits. Nous nous sommes intéressés dans cette partie à la première étape. Plus précisément, nous avons présenté une procédure permettant de détecter si l'ensemble de composants d'un modèle est un ensemble de conflits. Lorsqu'un système est en panne, il existe deux raisons de ne pas détecter d'ensembles de conflits : soit la non détection provient de l'incomplétude de la procédure, soit la non détection provient de l'imprécision de la description des comportements du système. C'est pourquoi nous avons introduit la notion de C-consistance d'un modèle. Lorsque le modèle décrivant le comportement d'un système en panne est C-consistant, le fait de ne pas détecter cette panne provient nécessairement de l'imprécision des informations fournies. Pour vérifier la C-inconsistance des modèles, la procédure consiste

- à calculer les SBT encadrant les fonctions du modèle,
- à intégrer l'équation du modèle si son ordre est supérieur à 2 de manière à obtenir un modèle d'ordre 1,
- à vérifier la consistance du modèle avec les algorithmes spécifiques si ce modèle est d'ordre 0 ou 1.

Cette procédure est complète lorsque

- les fonctions du modèle invoquent uniquement les opérations d'addition et de multiplication,
- les grandeurs n'apparaissent qu'une seule fois dans les fonctions des modèles,
- aucune grandeur n'est constante au cours du temps,
- les modèles sont d'ordre 0 ou 1.

Deux extensions de ce travail sont envisagées. La première consiste à calculer les intervalles encadrant les grandeurs constantes au cours du temps à partir des SBT représentant les autres grandeurs. Cette extension permet de prendre en considération l'imprécision introduite par les intervalles. La deuxième consiste à calculer les dérivées des fonctions à partir des SBT. Cette extension permet de limiter l'incomplétude de la procédure lorsque les modèles sont d'ordre 2.

PARTIE III

Systemes de Diagnostic

Introduction

Diagnostiquer un système consiste à comparer le comportement théorique et les observations de ce système. Une procédure de comparaison a été présentée pour un comportement théorique décrit par un unique modèle : la procédure de détection d'inconsistance. Nous nous intéressons dans cette partie aux systèmes dont les comportements sont décrits par des ensembles de modèles. Le but n'est pas d'étendre la procédure de détection d'inconsistance à un ensemble de modèles. Il est plutôt d'utiliser cette procédure comme un outil afin de créer un système permettant de diagnostiquer un processus décrit par un ensemble de modèles.

Deux approches sont proposées. La première consiste à intégrer les SBT dans le General Diagnosis Engine (GDE) [de Kleer et al, 87]. Cette intégration consiste à réaliser une procédure d'inférence permettant de calculer les SBT représentant les grandeurs des systèmes à partir des modèles et des observations.

L'utilisation d'une procédure d'inférence engendre quelques inconvénients. Ceux ci sont particulièrement exacerbés dans le cadre des SBT. C'est pourquoi nous proposons une alternative, DOGS. Ce système consiste à compiler un ensemble de modèles de manière à obtenir tous les ensembles de conflits lorsque l'on vérifie la consistance de ces modèles un par un.

Les deux premiers chapitres de cette partie sont consacrés à l'étude de ces deux approches. Le troisième chapitre est consacré à l'implémentation de DOGS et aux résultats obtenus. Nous proposons d'abord une procédure de création des SBT à partir des mesures échantillonnées d'un signal. Les résultats obtenus sur le monostable, problème de référence pour les systèmes de diagnostic analogique [Dague et al, 96], sont ensuite présentés.

Chapitre 9

G.D.E et les SBT

Le General Diagnosis Engine est un système de diagnostic indépendant du domaine. Toutefois, il suppose l'existence d'une procédure d'inférence qui dépend de la modélisation des observations. Après un bref rappel de GDE, cette procédure est étudiée dans le cadre des SBT. Les inconvénients engendrés par l'utilisation d'une procédure d'inférence sont alors présentés.

9.1 Rappel de G.D.E.

Il n'est pas question ici de détailler le General Diagnosis Engine. Les principales étapes sont d'abord brièvement présentées. Une présentation un peu moins sommaire des étapes qui seront étendues aux SBT est ensuite proposée.

9.1.1 Les 4 étapes

Schématiquement, GDE est composé de 4 étapes réalisées itérativement. Ces étapes se résument à :

La prédiction des valeurs :

A partir de l'hypothèse que les composants fonctionnent correctement, le but est de calculer les valeurs des grandeurs du système à partir des modèles des composants et des observations.

La détection de conflits :

Cette étape consiste à comparer les valeurs prédites entre elles ou avec les observations. Lorsque ces valeurs sont incohérentes, un ensemble de conflits est détecté (i.e. un ensemble de composants comportant au moins un composant défectueux).

La génération des candidats :

A partir des ensembles de conflits détectés, générer les différents ensembles de candidats possibles (i.e. les composants dont les pannes peuvent expliquer le comportement réel du système).

Le choix de la mesure suivante :

Cette étape consiste à déterminer la grandeur dont l'observation est susceptible de permettre l'affinement du diagnostic. Une approche probabiliste est proposée dans [de Kleer et al, 87] et dans [de Kleer, 90]. Toutefois, il semble que, dans le domaine du diagnostic de systèmes analogiques, une approche heuristique soit préférable [Dague et al, 91].

Une nouvelle observation permet de réaliser de nouvelles prédictions (retour à la première étape). Le processus s'arrête lorsque le diagnostic est jugé satisfaisant (par l'utilisateur ou vis-à-vis d'un critère probabiliste montrant que le diagnostic ne peut plus être affiné) ou lorsque toutes les observations ont été réalisées.

Les deux dernières étapes sont indépendantes de la modélisation des observations. C'est pourquoi nous ne nous y intéressons pas dans cette partie. GDE suppose l'existence d'une

procédure d'inférence permettant de réaliser les deux premières étapes. Toutefois, de Kleer en propose une dont nous donnons les principales caractéristiques.

9.1.2 La procédure d'inférence

La procédure d'inférence proposée par de Kleer est composée d'un propagateur de contraintes couplé à un ATMS (Assumption-based Truth Maintenance System) [de Kleer, 86].

Nous utiliserons comme exemple le circuit illustré figure 9.1¹.

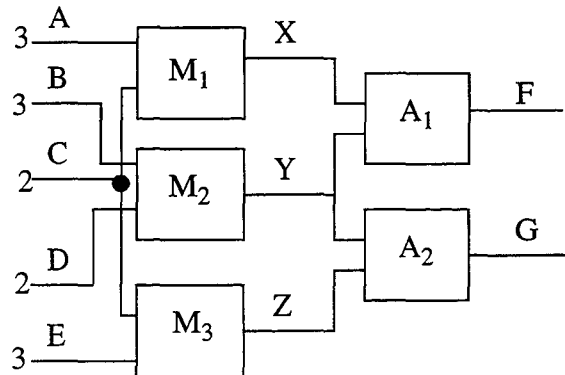


Figure 9.1 : Full Adder

Ce système est constitué de 3 multipliers et de 2 additionneurs dont les modèles sont

$$\begin{aligned} X &= A * C && \text{pour } \{M_1\} \\ Y &= B * D && \text{pour } \{M_2\} \\ Z &= C * E && \text{pour } \{M_3\} \\ F &= X + Y && \text{pour } \{A_1\} \\ G &= Y + Z && \text{pour } \{A_2\} \end{aligned}$$

Le propagateur permet de calculer les valeurs des grandeurs du système à partir des observations et des modèles. Les valeurs sont calculées une par une à partir d'un modèle et des valeurs des autres grandeurs invoquées dans ce modèle.

Par exemple, à partir du modèle de M_1

$$X = A * C,$$

et des observations

$$A = 3 \text{ et } C = 2,$$

la valeur de X est prédite

$$X = 6.$$

A chaque valeur prédite est associé un ou plusieurs ensembles de composants correspondant aux différents modèles utilisés pour prédire la valeur. Un ensemble de composants est appelé un environnement. La liste des environnements est appelée un label. Une prédiction peut avoir plusieurs environnements car une valeur peut être déduite de plusieurs manières différentes. Par exemple, la valeur de F peut être déduite

$$\text{à partir de } M_1, M_2, A_1 : F = X + Y = AC + BD$$

$$\text{à partir de } M_1, M_3, A_1, A_2 : F = X + G - Z = AC + G - EC$$

1. Cet exemple est aussi utilisé par [De Kleer, 86] pour illustrer GDE.

Les prédictions sont notées par un couple [valeur prédite, environnement(s)]. Par exemple, la prédiction de X est notée $[X = 6, \{M_1\}]$. La même notation est utilisée pour les observations ; l'environnement est alors l'ensemble vide (i.e. $\{\}$). Par exemple, l'observation de A est notée $[A = 3, \{\}]$. Ceci traduit le fait qu'une observation est toujours vraie.

Le calcul d'une prédiction est noté
valeurs utilisées ; modèle utilisé \rightarrow valeur prédite

Par exemple, on a :

$[A = 3, \{\}], [C = 2, \{\}]; \{M_1\} \rightarrow [X = 6, \{M_1\}]$

mais aussi

$[B = 3, \{\}], [D = 2, \{\}]; \{M_2\} \rightarrow [Y = 6, \{M_2\}]$

$[C = 2, \{\}], [E = 3, \{\}]; \{M_3\} \rightarrow [Z = 6, \{M_3\}]$

$[X = 6, \{M_1\}], [Y = 6, \{M_2\}]; \{A_1\} \rightarrow [F = 12, \{M_1, M_2, A_1\}]$

$[Y = 6, \{M_2\}], [Z = 6, \{M_3\}]; \{A_2\} \rightarrow [G = 12, \{M_2, M_3, A_2\}]$

L'A.T.M.S. a essentiellement trois rôles :

Le calcul des environnements

Le propagateur envoie à l'ATMS les justifications (i.e. les valeurs et le modèle utilisés) ayant permis de prédire une valeur. Les justifications sont enregistrées et les environnements associés à la valeur prédite sont calculés avec la règle suivante : les environnements sont les unions des environnements associés aux justifications.

Par exemple,

$[v1, \{h1\}, \{h2\}], [v2; \{h3\}]; \{h4\} \rightarrow [v3, \{h1, h3, h4\}, \{h2, h3, h4\}]$

L'arrêt de la propagation :

Si une valeur prédite existe déjà alors l'ATMS évite la propagation de la nouvelle valeur. Seuls les environnements des différentes prédictions sont calculés (grâce à l'enregistrement des justifications).

La gestion des environnements de manière à obtenir uniquement les ensembles de conflits minimaux :

- L'ATMS supprime les environnements sur-ensembles des ensembles de conflits détectés.
- L'ATMS supprime les environnements sur-ensembles d'un des autres environnements appartenant au même label.

Par exemple, il supprime l'environnement $\{M_1\}$ de la prédiction $[A = 3, \{\}, \{M_1\}]$.

Le propagateur permet la création d'une base de données contenant les différentes valeurs prédites ou observées pour les différentes grandeurs. Détecter un conflit consiste non seulement à montrer que cette base de données est inconsistante, mais surtout à déterminer les prédictions qui provoquent cette inconsistance. Intuitivement, cette base de données est inconsistante lorsque les valeurs prédites ou observées d'une grandeur sont incohérentes entre

elles. Par exemple, supposons que l'on observe $F = 10$. Cette observation est incohérente avec la valeur prédite $F = 12$. De Kleer propose de noter la détection d'une incohérence comme étant la prédiction de la valeur FALSE. Par exemple, on a :

$$[F = 10, \{\}], [F = 12, \{A_1, M_1, M_2\}] \rightarrow [FALSE, \{A_1, M_1, M_2\}]$$

Les environnements associés à la prédiction FALSE sont des ensembles de conflits (appelés nogoods dans l'ATMS). Pour l'exemple proposé, l'ensemble de conflits détecté est $\{A_1, M_1, M_2\}$.

Nous présentons maintenant l'étape de prédiction des grandeurs lorsque les observations sont des SBT.

9.2 Prédiction de SBT

Il n'est pas question dans ce paragraphe de réaliser un propagateur. Nous nous limitons à décrire la procédure permettant de calculer (ou prédire) une SBT représentant une grandeur à partir

- de l'équation différentielle d'un modèle mentionnant cette grandeur,
- des SBT représentant les autres grandeurs du modèle.

La réalisation d'un propagateur nécessite d'intégrer cette procédure dans les algorithmes de propagation déjà définis.

Les notations définies dans GDE (plus précisément dans l'ATMS) sont reprises :

Les prédictions seront notées $[(\text{grandeur}, \text{ordre}, \text{valeur}), \text{environnement}(s)]$

où

- *grandeur* est le nom d'une grandeur élémentaire,
- *ordre* est un nombre entier correspondant à l'ordre de la grandeur (-1 pour une grandeur constante, 0 pour la grandeur, 1 pour la dérivée de cette grandeur, ..., i pour la dérivée i ème, ...)
- *valeur* est une SBT ou un intervalle encadrant la grandeur,
- *environnement* est un ensemble de composants correspondant à l'hypothèse supportant la prédiction. Une prédiction peut avoir plusieurs environnements¹.

Les prédictions seront notées :

valeurs utilisées ; modèle \rightarrow valeur prédite

Par exemple, la prédiction de $g_3^{(k)}(t)$ à partir des grandeurs $g_1^{(i)}(t)$ et $g_2^{(j)}(t)$ et du modèle de C_3 sera notée

$$[(g_1(t), i, SB_1), \{C_1\}], [(g_2(t), j, SB_2), \{C_2\}], \{C_3\} \rightarrow [(g_3(t), k, SB_3), C_1 \cup C_2 \cup C_3]$$

Nous présentons maintenant la procédure de calcul de la valeur d'une grandeur.

1. Toutefois, comme le fait remarquer [Dague et al, 90c], il est rare d'obtenir deux intervalles égaux. Par conséquent, l'obtention de plusieurs environnements pour une prédiction est une occurrence rare. La même remarque peut être réalisée pour les SBT.

9.2.1 Calcul des prédictions

La première étape de ce calcul consiste à réécrire l'équation d'un modèle de manière à obtenir une grandeur élémentaire en fonction des autres grandeurs élémentaires. Par exemple :

Pour le modèle d'une résistance : $(\{r\}, \mathbf{R}, (v_r(t) - r_v i_r(t)))$.

Les grandeurs élémentaires sont $v_r(t)$, r_v et $i_r(t)$.

L'équation du modèle

$$v_r(t) - r_v i_r(t) = 0$$

peut se réécrire :

$$v_r(t) = r_v i_r(t)$$

ou
$$i_r(t) = v_r(t) / r_v$$

ou
$$r_v = v_r(t) / i_r(t)$$

Pour le modèle d'une capacité : $(\{c\}, \mathbf{R}, (i_c(t), -c_v v_c(t)))$.

On a :
$$v_c'(t) = i_c(t) / c_v$$

$$i_c(t) = c_v v_c'(t)$$

$$c_v = i_c(t) / v_c'(t)$$

Nous supposons l'existence d'une procédure permettant de réécrire les équations des modèles de cette manière (lorsque cela est possible).

Nous présentons maintenant trois règles permettant de prédire les valeurs des grandeurs.

9.2.1.1 Règle 1 : principe fondamental

Le principe fondamental de la prédiction des grandeurs est que, lorsque deux grandeurs sont égales, toute SBT représentant l'une représente aussi l'autre. Ainsi, étant donné un modèle dont l'équation a été réécrite de la manière suivante

$$g_1^{(i)}(t) = op(g_2^{(j)}(t), g_3^{(k)}(t), \dots),$$

la connaissance des SBT SB_2, SB_3, \dots représentant les fonctions $g_2^{(j)}(t), g_3^{(k)}(t), \dots$

permet de déduire la SBT $SB_1 = OP(SB_2, SB_3, \dots)$ (où OP est l'OBS-extension de op) représentant la fonction $g_1^{(i)}(t)$.

Cette règle permet de prédire la valeur de bon nombre de grandeurs.

- Prédiction d'une grandeur (i.e. calcul d'une SBT représentant une grandeur) :

Exemple : prédiction du courant traversant une résistance à partir des encadrements de la tension aux bornes de et de la valeur de la résistance.

Soient SB_{v_r} la SBT représentant $v_r(t)$ et R l'intervalle encadrant r_v . On a :

$$v_r(t) / r_v \text{ Rep } (1 / R) * SB_{v_r}$$

A partir de l'équation

$$i_r(t) = v_r(t) / r_v$$

on prédit que

$$i_r(t) \text{ Rep } (1 / R) * SB_{v_r}$$

Cette opération sera écrite :

$$[(v_r(t), 0, SB_{v_r}), \{\}, [(r_v, -1, R), \{\}], \{r\} \rightarrow [(i_r(t), 0, (1 / R) * SB_{v_r}), \{r\}]$$

• Prédiction de la dérivée d'une grandeur :

Exemple : prédiction de la dérivée de la tension aux bornes d'une capacité à partir des encadrements du courant et de la valeur de cette capacité :

Soient SB_{i_c} la SBT représentant $i_c(t)$ et C l'intervalle encadrant c_v . On a :

$$i_c(t) / c_v \text{ Rep } SB_{i_c} * (1 / C)$$

et $v_c'(t) = i_c(t) / c_v$

implique que

$$v_c'(t) \text{ Rep } SB_{i_c} * (1 / C)$$

Soit

$$[(i_c(t), 0, SB_{i_c}), \{\}], [(c_v, -1, C), \{\}], \{c\} \rightarrow [(v_c(t), 1, (1 / C) * SB_{i_c}), \{c\}]$$

• Prédiction d'une grandeur constante au cours du temps (i.e. calcul de l'intervalle encadrant une grandeur constante) :

Exemple : Prédiction de la valeur d'une résistance à partir des SBT représentant le courant et la tension d'une résistance.

La représentation d'une grandeur constante au cours du temps est un intervalle. Pour réaliser le calcul de cet intervalle, deux approches sont possibles. La première consiste à considérer une grandeur constante comme une grandeur normale et à calculer une SBT représentant cette grandeur. L'intervalle est alors déduit à partir de la SBT calculée. La deuxième consiste à calculer directement l'intervalle encadrant cette valeur.

Passage par une SBT :

Soient SB_{v_r} et SB_{i_r} les SBT représentant les grandeurs $v_r(t)$ et $i_r(t)$. On a

$$v_r(t) / i_r(t) \text{ Rep } SB_{v_r} / SB_{i_r}$$

A partir de l'équation du modèle

$$r_v = v_r(t) / i_r(t)$$

on déduit

$$r_v \text{ Rep } SB_{v_r} / SB_{i_r}$$

La grandeur constante r_v est interprétée comme la fonction constante $r_v(t) = r_v$ pour tout instant t .

Supposons que la division de deux SBT soit définie comme une opération interne. Soit SB la SBT telle que $SB = SB_{v_r} / SB_{i_r}$. On a alors

$$r_v \text{ Rep } SB$$

Il reste alors à prendre en compte le fait que la fonction $r_v(t)$ est constante au cours du temps. On peut montrer que :

$$r_v \text{ Rep } SB \Rightarrow r_v \in [\max_{\theta \in T} p_i(t), \min_{\theta \in T} p_s(t)]$$

où T est l'extension temporelle de SB et $p_i(t)$, $p_s(t)$ sont les fonctions associées à SB .

La preuve est immédiate :

$$\text{On a } r_v \text{ Rep } SB \Leftrightarrow \forall t \in [t_i(SB), t_s(SB)] p_i(t) \leq r_v \leq p_s(t).$$

$$\text{Or, } \forall t \in [t_i(SB), t_s(SB)] p_i(t) \leq r_v \Leftrightarrow \max_{\theta \in T} p_i(t) \leq r_v$$

$$\text{et } \forall t \in [t_i(SB), t_s(SB)] r_v \leq p_s(t) \Leftrightarrow r_v \leq \min_{\theta \in T} p_s(t)$$

Graphiquement, l'intervalle encadrant une grandeur constante est la plus large bande temporelle d'ordre 0 incluse dans la SBT (i.e. la bande temporelle dont les polynômes d'ordre 0 sont $\max_{\theta \in T} p_i(t)$ et $\min_{\theta \in T} p_s(t)$).

La figure 9.2 illustre la plus large bande temporelle d'ordre 0 incluse dans une SBT.

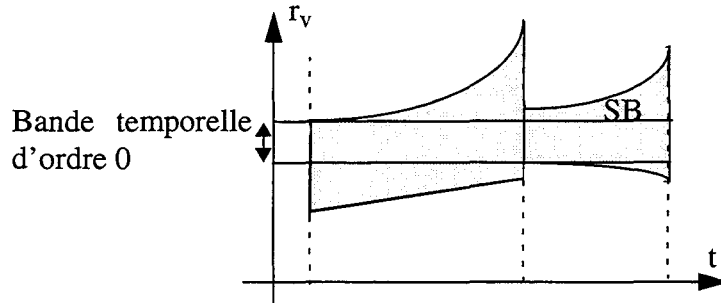


Figure 9.2 : Calcul de l'intervalle encadrant une grandeur constante

Avec cette méthode, la précision du calcul de l'intervalle encadrant une grandeur constante dépend de la précision du calcul de la SBT représentant cette grandeur. Pour l'exemple proposé, le calcul de l'intervalle encadrant la valeur de la résistance est imprécis car le calcul de la division de 2 SBT est imprécis car cette opération n'est pas une opération primitive. Nous présentons maintenant l'approche consistant à calculer directement l'intervalle. L'avantage de cette méthode est d'éviter le calcul parfois complexe et imprécis de la SBT encadrant la grandeur constante, tel que celui engendré par la division de deux SBT.

Calcul direct de l'intervalle :

Soit c une grandeur constante. A partir d'une équation et des valeurs des grandeurs invoquées dans cette équation, il est toujours possible de calculer deux fonctions $f_i(t)$ et $f_s(t)$ telles que

$$\forall t, f_i(t) \leq c \leq f_s(t)$$

L'intervalle encadrant la constante c est alors $[\max_{\theta \in T} f_i(t), \min_{\theta \in T} f_s(t)]$.

A partir de l'expression d'une fonction, il est parfois plus facile de calculer les extrema que la fonction elle-même. Par exemple, il est possible de calculer précisément l'intervalle issu de la division de deux SBT. Nous le montrons à l'aide du calcul de la valeur d'une résistance à partir de l'équation $r_v = v_r(t) / i(t)$. On a :

$$\forall t, r_v \geq f_i(t) = \min(p_{i_v}(t) / p_{i_i}(t), p_{i_v}(t) / p_{s_i}(t), p_{s_v}(t) / p_{i_i}(t), p_{s_v}(t) / p_{s_i}(t))$$

$$\text{et } r_v \leq f_s(t) = \max(p_{i_v}(t) / p_{i_i}(t), p_{i_v}(t) / p_{s_i}(t), p_{s_v}(t) / p_{i_i}(t), p_{s_v}(t) / p_{s_i}(t))$$

où $p_{i_v}(t)$, $p_{s_v}(t)$, $p_{i_i}(t)$ et $p_{s_i}(t)$ sont les fonctions associées à SB_{v_r} et SB_{i_r} .

Montrons que le calcul des extrema de $f_i(t)$ et $f_s(t)$ peut être réalisé directement.

Soit t_0 l'instant pour lequel $f_i(t)$ est maximum (resp. $f_s(t)$ est minimum). On peut montrer que si $f_i(t_0) = p_{i_v}(t_0) / p_{i_i}(t_0)$ (resp. $f_s(t_0) = p_{i_v}(t_0) / p_{i_i}(t_0)$) alors

- $p_{i_v}(t_0) / p_{i_i}(t_0)$ est un maximum local (resp. minimum local) de $p_{i_v}(t) / p_{i_i}(t)$ (i.e. t_0 est une des racines du polynôme $p_{i_v}'(t)p_{i_i}(t) - p_{i_v}(t)p_{i_i}'(t)$).
- $p_{i_v}(t_0) / p_{i_i}(t_0)$ est inférieur (resp. supérieur) aux trois autres fonctions à cet instant.

Le même raisonnement peut être tenu pour les 3 autres fonctions. D'où l'algorithme de calcul de l'intervalle encadrant la valeur d'une résistance.

Soit $T = \{t_1, \dots, t_n\}$ l'ensemble des extrema des fonctions $p_{i_v}(t) / p_{i_i}(t)$, $p_{i_v}(t) / p_{s_i}(t)$, $p_{s_v}(t) / p_{i_i}(t)$ ou $p_{s_v}(t) / p_{s_i}(t)$ (i.e. les racines des dérivées de ces fonctions).

L'algorithme consiste à :

Calculer les réels

$$m_i = \min(p_{i_v}(t_i) / p_{i_i}(t_i), p_{i_v}(t_i) / p_{s_i}(t_i), p_{s_v}(t_i) / p_{i_i}(t_i), p_{s_v}(t_i) / p_{s_i}(t_i))$$

$$\text{et } M_i = \max(p_{i_v}(t_i) / p_{i_i}(t_i), p_{i_v}(t_i) / p_{s_i}(t_i), p_{s_v}(t_i) / p_{i_i}(t_i), p_{s_v}(t_i) / p_{s_i}(t_i))$$

pour chaque t_i de T .

La borne inférieure de l'intervalle encadrant r_v (i.e. le maximum de $f_i(t)$) est le maximum des m_i et la borne supérieure est le minimum des M_i .

Remarque :

Les grandeurs constantes sont en général les paramètres des composants. Ces grandeurs interviennent dans l'équation d'un unique modèle. Par conséquent, ces valeurs ne sont pas propagées. Toutefois, leur calcul a deux intérêts : le premier est de permettre la prédiction de meilleures valeurs pour les autres grandeurs et le deuxième est de permettre la détection d'ensembles de conflits grâce à la prise en compte du caractère constant de ces paramètres (cf. §8.3.3.3).

Le principe fondamental est sans aucun doute la règle la plus importante. Le résultat intéressant est que **cette règle est suffisante pour prédire les valeurs des grandeurs des systèmes décrit par des modèles d'ordre 1 et que deux modèles ne contiennent jamais une même grandeur à des ordres différents.** Ceci est le cas du circuit RC illustré figure 9.3.

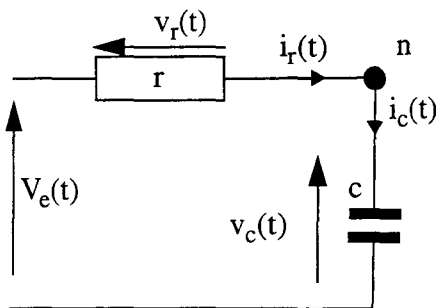


Figure 9.3 : Circuit RC

Le comportement de ce système peut être décrit par un modèle d'ordre 1 :

$$(\{r, c, n\}, \mathbf{R}, (v_r(t), -r_v c_v v_c(t)))$$

A partir de ce modèle, il est possible de prédire $v_c'(t)$ à partir de $v_r(t)$. En effet,

$$[(v_r(t), 0, SB_{v_r}), \{\}], [(r_v, -1, R), \{\}], [(c_v, -1, C), \{\}]; \{r, c, n\} \rightarrow [(v_c(t), 1, (1/RC) * SB_{v_r}), \{r, n, c\}]$$

Cette prédiction est aussi réalisable à partir des modèles des composants :

$$(\{c\}, \mathbf{R}, (i_c(t), -c_v v_c(t)))$$

$$(\{r\}, \mathbf{R}, (v_r(t) - r_v i_r(t)))$$

$$(\{n\}, \mathbf{R}, (i_r(t) - i_c(t)))$$

La prédiction de $v_c'(t)$ est alors obtenue par étapes successives :

$$[(v_r(t), 0, SB_{v_r}), \{\}], [(r_v, -1, R), \{\}]; \{r\} \rightarrow [(i_r(t), 0, (1/R) * SB_{v_r}), \{r\}]$$

$$[(i_r(t), 0, (1/R) * SB_{v_r}), \{r\}]; \{n\} \rightarrow [(i_c(t), 0, (1/R) * SB_{v_r}), \{r, n\}]$$

$$[(i_c(t), 0, (1/R) * SB_{v_r}), \{r, n\}], [(c_v, -1, C), \{\}]; \{c\} \rightarrow [(v_c(t), 1, (1/RC) * SB_{v_r}), \{r, n, c\}]$$

On remarque qu'aucune prédiction ne peut être réalisée à partir de l'observation de $v_c(t)$.

Nous présentons maintenant les deux autres règles permettant de prédire des valeurs.

9.2.1.2 Règles 2 & 3 : règles internes

- Règle 2 : dérivée de l'équation d'un modèle

Soit un modèle dont l'équation est $g_1(t) = g_2(t)$ et SB une SBT représentant $g_2'(t)$.

La déduction de $g_1'(t)$ nécessite la dérivation de l'équation du modèle. Cette opération est toujours réalisable. En effet, on a :

$$g_1(t) = g_2(t) \Rightarrow g_1'(t) = g_2'(t)$$

A partir de l'équation obtenue, on déduit alors que $g_1'(t)$ est représentée par SB.

Nous présentons un exemple d'utilisation de la règle de dérivation des modèles. Soit le circuit LC illustré figure 9.4.

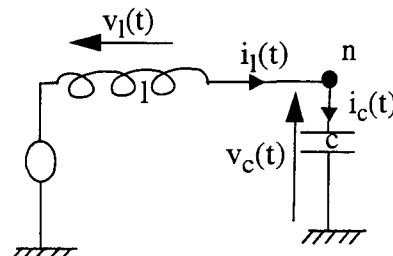


Figure 9.4 : Circuit LC

Les modèles des différents composants de ce circuit sont tous d'ordre 0 ou 1. Soient

$$(\{l\}, \mathbf{R}, (v_l(t), -l_v i_l(t)))$$

$$(\{c\}, \mathbf{R}, (i_c(t), -c_v v_c(t)))$$

$$(\{n\}, \mathbf{R}, (i_l(t) - i_c(t)))$$

Toutefois, le principe fondamental n'est pas suffisant car ce circuit peut être décrit par un modèle d'ordre 2. A partir de l'observation de $v_l(t)$, nous allons montrer que les valeurs des autres grandeurs peuvent être prédites.

Déduisons d'abord le courant traversant la self :

$$[(v_l(t), 0, SB_{v_l}), \{\}], [(l_v, -1, L), \{\}]; \{l\} \rightarrow [(i_l(t), 1, (1/L) * SB_{v_l}), \{l\}]$$

En dérivant l'équation du modèle du noeud (i.e. $i_c'(t) - i_l'(t)$), on obtient :

$$[(i_l(t), 1, (1/L) * SB_{v_l}), \{l\}]; \{n\} \rightarrow [(i_c(t), 1, (1/L) * SB_{v_l}), \{l, n\}]$$

De même, en dérivant l'équation du modèle de la capacité (i.e. $i_c'(t) = c_v v_c''(t)$), on obtient :

$$[(i_c(t), 1, (1/L) * SB_{v_l}), \{l, n\}], [(c_v, -1, C), \{\}]; \{c\} \rightarrow [(v_c(t), 2, (1/LC) * SB_{v_l}), \{l, n, c\}]$$

- Règle 3 : calcul de la valeur d'une grandeur à partir de l'encadrement de sa dérivée.

Soit un modèle dont l'équation est $g_1(t) = g_2'(t) + g_3(t)$

et SB_2, SB_3 deux SBT représentant respectivement $g_2'(t)$ et $g_3'(t)$.

La prédiction de $g_1(t)$ nécessite la connaissance des SBT représentant $g_2'(t)$ et $g_3(t)$. Dans ce cas, il est alors possible d'intégrer la SBT pour prédire un encadrement de $g_3(t)$ à partir de celui de $g_3'(t)$. Cette prédiction est réalisée de la manière suivante :

Par intégration, une grandeur peut s'exprimer en fonction de sa dérivée :

$$g(t) = g(\theta) + \int_{\theta}^t g'(x)dx$$

Il découle directement du principe fondamental et de cette équation qu'un encadrement d'une grandeur $g(t)$ est calculable à partir de la connaissance :

- d'un encadrement de la dérivée de cette grandeur (i.e. une SBT),
- d'un encadrement de cette grandeur en un instant (i.e. un intervalle).

Soit SB' une SBT représentant $g'(t)$ et I un intervalle encadrant $g(\theta)$. On a :

$$g(t) \text{ Rep } [I + \int_{\theta} SB']$$

Le choix du point d'intégration (i.e. de l'instant θ) dépend de la connaissance de $g(t)$ à cet instant.

- Si $g(t)$ est connu en un nombre infini d'instants (i.e. on dispose d'une SBT représentant $g(t)$) alors il est nécessaire de vérifier la cohérence des SBT représentant $g(t)$ et $g'(t)$. Nous montrerons que la procédure de détection d'incohérence permet d'éviter l'utilisation de cette règle.

- Si $g(t)$ est connu en un nombre fini d'instants (i.e. $g(t)$ provient du calcul de la dérivée d'une grandeur) alors une prédiction est réalisée pour chacun de ces instants. Cette règle est donc utile uniquement lorsque l'on dispose d'une procédure permettant de déduire (et donc de propager) la dérivée d'une grandeur à partir de son observation.

Remarque :

Lorsqu'une prédiction nécessite la dérivation d'un modèle (règle 2), l'opération d'intégration (règle 3) peut aussi être utilisée. Toutefois, il semble préférable d'utiliser la dérivation car cela permet de retarder l'introduction d'imprécision due à l'intégration d'une SBT.

Nous présentons maintenant un exemple d'utilisation de la troisième règle. Soit le circuit RLC illustré figure 9.5.

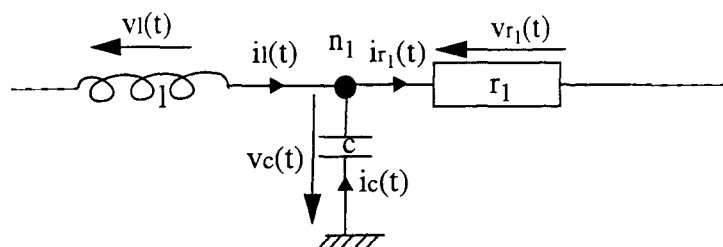


Figure 9.5 : Circuit RLC

Les modèles des différents composants de ce système sont :

$$\begin{aligned} & (\{1\}, \mathbf{R}, (v_l(t), -l_v i_l(t))) \\ & (\{c\}, \mathbf{R}, (i_c(t), -c_v v_c(t))) \\ & (\{r_1\}, \mathbf{R}, (v_{r_1}(t) - r_{1v} i_{r_1}(t))) \\ & (\{n_1\}, \mathbf{R}, (i_l(t) + i_c(t) - \dot{i}_1(t))) \end{aligned}$$

Supposons que l'on observe $v_{r_1}(t)$ et $v_l(t)$. A partir des règles 1 & 2, on obtient :

$$\begin{aligned} & [(v_{r_1}(t), 0, SB_{v_{r_1}}), \{\}], [(r_{1v}, -1, R_1), \{\}]; \{r_1\} \rightarrow [(i_{r_1}(t), 0, (1/R_1) * SB_{v_{r_1}}), \{r_1\}] \\ & [(v_l(t), 0, SB_{v_l}), \{\}], [(l_v, -1, L), \{\}]; \{1\} \rightarrow [(i_l(t), 1, (1/L) * SB_{v_l}), \{1\}] \end{aligned}$$

La prédiction de $i_c(t)$ à partir du modèle du noeud n_1 , (i.e. $i_c(t) = i_{r_1}(t) - i_l(t)$) nécessite la connaissance des valeurs de $i_{r_1}(t)$ et de $i_l(t)$ pour des ordres identiques. Or, nous disposons uniquement de ces valeurs pour des ordres différents (i.e. on a $i_{r_1}(t)$ et $i_l'(t)$). La règle d'intégration permet de diminuer l'ordre d'une grandeur. On a :

$$i_l(t) = i_l(\theta) + \int_{\theta}^t i_l'(x) dx$$

La déduction de $i_l(t)$ en tout instant t nécessite la connaissance de la valeur de cette grandeur en un instant θ . Nous supposons l'existence d'une procédure capable de déduire cette valeur à partir des modèles et d'une procédure de calcul de la dérivée d'une grandeur observée. Par exemple, à partir des modèles, il est possible de déduire que $i_l(\theta) = v_{r_1}(\theta) / r_{1v} - c_v v_c'(\theta)$. Le calcul de $v_c'(\theta)$ est réalisé à partir de l'observation de $v_c(t)$ (cette prédiction sera notée $[(v_c'(\theta), -1, I), \{\}])$. Il découle alors que

$$[(i_l(\theta), -1, I_i), \{c, r_1, n_1\}] \text{ où } I_i = (1/R_1) * SB_{v_{r_1}}(\theta) - C * I$$

Ayant une valeur pour $i_l(t)$ en un instant θ , la règle d'intégration nous permet d'obtenir la prédiction de $i_l(t)$ pour tout instant t :

$$[(i_l(t), 1, SB_{v_l}/L); \{1\}] \rightarrow [(i_l(t), 0, SB_{i_l}), \{r_1, l, c, n_1\}] \text{ où } SB_{i_l} = I_i + \int_{\theta}^t (1/L) * SB_{v_l}$$

En continuant la propagation, on obtient alors :

$$\begin{aligned} & [(i_l(t), 0, SB_{i_l}), \{r_1, l, c, n_1\}], [(i_{r_1}(t), 0, (1/R_1) * SB_{v_{r_1}}), \{r_1\}]; \{n_1\} \\ & \rightarrow [(i_c(t), 0, SB_{i_c}), \{r_1, l, c, n_1\}] \text{ où } SB_{i_c} = (1/R_1) * SB_{v_{r_1}} - SB_{i_l} \end{aligned}$$

et

$$[(i_c(t), 0, SB_{i_c}), \{r_1, l, c, n_1\}], [(c_v, -1, C), \{\}]; \{c\} \rightarrow [(v_c(t), 1, (1/C) * SB_{i_c}), \{r_1, l, c, n_1\}]$$

En conclusion, les observations de $v_{r_1}(t)$, $v_l(t)$ permettent de prédire $i_{r_1}(t)$, $i_l'(t)$. La prédiction de $v_c'(t)$ en tout instant t nécessite la prédiction de cette grandeur en un instant θ .

Procédure de calcul de $i_l(\theta)$:

Pour calculer $i_l(\theta)$, plusieurs possibilités sont envisageables. La première, facile à mettre en oeuvre, consiste à toujours calculer et propager les dérivées des observations. Ainsi, en calculant la dérivée de $v_c(t)$ et en la propageant à travers les modèles de c , de r_1 et de n_1 , une prédiction de $i_l(t)$ est obtenue. Toutefois, cette solution ne semble pas envisageable car elle nécessite un nombre de calculs et de propagations trop important (car inutiles). Par exemple, le

calcul et la propagation des dérivées de $v_{r_1}(t)$ et de $v_l(t)$ sont inutiles (i.e. n'apportent aucune information pour le diagnostic). La deuxième solution consiste à établir le raisonnement suivant : étant donné que la prédiction de $i_c(t)$ nécessite un encadrement de $i_l(t)$ en un instant θ (c'est lors de cette prédiction que la règle 3 est nécessaire), inversons le raisonnement et essayons de calculer $i_l(\theta)$ à partir de $i_c(t)$. Cette prédiction est alors conservée car elle est réalisée à partir de l'encadrement de la dérivée d'une grandeur. On remarque alors qu'en remplaçant la capacité par une résistance, ce calcul n'est pas conservé car la valeur du courant est alors obtenue en tout instant (ce calcul est donc inutile).

L'algorithme permettant de prédire les valeurs des grandeurs découle de ces 3 règles.

9.2.1.3 Algorithme de calcul des prédictions

Prédire la valeur de $g(t)$ à partir de l'équation $g(t) = op(g_1^{(i_1)}(t), \dots, g_k^{(i_k)}(t))$ et des prédictions des valeurs $g_1^{(j_1)}(t), \dots, g_k^{(j_k)}(t)$ consiste à :

- Si les ordres des grandeurs observées sont identiques aux ordres des grandeurs de l'équation (i.e. $i_1 = j_1, \dots, i_k = j_k$), alors le calcul de $g(t)$ découle directement du principe fondamental (i.e. $g(t)$ est représentée par $OP(SB_1, \dots, SB_k)$ où OP est l'extension aux SBT de op).
- Si les différences entre les ordres des grandeurs observées et les ordres des grandeurs de l'équation sont toutes identiques (i.e. $i_1 - j_1 = \dots = i_k - j_k = n$) alors, après n dérivation(s) des valeurs, le calcul de $g(t)$ découle directement du principe fondamental.
- Sinon, intégrer ou dériver les grandeurs jusqu'à ce que les différences entre les ordres des grandeurs observées et les ordres des grandeurs de l'équation soient nulles. Le calcul découle alors directement des règles précédentes.

Cette procédure de calcul s'inscrit dans une procédure de propagation dont nous rappelons uniquement le principe. Propager une grandeur consiste à

- récupérer les modèles invoquant cette grandeur,
- calculer les valeurs des autres grandeurs invoquées dans les modèles à partir des valeurs disponibles,
- propager les grandeurs prédites.

Deux variantes sont possibles : déduire d'abord toutes les valeurs possibles avec tous les modèles (algorithme en largeur d'abord) ou propager les valeurs issues d'un modèle, puis calculer les autres valeurs à partir des autres modèles (algorithme en profondeur d'abord).

Nous présentons maintenant la procédure de détection des conflits.

9.3 Détection des conflits

Le propagateur permet d'obtenir une base de données contenant des prédictions de la forme $[(g(t), i, SB), C]$. Nous nous intéressons dans ce paragraphe à la vérification de la cohérence d'un ensemble de prédictions d'une même grandeur (les ordres pouvant être différents). Nous montrons dans un premier temps que confronter deux prédictions (i.e. vérifier leur cohérence) revient à vérifier la consistance d'un modèle. Nous étendons ensuite ces résultats à un ensemble de prédictions.

9.3.1 Confrontation de 2 prédictions

Les procédures de vérification de la consistance d'un modèle dépendent de l'ordre du modèle. De la même manière, nous différencions les confrontations selon l'ordre des grandeurs.

9.3.1.1 Confrontation de deux intervalles encadrant une grandeur constante

Soient les prédictions $[(c, -1, I_1), C_1]$ et $[(c, -1, I_2), C_2]$.

Ces valeurs sont cohérentes si et seulement si la constante c peut appartenir aux deux intervalles I_1 et I_2 , c'est-à-dire si ces deux intervalles s'intersectent.

Une formalisation par l'intermédiaire de 2 règles est proposée par [Dague et al, 91] :

Règle d'intersection :

$$[(c, -1, I_1), C_1], [(c, -1, I_2), C_2] \rightarrow [(c, -1, I_1 \cap I_2), C_1 \cup C_2]$$

Règle d'incohérence :

$$[(c, -1, \emptyset), C] \rightarrow C \text{ ensemble de conflits}$$

(Le noeud FALSE est le noeud où la valeur d'une grandeur est l'ensemble vide)

Remarque :

On peut montrer que les prédictions $[(c, -1, [a_1, b_1]), C_1]$ et $[(c, -1, [a_2, b_2]), C_2]$ sont cohérentes si et seulement si tout modèle d'ordre 0 est consistant pour une SBT dont les fonctions associées sont les constantes $a_1 - b_2$ et $b_1 - a_2$.

9.3.1.2 Confrontation de deux SBT représentant une grandeur

Soient les prédictions $[(g(t), i, SB_1), C_1]$ et $[(g(t), i, SB_2), C_2]$.

Ces valeurs sont cohérentes si et seulement si la grandeur $g^{(i)}(t)$ est représentée par les deux SBT SB_1 et SB_2 , c'est à dire si ces deux SBT s'intersectent.

Les règles établies dans le cas des grandeurs constantes peuvent être étendues aux grandeurs quelconques.

Règle d'intersection :

$$[(g(t), i, SB_1), C_1], [(g(t), i, SB_2), C_2] \rightarrow [(g(t), i, SB_1 \cap SB_2), C_1 \cup C_2]$$

Règle d'incohérence :

$$[(g(t), i, \emptyset), C] \rightarrow C \text{ ensemble de conflits}$$

(Le noeud FALSE est le noeud pour lequel la valeur d'une grandeur est l'ensemble vide)

Il découle directement de ces règles que, pour vérifier la cohérence de 2 SBT, il suffit de calculer l'intersection de 2 SBT, puis de vérifier que la SBT obtenue n'est pas vide.

Une autre méthode découle directement de la procédure de détection d'inconsistance des modèles d'ordre 0. En effet, on peut montrer que **les prédictions $[(g(t), i, SB_1), C_1]$ et $[(g(t), i, SB_2), C_2]$ sont cohérentes si et seulement si tout modèle d'ordre 0 est consistant pour $(SB_2 - SB_1)$.**

La preuve est immédiate : il suffit de prendre comme fonction du modèle la fonction $f(t) =$

$g^{(i)}(t) - g^{(i)}(t)$. La preuve découle alors directement du fait que la soustraction de 2 SBT n'introduit pas d'imprécision.

Pour vérifier la cohérence de 2 prédictions, l'utilisation de la procédure de détection d'inconsistance a 2 avantages. Pour les présenter, nous comparons ces 2 procédures :

- Le calcul de l'intersection de deux SBT augmente le nombre de bandes temporelles des suites. Si SB_1 et SB_2 contiennent n bandes temporelles et sont de degré p alors le nombre de bandes temporelles de $SB_1 \cap SB_2$ est, au pire, $n * (2p + 1)$. De plus, l'algorithme de calcul de l'intersection a des solutions analytiques uniquement pour les SBT d'ordre inférieur ou égal à 2. En résumé, l'algorithme vérifiant l'intersection de deux SBT
 - est en $O(n * p)$,
 - a des solutions analytiques pour les SBT d'ordre inférieur ou égal à 2.
- L'algorithme de détection de l'inconsistance de 2 SBT nécessite la soustraction de 2 SBT. Cette opération n'augmente ni le nombre de bandes temporelles, ni l'ordre des SBT. Par conséquent cet algorithme
 - est en $O(n)$,
 - a des solutions analytiques pour les SBT d'ordre inférieur ou égal à 3 (il faut déterminer les zéros de la différence entre les fonctions associées aux SBT).

Ce résultat est logique. Vérifier la consistance d'un modèle d'ordre 0 revient à vérifier que les SBT s'intersectent. Or, il est évident qu'un algorithme vérifiant uniquement l'intersection est meilleur que celui la calculant.

9.3.1.3 Confrontation de deux SBT représentant une grandeur et sa dérivée

Soient les prédictions $[(g(t), i, SB_1), C_1]$ et $[(g(t), i+1, SB_2), C_2]$.

Ces valeurs sont cohérentes si et seulement si la grandeur $g^{(i)}(t)$ et sa dérivée (i.e. $g^{(i+1)}(t)$) sont respectivement représentées par SB_1 et SB_2 .

On peut montrer que ces prédictions sont cohérentes si et seulement si tout modèle d'ordre 1 ($C_1 \cup C_2, T, (f_0(t), f_1(t))$) est consistant pour le couple $(SB_2, -SB_1)$.

La preuve est immédiate : il suffit de prendre $f_0(t) = g^{(i+1)}(t)$ et $f_1(t) = -g^{(i)}(t)$.

Il découle directement du théorème 8.1 que, lorsque les prédictions sont cohérentes, une nouvelle prédiction de $g^{(i)}(t)$ est obtenue. Nous noterons $[(g(t), i, \text{cons}(SB_1, SB_2)), C_1 \cup C_2]$ cette prédiction avec $\text{cons}(SB_1, SB_2) = D(-SB_2, SB_1) \cap F(-SB_2, SB_1) \cap SB_1$.

D'où la règle :

Règle d'intersection de deux SBT représentant une grandeur et sa dérivée :

$[(g(t), i, SB_1), C_1], [(g(t), i+1, SB_2), C_2] \rightarrow [(g(t), i, \text{cons}(SB_1, SB_2)), C_1 \cup C_2]$

Remarque :

La SBT $\text{cons}(SB_1, SB_2)$ est toujours un meilleur encadrement de $g^{(i)}(t)$ que SB_1 (i.e. on a toujours $\text{cons}(SB_1, SB_2) \subseteq SB_1$).

Une remarque intéressante est que $\text{cons}(\text{SB}_1, \text{SB}_2)$ est la SBT obtenue en appliquant la troisième règle de propagation (calcul d'une grandeur à partir de l'encadrement de sa dérivée et de la valeur de cette grandeur en un instant) avec une SBT comme encadrement de la grandeur. Reprenons le circuit illustré figure 9.5 en rajoutant une résistance en série avec la self (cf. figure 9.6).

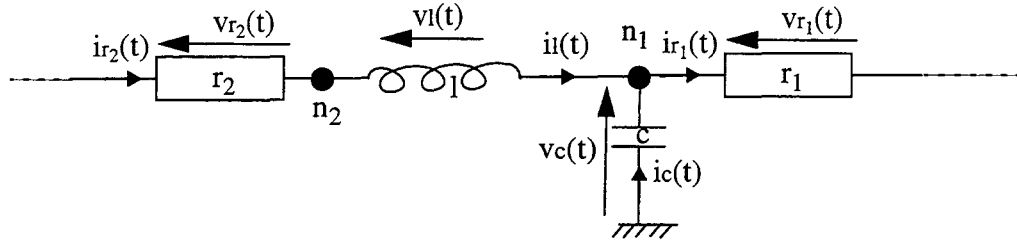


Figure 9.6 : Circuit RLC

Les observations de $v_{r1}(t)$ et de $v_l(t)$ permettent de prédire $i_{r1}(t)$ et $i_l'(t)$.

$$[(v_{r1}(t), 0, \text{SB}_{v_{r1}}), \{\}], [(i_{r1}(t), 0, (1/R_1) * \text{SB}_{v_{r1}}), \{r_1\}] \rightarrow [(i_{r1}(t), 0, (1/R_1) * \text{SB}_{v_{r1}}), \{r_1\}]$$

$$[(v_l(t), 0, \text{SB}_{v_l}), \{\}], [(i_l'(t), 1, (1/L) * \text{SB}_{v_l}), \{1\}] \rightarrow [(i_l'(t), 1, (1/L) * \text{SB}_{v_l}), \{1\}]$$

La troisième règle de propagation est utilisée pour prédire $i_c(t)$ à partir du modèle du noeud n_1 .

On a :

$$i_l(t) = i_l(\theta) + \int_{\theta}^t i_l'(x) dx$$

On a vu que $i_l(\theta)$ pouvait être calculé à partir de la dérivée de $v_c(t)$. Toutefois, cette valeur peut aussi être prédite à partir de l'observation de $v_{r2}(t)$. En effet,

$$[(v_{r2}(t), 0, \text{SB}_{v_{r2}}), \{\}], [(i_{r2}(t), 0, (1/R_2) * \text{SB}_{v_{r2}}), \{r_2\}] \rightarrow [(i_{r2}(t), 0, (1/R_2) * \text{SB}_{v_{r2}}), \{r_2\}]$$

$$[(i_{r2}(t), 0, (1/R_2) * \text{SB}_{v_{r2}}), \{r_2\}], \{n_2\} \rightarrow [(i_l(t), 0, (1/R_2) * \text{SB}_{v_{r2}}), \{r_2, n_2\}]$$

En confrontant les prédictions de $i_l'(t)$ et de $i_l(t)$ (obtenues respectivement à partir des observations de $v_l(t)$ et de $v_{r2}(t)$), on obtient la nouvelle prédiction :

$$[(i_l(t), 0, \text{cons}((1/R_2) * \text{SB}_{v_{r2}}, (1/L) * \text{SB}_{v_l}), \{r_2, n_2\})]$$

Cette prédiction est celle qui aurait été obtenue en utilisant la règle 3.

9.3.1.4 Généralisation

Soient les prédictions $[(g(t), i, \text{SB}_1), C_1]$ et $[(g(t), i+p, \text{SB}_2), C_2]$.

Ces prédictions sont cohérentes si et seulement si le modèle d'ordre p ($C_1 \cup C_2, T, (f_0(t), 0, \dots, 0, f_p(t))$) est consistant pour $(\text{SB}_2, 0, \dots, 0, -\text{SB}_1)$.

De la même manière que pour la confrontation d'une grandeur et de sa dérivée, une nouvelle prédiction pour $g^{(i)}(t)$ peut être obtenue. En effet, vérifier la consistance d'un modèle d'ordre p revient, après $p - 1$ intégrations, à vérifier la consistance d'un modèle d'ordre 1.

Nous noterons $\text{cons}(\text{SB}_1, \text{SB}_2; p)$ la SBT obtenue.

D'où la règle :

$$[(g(t), i, \text{SB}_1), C_1], [(g(t), i+p, \text{SB}_2), C_2] \rightarrow [(g(t), i, \text{cons}(\text{SB}_1, \text{SB}_2; p), C_1 \cup C_2)]$$

Il découle directement de la procédure de détection d'inconsistance d'un modèle d'ordre p que

le calcul de $\text{cons}(\text{SB}_1, \text{SB}_2 ; p)$ nécessite la connaissance des valeurs des dérivées de la fonction $g(t)$ en certains instants. Pour simplifier l'écriture, nous n'avons pas fait apparaître ces données dans la règle (bien que ceci devrait être le cas).

Les procédures de détection d'inconsistance des modèles permettent de confronter 2 valeurs. Nous étudions maintenant la confrontation de n valeurs.

9.3.2 Confrontation de n valeurs

Par propagation, k prédictions pour une grandeur peuvent être obtenues. Comme l'algorithme est incrémental, nous étudions la cohérence d'une prédiction vis à vis d'un ensemble contenant k prédictions dont la cohérence a déjà été vérifiée¹. De la même manière que précédemment, nous étudions les cas selon l'ordre des grandeurs.

9.3.2.1 Confrontation de k intervalles encadrant une grandeur constante

Soient k prédictions cohérentes entre elles $[(c, -1, I_1), C_1], [(c, -1, I_2), C_2], \dots, [(c, -1, I_k), C_k]$ et une nouvelle prédiction $[(c, -1, I), C]$.

En appliquant k fois la règle d'intersection, on obtient :

$$\begin{aligned} & [(c, -1, I), C], [(c, -1, I_1), C_1] \rightarrow [(c, -1, I_1 \cap I), C_1 \cup C] \\ & [(c, -1, I_1 \cap I), C_1 \cup C], [(c, -1, I_2), C_2] \rightarrow [(c, -1, I_1 \cap I_2 \cap I), C_1 \cup C_2 \cup C] \\ & \dots \\ & [(c, -1, I_1 \cap \dots \cap I_{k-1} \cap I), C_1 \cup \dots \cup C_{k-1} \cup C], [(c, -1, I_k), C_k] \\ & \rightarrow [(c, -1, I_1 \cap \dots \cap I_k \cap I), C_1 \cup \dots \cup C_k \cup C] \end{aligned}$$

Il découle alors directement de la règle d'incohérence que

- Les prédictions sont cohérentes lorsque $I_1 \cap \dots \cap I_k \cap I \neq \emptyset$. Aucun ensemble de conflits ne peut alors être détecté.
- Les prédictions sont incohérentes lorsque $I_1 \cap \dots \cap I_k \cap I = \emptyset$. L'ensemble de conflits détecté est $C_1 \cup \dots \cup C_k \cup C$. Le problème est alors de déterminer le ou les ensembles de conflits minimaux. La propriété suivante permet de caractériser les ensembles de conflits minimaux.

Propriété 9.1 : "Intersection de k intervalles"²

L'intersection de k intervalles est égale à l'intersection de deux des intervalles.

D'une manière formelle :

$$\exists i, j = 1 \dots k \text{ tels que } I_1 \cap \dots \cap I_k = I_i \cap I_j.$$

1. Cette démarche revient à réaliser une démonstration récursive dont le premier cas est la confrontation de 2 grandeurs.
2. Cette propriété est déjà exprimée dans [Dague et al, 91].

Preuve :

Soient les intervalles $[a_1, b_1], \dots, [a_k, b_k]$.

Par définition de l'intersection, on a

$$[a_1, b_1] \cap \dots \cap [a_k, b_k] = [\max(a_1, \dots, a_k), \min(b_1, \dots, b_k)]$$

Or, $\exists i \in [1, k]$ tel que $\max(a_1, \dots, a_k) = a_i$

et $\exists j \in [1, k]$ tel que $\min(b_1, \dots, b_k) = b_j$

$$D'où [a_1, b_1] \cap \dots \cap [a_k, b_k] = [a_i, b_i] \cap [a_j, b_j] = [a_i, b_j]$$

□

Il découle directement de cette propriété que les ensembles de conflits minimaux sont l'union de deux environnements (i.e. $C_i \cup C$). D'où l'algorithme :

Vérifier la cohérence de la prédiction $[(c, -1, I), C]$ vis-à-vis des k prédictions cohérentes $[(c, -1, I_i), C_i]$ ($i \in [1, k]$) consiste à montrer que les intervalles I et I_i ($i \in [1, k]$) s'intersectent. En cas de non intersection, $C \cup C_i$ est un ensemble de conflits minimal.

9.3.2.2 Confrontation de k SBT représentant une grandeur

Soient k prédictions cohérentes entre elles $[(g(t), i, SB_1), C_1], [(g(t), i, SB_2), C_2], \dots, [(g(t), i, SB_k), C_k]$ et une nouvelle prédiction $[(g(t), i, SB), C]$.

Nous allons montrer que la confrontation de k SBT est similaire à la confrontation de k intervalles. En appliquant $k - 1$ fois la règle d'intersection, on obtient la prédiction $[(g(t), i, SB_1 \cap \dots \cap SB_k \cap SB), C_1 \cup \dots \cup C_k \cup C]$

Cette prédiction est cohérente lorsque la SBT $SB_1 \cap \dots \cap SB_k \cap SB$ n'est pas vide. Or, vérifier si k SBT s'intersectent revient à vérifier ces intersections deux à deux (proposition 8.3). D'où l'algorithme (identique à celui des intervalles) :

Vérifier la cohérence de la prédiction $[(g(t), i, SB), C]$ vis-à-vis des k prédictions cohérentes $[(g(t), i, SB_i), C_i]$ consiste à vérifier que les SBT SB et SB_i ($i \in [1, k]$) s'intersectent (i.e. sont cohérentes). En cas de non intersection, $C \cup C_i$ est un ensemble de conflits minimal.

La confrontation des grandeurs variant au cours du temps est donc similaire à celle des grandeurs constantes : il faut vérifier la cohérence des prédictions (i.e. l'intersection des SBT) 2 à 2. Toutefois, contrairement aux grandeurs constantes, il semble préférable d'utiliser l'algorithme de détection d'inconsistance d'un modèle d'ordre 0 pour vérifier la cohérence de 2 SBT.

Nous étudions maintenant trois cas de confrontations de grandeurs d'ordre différent.

9.3.2.3 Confrontation de k prédictions d'une grandeur et d'une prédiction de la dérivée de cette grandeur

Nous présentons d'abord le cas $k = 2$. Une généralisation est ensuite proposée.

Soient les prédictions $[(g(t), i, SB_1), C_1]$, $[(g(t), i, SB_2), C_2]$ et $[(g(t), i + 1, SB_3), C_3]$

Ces prédictions sont cohérentes si et seulement si la grandeur $g^{(i)}(t)$ peut être représentée par SB_1 et SB_2 et la dérivée de cette grandeur (i.e. $g^{(i+1)}(t)$) peut être représentée par SB_3 .

Appliquons les règles d'intersection, on obtient :

$$[(g(t), i, SB_1), C_1], [(g(t), i, SB_2), C_2] \rightarrow [(g(t), i, SB_{12}), C_1 \cup C_2] \text{ où } SB_{12} = SB_1 \cap SB_2$$

$$[(g(t), i, SB_1), C_1], [(g(t), i + 1, SB_3), C_3] \rightarrow [(g(t), i, \text{cons}(SB_1, SB_3)), C_1 \cup C_3]$$

$$[(g(t), i, SB_2), C_2], [(g(t), i + 1, SB_3), C_3] \rightarrow [(g(t), i, \text{cons}(SB_2, SB_3)), C_2 \cup C_3]$$

mais aussi

$$[(g(t), i, SB_{12}), C_1 \cup C_2], [(g(t), i+1, SB_3), C_3] \rightarrow [(g(t), i, \text{cons}(SB_{12}, SB_3)), C_1 \cup C_2 \cup C_3]$$

Il découle alors directement de la règle d'incohérence que

- si $\text{cons}(SB_{12}, SB_3) \neq \emptyset$ alors les 3 prédictions sont cohérentes¹
- sinon
 - si $SB_{12} = SB_1 \cap SB_2 = \emptyset$ alors $C_1 \cup C_2$ est un ensemble de conflits minimal (car il provient de 2 prédictions)
 - si $\text{cons}(SB_1, SB_3) = \emptyset$ alors $C_1 \cup C_3$ est un ensemble de conflits minimal.
 - si $\text{cons}(SB_2, SB_3) = \emptyset$ alors $C_2 \cup C_3$ est un ensemble de conflits minimal.
 - si $\text{cons}(SB_{12}, SB_3) = \emptyset$ alors $C_1 \cup C_2 \cup C_3$ est un ensemble de conflits.

Dans le cas des prédictions pour une grandeur de même ordre, une règle permet d'éviter la confrontation de 3 grandeurs (car l'union de 3 environnements n'est jamais un ensemble de conflits minimal). Ce n'est malheureusement pas le cas lorsque les prédictions correspondent à des ordres différents. Nous le montrons à l'aide d'un exemple. Soient les SBT illustrées figure 9.7.

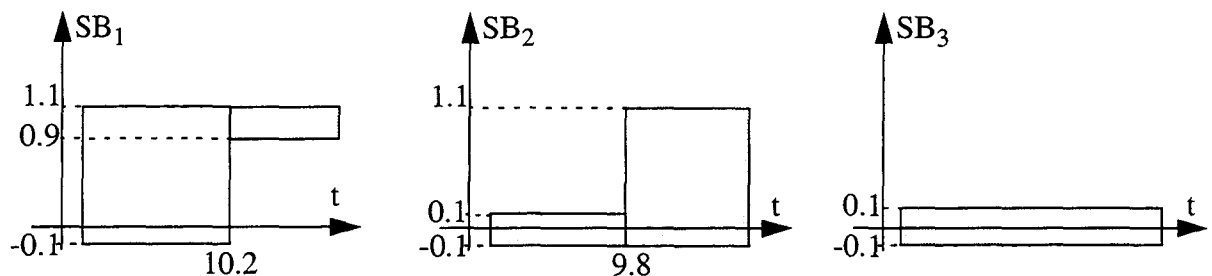


Figure 9.7 : Exemples de SBT

Ces prédictions sont cohérentes 2 à 2. En effet, les SBT $SB_1 \cap SB_2$, $\text{cons}(SB_1, SB_3)$ et $\text{cons}(SB_2, SB_3)$, illustrées respectivement figures 9.8a, 9.8b et 9.8c, ne sont pas vides.

1. Le fait que cette SBT ne soit pas vide permet d'affirmer que les autres SBT prédites ne sont pas vides. La preuve découle directement de l'inclusion de $\text{cons}(SB_1, SB_2)$ dans SB_1 et du fait que si SB_1 est incluse dans SB_1' alors $\text{cons}(SB_1, SB_2)$ est incluse dans $\text{cons}(SB_1', SB_2)$.

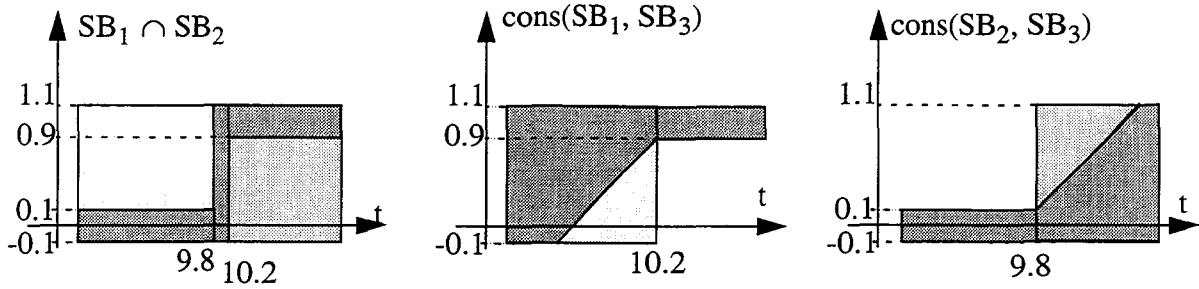


Figure 9.8 : Cohérence 2 à 2 des prédictions

Or, les 3 prédictions sont incohérentes. En effet, la SBT $\text{cons}(SB_1 \cap SB_2, SB_3)$, illustrée figure 9.9, est vide.

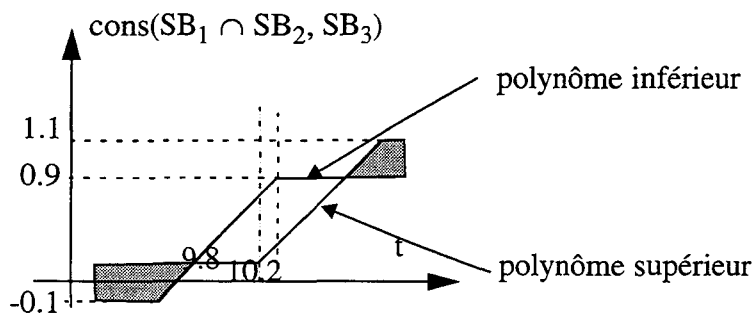


Figure 9.9 : Incohérence des 3 prédictions

Comme l'unique SBT vide est $\text{cons}(SB_1 \cap SB_2, SB_3)$, le seul ensemble de conflits détectable (et donc l'ensemble de conflits minimal) est $C_1 \cup C_2 \cup C_3$. Par conséquent, dans le cas de prédictions pour des ordres différents, l'union de 3 environnements peut être un ensemble de conflits minimal. La propriété suivante permet d'assurer que 3 est le nombre maximal.

Propriété 9.2 : "Associativité de *cons* vis à vis de l'intersection"

La fonction *cons* est associative à gauche par rapport à l'intersection.

D'une manière formelle :

$$\text{cons}(SB_1 \cap SB_2, SB_3) = \text{cons}(SB_1, SB_3) \cap \text{cons}(SB_2, SB_3)$$

La preuve découle directement du fait que le maximum des intégrales de 2 fonctions est égal à l'intégrale du maximum de ces 2 fonctions (i.e. $\max(\int_{\theta}^t f(x)dx, \int_{\theta}^t g(x)dx) = \int_{\theta}^t \max(f(x), g(x))dx$).

Cette propriété permet d'affirmer que dans le cas de la confrontation de k SBT représentant une grandeur et de la SBT représentant la dérivée de cette grandeur, **les ensembles de conflits minimaux sont l'union de 2 ou de 3 environnements associés à une prédiction.** (Ceci découle directement de propriété d'associativité de l'intersection).

La généralisation à k prédictions découle directement de cette propriété. Nous différencions deux cas :

Arrivée de la prédiction de la dérivée d'une grandeur dont la valeur est déjà prédite k fois.

Soient les prédictions cohérentes entre elles $[(g(t), i, SB_1), C_1], \dots, [(g(t), i, SB_k), C_k]$ et une nouvelle prédiction $[(g(t), i + 1, SB), C]$.

Vérifier la cohérence de cette prédiction avec celles déjà existantes consiste à :

- vérifier que $\text{cons}(SB_i, SB)$ n'est pas vide. Si ce n'est pas le cas, l'ensemble de conflits détecté est $C_i \cup C$,
- vérifier que $\text{cons}(SB_i, SB)$ et $\text{cons}(SB_j, SB)$ s'intersectent. Si ce n'est pas le cas, l'ensemble de conflits détecté est $C_i \cup C_j \cup C$.

Ces vérifications sont suffisantes car $\text{cons}(SB_1 \cap \dots \cap SB_k, SB) = \text{cons}(SB_1, SB) \cap \dots \cap \text{cons}(SB_k, SB)$ et vérifier l'intersection de k SBT revient à vérifier ces intersections 2 à 2 (propriété 9.2).

Arrivée de la prédiction d'une grandeur déjà prédite k fois et dont la dérivée est aussi prédite.

Soient les prédictions cohérentes entre elles $[(g(t), i, SB_1), C_1], \dots, [(g(t), i, SB_k), C_k]$ et $[(g(t), i + 1, SB'), C']$ et une nouvelle prédiction $[(g(t), i, SB), C]$.

Vérifier la cohérence de la nouvelle prédiction avec celles déjà existantes consiste à :

- vérifier que SB et SB_i ($i \in [1, k]$) s'intersectent. Si ce n'est pas le cas, l'ensemble de conflits détecté est $C_i \cup C$,
- vérifier que $\text{cons}(SB, SB')$ n'est pas vide. Si ce n'est pas le cas, l'ensemble de conflits détecté est $C \cup C'$,
- vérifier que $\text{cons}(SB, SB')$ et $\text{cons}(SB_i, SB)$ ($i \in [1, k]$) s'intersectent. Si ce n'est pas le cas, l'ensemble de conflits détecté est $C \cup C' \cup C_i$.

Nous nous intéressons maintenant au cas inverse.

9.3.2.4 Confrontation de la prédiction d'une grandeur et de k prédictions de la dérivée de cette grandeur

Nous présentons d'abord le cas $k = 2$. Une généralisation est ensuite proposée.

Soient les prédictions $[(g(t), i, SB), C], [(g(t), i + 1, SB_1), C_1]$ et $[(g(t), i + 1, SB_2), C_2]$.

Ces prédictions sont cohérentes si et seulement si la grandeur $g^{(i)}(t)$ peut être représentée par SB et la dérivée de cette grandeur (i.e. $g^{(i+1)}(t)$) peut être représentée par SB_1 et SB_2 .

Appliquons les règles d'intersection, on obtient :

$$[(g(t), i, SB), C], [(g(t), i + 1, SB_1), C_1] \rightarrow [(g(t), i, \text{cons}(SB, SB_1)), C \cup C_1]$$

$$[(g(t), i, SB), C], [(g(t), i + 1, SB_2), C_2] \rightarrow [(g(t), i, \text{cons}(SB, SB_2)), C \cup C_2]$$

$$[(g(t), i + 1, SB_1), C_1], [(g(t), i + 1, SB_2), C_2] \rightarrow [(g(t), i + 1, SB_{12}), C_1 \cup C_2] \text{ où } SB_{12} = SB_1 \cap SB_2$$

mais aussi

$$[(g(t), i + 1, SB_{12}), C_1 \cup C_2], [(g(t), i, SB), C] \rightarrow [(g(t), i, \text{cons}(SB, SB_{12})), C \cup C_1 \cup C_2]$$

Il découle alors directement de la règle d'incohérence que

- si $\text{cons}(\text{SB}, \text{SB}_{12}) \neq \emptyset$ alors les 3 prédictions sont cohérentes¹
- sinon
 - si $\text{SB}_{12} = \text{SB}_1 \cap \text{SB}_2 = \emptyset$ alors $C_2 \cup C_3$ est un ensemble de conflits minimal (car il provient de 2 prédictions)
 - si $\text{cons}(\text{SB}, \text{SB}_1) = \emptyset$ alors $C \cup C_1$ est un ensemble de conflits minimal.
 - si $\text{cons}(\text{SB}, \text{SB}_2) = \emptyset$ alors $C \cup C_2$ est un ensemble de conflits minimal.
 - si $\text{cons}(\text{SB}, \text{SB}_{12}) = \emptyset$ alors $C \cup C_1 \cup C_2$ est un ensemble de conflits.

Montrons avec un exemple que $C \cup C_1 \cup C_2$ peut être un ensemble de conflits minimal. Soient les SBT illustrées figure 9.10.

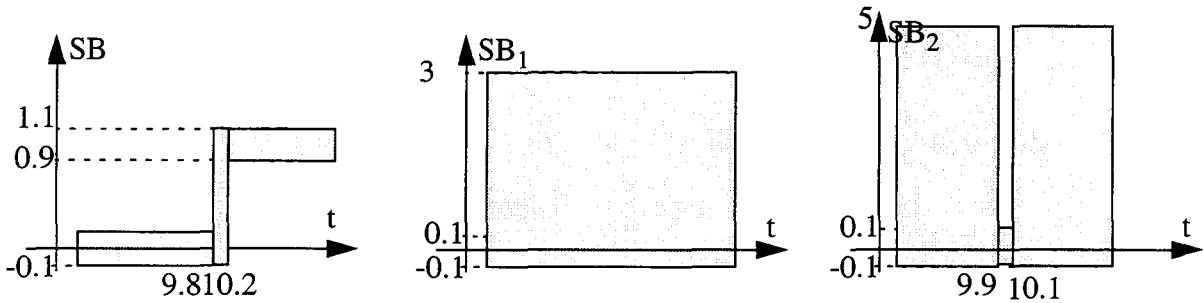


Figure 9.10 : Exemples de SBT

Montrons que ces prédictions sont cohérentes 2 à 2, c'est à dire que les SBT $\text{SB}_1 \cap \text{SB}_2$, $\text{cons}(\text{SB}, \text{SB}_1)$ et $\text{cons}(\text{SB}, \text{SB}_2)$, illustrées respectivement figures 9.11a, 9.11b et 9.11c, ne sont pas vides.

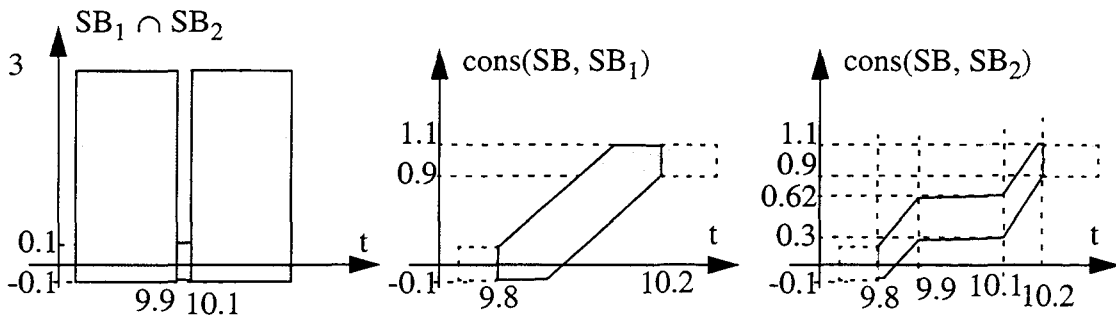


Figure 9.11 : Cohérence de deux prédictions

Montrons maintenant que ces 3 prédictions sont incohérentes (i.e. $\text{cons}(\text{SB}, \text{SB}_1 \cap \text{SB}_2) = \emptyset$). La figure 9.12 illustre cette SBT.

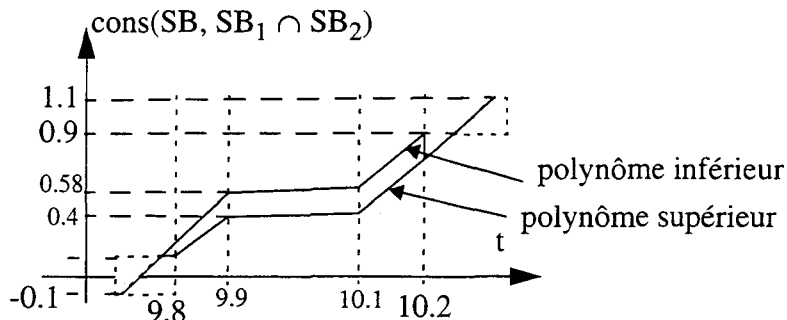


Figure 9.12 : Incohérence des 3 prédictions

1. car si SB_1 est incluse dans SB_1' alors $\text{cons}(\text{SB}_1, \text{SB}_2)$ est incluse dans $\text{cons}(\text{SB}_1', \text{SB}_2)$.

Comme l'unique SBT vide est $\text{cons}(\text{SB}, \text{SB}_1 \cap \text{SB}_2)$, le seul ensemble de conflits détectable (et donc l'ensemble de conflits minimal) est $C \cup C_1 \cup C_2$.

L'associativité à gauche de cons nous a permis d'assurer que les ensembles de conflits sont l'union d'au maximum 3 environnements dans le cas de la confrontation de plusieurs SBT représentant une grandeur et d'une SBT représentant la dérivée de cette grandeur. Malheureusement, cons n'est pas associatif à droite. Par exemple, $\text{cons}(\text{SB}, \text{SB}_1 \cap \text{SB}_2)$ n'est pas égal à l'intersection de $\text{cons}(\text{SB}, \text{SB}_1)$ et de $\text{cons}(\text{SB}, \text{SB}_2)$ (intersection qui n'est d'ailleurs pas vide). Par conséquent, dans le cas de la confrontation d'une SBT représentant une grandeur et de k SBT représentant la dérivée d'une grandeur, les ensembles de conflits minimaux peuvent être l'union d'un nombre quelconque d'environnements. La généralisation à k prédictions consiste donc à énumérer toutes les confrontations possibles :

Arrivée de la prédiction d'une grandeur dont la valeur de la dérivée est déjà prédite k fois.

Soient les prédictions cohérentes entre elles $[(g(t), i+1, \text{SB}_1), C_1], \dots, [(g(t), i+1, \text{SB}_k), C_k]$ et une nouvelle prédiction $[(g(t), i, \text{SB}), C]$.

Vérifier la cohérence de cette prédiction avec celles déjà existantes consiste à :

- vérifier que $\text{cons}(\text{SB}, \text{SB}_i)$ ($i = 1 \dots k$) n'est pas vide. Si ce n'est pas le cas, l'ensemble de conflits détecté est $C_i \cup C$,
- vérifier que $\text{cons}(\text{SB}, \text{SB}_i \cap \text{SB}_j)$ ($i, j = 1 \dots k$ et $i < j$) n'est pas vide. Si ce n'est pas le cas, l'ensemble de conflits détecté est $C_i \cup C_j \cup C$.
- ...
- vérifier que $\text{cons}(\text{SB}, \text{SB}_1 \cap \dots \cap \text{SB}_k)$ n'est pas vide. Si ce n'est pas le cas, l'ensemble de conflits détecté est $C_1 \cup \dots \cup C_k \cup C$.

Certaines confrontations peuvent être évitées en remarquant que si $\text{cons}(\text{SB}, \text{SB}_1 \cap \dots \cap \text{SB}_k)$ n'est pas vide alors aucune des SBT $\text{cons}(\text{SB}, \text{SB}_i \cap \dots \cap \text{SB}_j)$ n'est vide. Inversement, si $\text{cons}(\text{SB}, \text{SB}_i \cap \text{SB}_j)$ est vide alors toutes les SBT $\text{cons}(\text{SB}, \text{SB}_i \cap \text{SB}_j \cap \text{SB}_1)$ sont vides (toutefois, ces confrontations sont évitées en remarquant que les ensembles de conflits qu'elles permettent de détecter sont sur-ensembles d'un ensemble de conflits).

Nous présentons maintenant le troisième cas.

9.3.2.5 Confrontation d'une grandeur et de ses dérivées première et seconde

Soient les prédictions $[(g(t), i, \text{SB}_1), C_1], [(g(t), i + 1, \text{SB}_2), C_2], [(g(t), i + 2, \text{SB}_3), C_3]$

Il découle directement des règles d'intersection que :

- (1) $[(g(t), i, \text{SB}_1), C_1], [(g(t), i + 1, \text{SB}_2), C_2] \rightarrow [(g(t), i, \text{cons}(\text{SB}_1, \text{SB}_2)), C_1 \cup C_2]$
- (2) $[(g(t), i + 1, \text{SB}_2), C_2], [(g(t), i + 2, \text{SB}_3), C_3] \rightarrow [(g(t), i + 1, \text{cons}(\text{SB}_2, \text{SB}_3)), C_2 \cup C_3]$
- (3) $[(g(t), i, \text{SB}_1), C_1], [(g(t), i + 2, \text{SB}_3), C_3] \rightarrow [(g(t), i, \text{cons}(\text{SB}_1, \text{SB}_3; 2)), C_1 \cup C_3]$

mais aussi

- (4) $[(g(t), i + 1, \text{cons}(\text{SB}_2, \text{SB}_3)), C_2 \cup C_3], [(g(t), i, \text{SB}_1), C_1] \rightarrow [(g(t), i, \text{cons}(\text{SB}_1, \text{cons}(\text{SB}_2, \text{SB}_3))), C_1 \cup C_2 \cup C_3]$

Il découle directement de l'inclusion de $\text{cons}(SB_2, SB_3)$ dans SB_2 que la cohérence de la SBT obtenue avec (4) implique celle des SBT obtenues avec (1) et (2). La vérification de (3) est nécessaire car elle dépend de valeurs de $g^{(i+1)}(t)$. Toutefois, si ces valeurs sont de moins bonnes qualités que celles fournies par SB_2 alors la vérification de (3) est inutile pour la cohérence. Les déductions (1) et (2) sont nécessaires pour détecter les ensembles de conflits minimaux.

9.3.2.6 Conclusion

Etant donné le nombre de confrontations à réaliser, un algorithme général semble stupide à mettre en oeuvre ; sa complexité rendant le coût des confrontations beaucoup trop important. Toutefois, cette affirmation doit être atténuée. En fait, vérifier la cohérence de n prédictions correspondant à une même grandeur mais à des ordres différents (i.e. la détection de l'existence d'un ensemble de conflits) peut être réalisé en temps presque constant. En effet, en intersectant les prédictions correspondant aux ordres identiques, il est toujours possible de se ramener à la confrontation d'une grandeur et de ses dérivées (9.3.2.5). La cohérence de ces prédictions assure la cohérence de toutes les prédictions entre elles. En cas d'incohérence, le problème est alors de déterminer quelles sont les prédictions incohérentes. Pour limiter le coût de cette opération, plusieurs résultats ont été démontrés. D'abord, l'incohérence de n prédictions correspondant à des ordres identiques provient de l'incohérence de 2 de ces prédictions. L'incohérence de n prédictions correspondant à des ordres identiques et d'une prédiction de la dérivée provient au maximum de l'incohérence de trois de ces prédictions. Enfin, la cohérence de p prédictions engendre la cohérence de tout sous-ensemble composé de ces p prédictions. Cette relation permet la réalisation d'un algorithme dichotomique pour limiter le nombre de confrontations à réaliser.

Nous présentons maintenant quelques règles de gestion de la base de données.

9.3.3 règles de gestion de la base de données

Les règles émises dans ce paragraphe permettent de supprimer et/ou de ne pas réaliser des prédictions inutiles, c'est-à-dire une prédiction qui ne permet pas de détecter d'ensembles de conflits minimaux.

Trois règles ont été établies par de Kleer : les 3 règles de l'ATMS.

Soit une nouvelle prédiction $[(g, i, SB), C]$.

- Si, dans la base de données, il existe une prédiction $[(g, i, SB), C']$ (même SBT) alors la nouvelle prédiction n'est pas propagée. La prédiction conservée est $[(g, i, SB), C, C']$.
- Supprimer les environnements qui sont sur-ensembles d'un des environnements du label.
- Supprimer les environnements qui sont sur-ensembles d'un ensemble de conflits détecté.

Dans le cas des SBT (comme pour celui des intervalles), la règle d'égalité (première règle) a peu d'intérêt car il est rare de prédire 2 fois la même SBT. C'est pourquoi nous ne conservons pas cette règle. La conséquence principale est que toute prédiction n'a qu'un seul environnement¹. La deuxième règle est donc aussi inutile. La troisième règle est conservée.

Toutefois, plutôt que de supprimer l'environnement, il est préférable d'éliminer la prédiction.

En fait, les propriétés des opérations sur les SBT étant similaires à celles des intervalles, les règles de gestion d'une base de SBT sont similaires à celles d'une base d'intervalles. Les règles suivantes sont des extensions aux SBT de celles établies dans [Dague et al, 91].

Règle d'inclusion 1 :

Soit une nouvelle prédiction $[(g, i, SB), C]$.

Si, dans la base de données, il existe une prédiction $[(g, i, SB'), C']$ telle que $SB' \subseteq SB$ et $C' \subseteq C$ alors la nouvelle prédiction n'est ni propagée, ni enregistrée.

Règle d'inclusion 2 :

Soit une nouvelle prédiction $[(g, i, SB), C]$.

Si, dans la base de données, il existe une prédiction $[(g, i, SB'), C']$ telle que $SB \subseteq SB'$ et $C \subseteq C'$ alors cette prédiction est supprimée.

Explication :

Si C' est un ensemble de conflits alors C est aussi un ensemble de conflits. En effet, toute incohérence détectée avec SB' le sera aussi avec SB car $SB \subseteq SB'$. Par conséquent, C' est toujours sur ensemble d'un ensemble de conflits.

Nous nous intéressons maintenant aux inconvénients engendrés par l'utilisation d'un propagateur.

9.4 Limites du propagateur

Nous exposons dans ce paragraphe cinq inconvénients du propagateur.

Confrontations identiques :

Détecter les incohérences consiste à comparer les prédictions d'une même grandeur. Etant donné la symétrie des équations, les mêmes comparaisons sont réalisées pour deux grandeurs.

Par exemple, lorsque deux résistances sont en série (cf. figure 9.13), l'observation de $v_{r_1}(t)$ permet de prédire $v_{r_2}(t)$:

$$[(v_{r_1}(t), 0, SB_{v_{r_1}}), \{\}] \rightarrow [(v_{r_2}(t), 0, (R_2 / R_1) * SB_{v_{r_1}}), \{r_1, r_2, n\}]$$

Dès que l'on observe $v_{r_2}(t)$, il faut confronter cette valeur à celle prédite. De plus, par propagation de cette observation, il est possible de prédire $v_{r_1}(t)$:

$$[(v_{r_2}(t), 0, SB_{v_{r_2}}), \{\}] \rightarrow [(v_{r_1}(t), 0, (R_1 / R_2) * SB_{v_{r_2}}), \{r_1, r_2, n\}]$$



Figure 9.13 : Résistances en série

1. Le principal intérêt de l'ATMS est d'enregistrer les justifications ayant permis de réaliser une prédiction. Cet enregistrement permet de recalculer les environnements des autres prédictions lorsque cette règle est utilisée. Par conséquent, supprimer cette règle revient à éliminer le principal intérêt de l'ATMS.

Ayant 2 valeurs pour $v_{r_1}(t)$, il faut confronter ces valeurs. Or, ces confrontations sont identiques à celles de $v_{r_2}(t)$. En effet, les propriétés des opérations sur les SBT nous assurent que :

$$SB_{v_{r_2}} \cap (R_2 / R_1) * SB_{v_{r_1}} = \emptyset \Leftrightarrow SB_{v_{r_1}} \cap (R_1 / R_2) * SB_{v_{r_2}} = \emptyset$$

On remarque qu'en cas d'incohérence des prédictions, la règle d'élimination des prédictions dont l'environnement est sur-ensemble d'un ensemble de conflits permet d'éviter de refaire plusieurs fois la même confrontation. Cette règle est toutefois insuffisante car elle ne permet pas d'éviter ces confrontations inutiles lorsque les prédictions sont cohérentes.

D'une manière générale, le problème des confrontations identiques est lié à la symétrie des équations des modèles. Nous dirons que l'équation d'un modèle invoquant les grandeurs g_1 , g_2 et g_3 est symétrique lorsque cette équation permet de déduire chaque grandeur à partir des observations des 2 autres (g_1 à partir de g_2 et g_3 , g_2 à partir de g_1 et g_3 , g_3 à partir de g_1 et g_2). Lorsqu'une grandeur est prédite avec un modèle symétrique, il est inutile de confronter cette prédiction avec les autres valeurs. Toutes les confrontations étant identiques à des confrontations déjà réalisées.

Par exemple, la prédiction de $v_{r_1}(t)$ provient d'un modèle symétrique et n'est donc pas confrontée à son observation.

Les modèles non symétriques ne sont pas des occurrences rares. C'est le cas (entre autres)

- des modèles d'ordre supérieur ou égal à 1 (car la dérivée d'une grandeur n'est jamais observée),
- des modèles contenant des préconditions (les grandeurs invoquées dans la précondition ne peuvent pas être prédites à partir des grandeurs du modèle),
- des prédictions des grandeurs constantes. L'opération permettant de prendre en compte le caractère de ces constantes rend le modèle asymétrique (cf. 8.3.3.3 : imprécision introduite par les grandeurs constantes).

Il découle directement de cette règle que toutes les prédictions des grandeurs non observables ne permettent pas de détecter directement des conflits par confrontation. Elles servent uniquement à prédire d'autres grandeurs observables. Un pré-traitement des modèles permettant d'éliminer les grandeurs observables n'aurait donc aucune influence sur la détection des conflits.

Confrontations non obtenues :

Certaines confrontations ne peuvent pas être obtenues avec un propagateur étant donné l'incapacité de celui-ci à calculer certaines valeurs.

Par exemple, pour un circuit contenant deux capacités en série (cf. figure 9.14a) aucune prédiction ne peut être réalisée à partir des observations de $v_{c_1}(t)$ et $v_{c_2}(t)$. Or, ces valeurs peuvent être comparées à partir de l'équation $v_{c_1}'(t) = v_{c_2}'(t)$.

C'est aussi le cas pour le pont de composants illustré figure 9.14b. Les équations des noeuds n_1 et n_2 (i.e. $i_3(t) + i_4(t) = i_1(t)$ et $i_3(t) + i_4(t) = i_2(t)$) sont inutiles lorsque les valeurs de $i_3(t)$ et de $i_4(t)$ ne sont pas disponibles (aucune prédiction n'est réalisable). Par conséquent les valeurs des grandeurs $i_1(t)$ et $i_2(t)$ ne peuvent être comparées. Or, ces grandeurs sont liées par l'équation $i_1(t) = i_2(t)$.

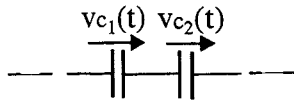


Figure 9.14a : Capacités en série

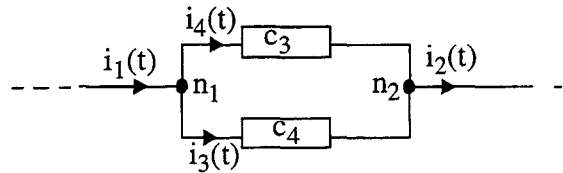


Figure 9.14b : Pont de composants

Confrontations erronées :

Un propagateur ne permet pas toujours de prédire la meilleure valeur pour une grandeur, l'imprécision introduite pouvant engendrer la non détection d'un ensemble de conflits.

Par exemple, reprenons le circuit illustré figure 9.14b. Soient SB_1 , SB_2 et SB_3 les SBT représentant $i_1(t)$, $i_2(t)$ et $i_3(t)$. Par propagation, on obtient $SB_1 - SB_3$ comme encadrement de $i_4(t)$ (par l'intermédiaire du modèle du noeud n_1), puis $SB_1 - SB_3 + SB_3$ comme encadrement de $i_2(t)$. Or, le meilleur encadrement de $i_2(t)$ est SB_1 .

Convergence lente :

La propagation d'une valeur est arrêtée lorsqu'une valeur déjà prédite est incluse dans la nouvelle valeur (i.e. règle d'inclusion 1). L'obtention d'un tel critère peut parfois prendre un temps infini car la nouvelle prédiction est toujours un peu meilleure que la précédente.

Nous illustrons ce problème avec des intervalles¹.

Soient les équations

$$(a) y = x + 1$$

$$(b) y = 2x$$

et les observations

$$x \in [0, 10] \text{ et } y \in [1, 11].$$

Par propagation, on obtient

- par (b), $x \in [0.5, 5.5]$
- par (a), $y \in [1.5, 6.5]$
- par (b), $x \in [0.75, 3.25]$
- par (a), $y \in [1.75, 4.25]$
- par (b), $x \in [0.875, 2.125]$

...

Après un certain nombre d'itérations, les encadrements de x et de y tendent vers les intervalles $[1, 1]$ et $[2, 2]$ (sans jamais les atteindre).

Pour limiter le nombre d'itérations, plusieurs possibilités sont envisageable. Par exemple, en définissant un réel ϵ tel que les intervalles $[a, b]$ et $[a + \epsilon, b - \epsilon]$ sont considérés identiques. Cette solution semble difficilement extensible aux SBT ; il faudrait définir une SBT SB_ϵ dont les fonctions sont $-\epsilon(t)$ et $\epsilon(t)$ telle que $SB + SB_\epsilon$ et SB sont considérées identiques. Une autre solution, plus facile à mettre en oeuvre, consiste à borner le nombre d'utilisations d'un modèle.

1. En remarquant que les SBT d'ordre 0 peuvent être représentées par des intervalles, il est clair que le problème de la convergence lente va aussi apparaître avec les SBT.

Propagations inutiles :

La propagation de certaines prédictions est utile uniquement lorsque des incohérences sont détectées.

Par exemple, lorsqu'il existe une prédiction $[(g(t), i, SB_1), C_1]$, la propagation d'une nouvelle prédiction $[(g(t), i, SB_2), C_2]$ telle que $SB_1 \subseteq SB_2$ et $C_1 \not\subseteq C_2$ est utile si et seulement si une incohérence est détectée avec une des prédictions obtenues lors de la propagation de $[(g(t), i, SB_1), C_1]$. En effet, la cohérence des valeurs prédites avec $[(g(t), i, SB_1), C_1]$ avec les autres valeurs assure la cohérence des valeurs prédites avec $[(g(t), i, SB_2), C_2]$. Par contre, si une incohérence est détectée alors la propagation de $[(g(t), i, SB_1), C_1]$ peut permettre de trouver un meilleur ensemble de conflits.

Pour éviter les propagations inutiles, [Dague et al, 91] propose la méthode du gel. Elle consiste à propager une seule valeur pour une grandeur et à geler les autres valeurs (i.e. les enregistrer dans une autre base)¹. Ceci permet de détecter toutes les incohérences. Pour obtenir les ensembles de conflits minimaux, il faut alors dégeler les valeurs gelées (i.e. les propager).

La règle élémentaire pour geler les prédictions est la suivante :

Soient une nouvelle prédiction de $g(t)$ $[(g(t), i, SB_1), C_1]$ et une ancienne $[(g(t), i, SB_2), C_2]$.

Une seule prédiction de $g(t)$ est conservée et propagée : $[(g(t), i, SB_1 \cap SB_2), C_1 \cup C_2]$. Les deux prédictions de départ sont gelées.

Les règles d'inclusion restent valides pour gérer les prédictions gelées.

Intuitivement, une prédiction doit être dégelée uniquement lorsque la prédiction supportant le gel a permis d'obtenir une incohérence (i.e. dégeler $[(g(t), i, SB_1), C_1]$ et $[(g(t), i, SB_2), C_2]$ uniquement si $[(g(t), i, SB_1 \cap SB_2), C_1 \cup C_2]$ ou une des prédictions obtenues par propagation a permis la détection d'une incohérence). Toutefois, il n'existe pas d'algorithme efficace pour réaliser cette opération.

L'utilisation de la technique du gel dans le cadre des SBT ne semble pas être un bon compromis car cette technique nécessite d'intersecter les SBT, opération qui augmente de manière significative la taille des SBT.

9.5 Conclusion

L'intégration des SBT dans GDE consiste à réaliser une procédure d'inférence permettant de propager les SBT et de les confronter.

La propagation nécessite 3 règles. La première (le principe fondamental) est aisée à mettre en oeuvre et consiste en un calcul de SBT. Cette règle correspond à la règle classique des propagateurs et permet de propager les valeurs des grandeurs des systèmes modélisés par des modèles d'ordre 1. La prise en compte des modèles d'ordre supérieur (pouvant provenir de combinaisons de modèles) nécessite les règles de dérivation et d'intégration. La règle de dérivation est simple à mettre en oeuvre mais son efficacité semble limitée. La mise en oeuvre de la règle d'intégration est beaucoup plus complexe car elle nécessite une procédure permettant d'obtenir la valeur d'une grandeur à partir de la dérivée de la valeur d'une autre grandeur. Nous n'avons pas réalisé cette procédure.

1. La méthode du gel a ensuite été formalisée par [Goldstone, 91].

La confrontation des SBT représentant une même grandeur (au même ordre) a une bonne propriété : tous les ensembles de conflits minimaux sont détectés en confrontant les prédictions 2 à 2. Malheureusement, cette propriété n'est pas conservée dans le cas de la confrontation des SBT représentant une même grandeur mais à des ordres différents : l'union des environnements de n prédictions peut être un ensemble de conflits minimal ! La détection de tous les ensembles de conflits minimaux nécessite donc un nombre exponentiel de confrontations. Toutefois, ceci n'est pas une catastrophe. D'abord, parce que la détection de l'existence d'un ensemble de conflits (i.e. d'une incohérence) peut être réalisée en une seule opération. Ensuite, parce que les propriétés des confrontations (notamment l'inclusion) permettent d'éviter un grand nombre de confrontations.

Montrer que deux prédictions sont incohérentes revient toujours à montrer qu'une SBT est vide (le calcul de cette SBT dépendant de l'ordre des prédictions). Ceci peut être réalisé de deux manières différentes :

- En calculant la SBT, puis en vérifiant qu'elle est vide.
- En vérifiant directement que la SBT est vide (avec les procédures de détection d'inconsistance des modèles).

La deuxième approche est toujours meilleure car son coût est moins important.

Ce raisonnement peut être généralisé : pour détecter l'inconsistance d'un système, une manière consiste à simuler ce système (i.e. prédire les valeurs des grandeurs). Une inconsistance est détectée lorsque la simulation n'est plus possible (i.e. des prédictions sont incohérentes). Une deuxième approche consiste à déterminer une procédure permettant de vérifier directement la consistance du système. C'est cette idée que nous avons développée pour créer DOGS.

Chapitre 10

DOGS

Diagnostiquer un système consiste à détecter les ensembles de contraintes inconsistants. Dans GDE, la détection d'inconsistance est réalisée par l'intermédiaire d'une étape de simulation du système (i.e. le calcul des grandeurs). L'avantage principal de cette méthode est sa "facilité" de mise en oeuvre. Le nombre important de prédictions inutiles (plus exactement, utiles pour la simulation, mais inutiles pour la détection d'inconsistance) engendre une certaine lenteur du système de diagnostic. C'est pourquoi, nous proposons un système, DOGS, permettant de détecter directement les inconsistances sans passer par l'étape de la simulation. Ce système consiste à compiler un ensemble de modèles, puis à vérifier la consistance de ces modèles un par un (avec les procédures présentées dans la troisième partie). La réalisation du compilateur est facilitée par les contraintes imposées sur les modèles : le comportement des composants est décrit par des équations différentielles linéaires par morceaux. Cette hypothèse, imposée par les opérations sur les SBT (les seules opérations primitives sont l'addition et la multiplication), n'est pas trop restrictive. Par exemple, les modèles de CATS sont des équations linéaires par morceaux alors que cette contrainte n'est pas imposée.

Cette méthode est d'abord présentée à l'aide de l'exemple du full-adder. Une démarche intuitive de compilation des modèles est ensuite proposée. Le dernier paragraphe est consacré à la caractérisation (théorique) de l'ensemble des modèles à générer.

10.1 DOGS et le Full adder

Nous sommes bien conscients que le "full adder" est un exemple jouet et que les résultats obtenus sur ce système ne garantissent pas la viabilité de notre méthode. Toutefois, l'utilisation de cet exemple est motivée par sa simplicité et la possibilité de comparer DOGS et GDE (plus exactement la procédure d'inférence de GDE).

Le comportement du full adder est décrit par 5 modèles

$\{M_1\}$	$X = A * C$	(a)
$\{M_2\}$	$Y = B * D$	(b)
$\{M_3\}$	$Z = E * C$	(c)
$\{A_1\}$	$F = X + Y$	(d)
$\{A_2\}$	$G = Y + Z$	(e)

A partir des observations $A = 3$, $B = 3$, $C = 2$, $D = 2$, $E = 3$, $G = 12$ et $F = 10$, 2 ensembles de conflits sont détectables : $\{A_1, M_1, M_2\}$ et $\{A_1, A_2, M_1, M_3\}$.

Dans GDE, ces conflits sont détectés

- en calculant $F = 12$ de deux manières différentes,
 - $F = X + Y = AC + BD$,
 - $F = X + Y = X + G - Z = AC + G - EC$,
- en détectant que ces valeurs sont incohérentes avec l'observation $F = 10$.

Ceci peut s'interpréter de la manière suivante : les comportements des composants $\{A_1, M_1, M_2\}$ et $\{A_1, A_2, M_1, M_3\}$ sont décrits par les modèles

$$\{A_1, M_1, M_2\} \quad F = AC + BD$$

$$\{A_1, A_2, M_1, M_3\} F = AC + G - EC$$

Les ensembles de conflits sont obtenus en détectant l'inconsistance de ces modèles. Par conséquent, en compilant ces modèles à partir des modèles initiaux, la détection d'inconsistance (i.e. le diagnostic) sera réalisée en vérifiant ces modèles un par un. C'est le principe de DOGS ; **à partir de la description du système (i.e. de l'ensemble de modèles SD), DOGS va générer un ensemble de modèles MSD (Model System Description) tel que toute inconsistance du système est détectée en vérifiant la consistance des modèles de MSD un par un (indépendamment de l'ordre d'arrivée des observations).**

Intuitivement, la génération de MSD dépend de la forme des modèles et de la modélisation des observations. Toutefois, **un critère est toujours valable (par conséquent, minimum) : il faut éliminer les grandeurs des équations en recombinaison les équations initiales¹.** En effet, la suppression d'une grandeur dans une équation peut permettre la détection d'un ensemble de conflits sans l'observation de cette grandeur. Par exemple, l'équation $F = AC + BD$ est obtenue en éliminant X et Y du modèle de A_1 à partir des modèles de M_1 et de M_2 . Le modèle ainsi obtenu permet la détection d'un ensemble de conflits sans les observations de X et de Y, conflit qui n'est pas détecté en vérifiant la consistance des modèles de M_1, M_2 et A_1 séparément.

En éliminant les grandeurs des équations, les modèles suivants sont obtenus pour le full adder²

$$\{M_1, M_3\} \quad X / Z = A / E \quad (f)$$

$$\{M_1, A_1\} \quad F = AC + Y \quad (g)$$

$$\{M_2, A_1\} \quad F = X + BD \quad (h)$$

$$\{M_2, A_2\} \quad G = BD + Z \quad (i)$$

$$\{M_3, A_2\} \quad G = Y + EC \quad (j)$$

$$\{A_1, A_2\} \quad F - G = X - Z \quad (k)$$

$$\{A_1, A_2, M_3\} \quad F - G = X - EC \quad (l)$$

$$\{A_1, M_1, M_2\} \quad F = AC + BD \quad (m)$$

$$\{M_1, A_1, A_2\} \quad F - G = AC - Z \quad (n)$$

$$\{M_2, M_3, A_2\} \quad G = BD + EC \quad (o)$$

$$\{A_1, A_2, M_1, M_3\} \quad F - G = AC - EC \quad (p)$$

En théorie, le nombre de modèles à générer est très important. En pratique aussi puisque 16 modèles ont été obtenus (en comptant les modèles initiaux). Pour ce problème, la méthode peut paraître absurde. Toutefois, il faut relativiser ce résultat :

- le nombre de modèles générés diminue très fortement lorsque certaines grandeurs ne peuvent pas être observées. Par exemple, en supposant que les grandeurs X, Y et Z ne sont pas observables, seuls 3 modèles sont conservés : (k), (m) et (n).
- lorsque les observations sont des réels, l'ATMS permet de ne jamais réaliser deux fois la même opération (grâce à la règle d'égalité). Dans ce cas, un grand nombre d'opérations réalisées en vérifiant la consistance des modèles un par un vont être évitées. Par contre, lorsque les observations sont des intervalles, le propagateur va en général réaliser toutes ces opérations plus un nombre important d'autres opérations.

1. Dans le cas du full adder où les observations sont des réels, on peut facilement montrer que ce critère est suffisant.

2. Nous laissons au lecteur le loisir de se convaincre de l'exhaustivité de cette liste.

Une autre information fournie par ces modèles est le nombre d'ensembles de conflits minimaux pouvant être obtenus ; ce nombre est nettement inférieur au nombre total des ensembles de conflits minimaux (16 au lieu de $2^5 = 32$).

DOGS repose sur l'existence d'un compilateur de modèles. Nous en proposons un dont le principe est d'éliminer les grandeurs des équations des modèles ; celles-ci sont supposées linéaires par morceaux.

10.2 DOGS : le compilateur

Nous présentons d'abord le compilateur pour les modèles d'ordre 0. Celui-ci sera ensuite étendu aux modèles d'ordre quelconque.

10.2.1 Modèles d'ordre 0

Un modèle d'ordre 0 a été défini comme un triplet $(C, T, (f(t)))$. Nous supposons que l'équation d'un modèle (i.e. $f(t) = 0$) est une équation linéaire¹ de la forme $\sum a_i g_i(t) = 0$ où

- les a_i sont des grandeurs élémentaires constantes ou des réels,
- les $g_i(t)$ sont des grandeurs élémentaires variant dans le temps.

Les grandeurs élémentaires sont séparées en deux catégories :

- les grandeurs observables, c'est-à-dire les grandeurs qui pourront être observées pendant la procédure de diagnostic,
- les grandeurs non observables.

Typiquement, en électronique analogique, les grandeurs observables sont les tensions aux bornes des composants et les grandeurs non observables sont les courants.

Nous noterons $\{g_1(t), \dots, g_n(t)\}$ les grandeurs élémentaires. Les k premières seront supposées non observables.

Le comportement d'un système est décrit par un ensemble de modèles. Soit $SD = \{(C_1, T_1, (f_1(t))), \dots, (C_p, T_p, (f_p(t)))\}$ cet ensemble. Les équations de ces modèles correspondent à un système d'équations linéaires :

$$\begin{array}{cc}
 C_1 & T_1 \\
 & \dots \\
 C_p & T_p
 \end{array}
 \begin{array}{c}
 \left[\begin{array}{ccc|ccc}
 a_{11} & \dots & a_{1k} & a_{1k+1} & \dots & a_{1n} \\
 \dots & & & & & \\
 \dots & & & & & \\
 a_{p1} & \dots & a_{pk} & a_{pk+1} & \dots & a_{pn}
 \end{array} \right]
 \begin{array}{c}
 \left[\begin{array}{c}
 g_1(t) \\
 \dots \\
 g_n(t)
 \end{array} \right] = 0
 \end{array}
 \end{array}
 \quad (1)$$

On dira qu'une grandeur $g_i(t)$ est invoquée dans un modèle d'équation $f_j(t)$ lorsque le coefficient a_{ji} est différent de 0.

Notre but est de diagnostiquer les systèmes en vérifiant la consistance des modèles un par un.

1. En associant l'extension temporelle à cette hypothèse, le comportement d'un ensemble de composants est donc décrit par une équation linéaire par morceaux.

Or, vérifier la consistance d'un modèle nécessite les observations de toutes les grandeurs du modèle. Nous allons combiner les modèles afin de supprimer les grandeurs non observables. La combinaison des modèles est réalisée avec la règle suivante.

Propriété 10.1 : "Combinaison de modèles"

La combinaison de deux modèles $(C_1, T_1, (f_1(t)))$ et $(C_2, T_2, (f_2(t)))$ est un modèle $(C_1 \cup C_2, T_1 \cap T_2, (a f_1(t) + b f_2(t)))$ où a et b sont deux réels quelconques.

D'une manière formelle :

$$(C_1, T_1, (f_1(t))) \wedge (C_2, T_2, (f_2(t))) \Rightarrow (C_1 \cup C_2, T_1 \cap T_2, (a f_1(t) + b f_2(t)))$$

Si un des coefficients a ou b est égal à 0 alors le modèle obtenu est inutile ; le seul effet de la combinaison étant d'augmenter la taille de l'ensemble de composants et de diminuer celle de l'extension temporelle.

Deux types de grandeurs ont été distingués. L'élimination de ces grandeurs est réalisée avec deux méthodes différentes.

Elimination des grandeurs non observables :

Eliminer une grandeur non observable consiste à combiner deux modèles (en choisissant adroitement les réels a et b). En étendant ce principe, supprimer les k grandeurs non observables consiste à combiner k + 1 modèles¹. L'environnement (i.e. l'ensemble de composants) du modèle obtenu est l'union des k + 1 environnements des modèles initiaux. L'élimination des grandeurs non observables est réalisée par élimination gaussienne. Cette méthode consiste à supprimer une grandeur en combinant un modèle avec tous les autres, puis éliminer une deuxième grandeur en combinant un autre modèle (ne contenant plus la première grandeur) avec les modèles restants, etc... (cf figure 10.1)

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots \\ a_{p1} & a_{p2} & \dots & a_{pn} \end{bmatrix}$$

Figure 10.1a : Etape 0

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & b_{22} & \dots & b_{2n} \\ 0 & b_{32} & b_{33} & \dots & b_{3n} \\ 0 & \dots & \dots & \dots & \dots \\ 0 & b_{p2} & c_{p3} & \dots & b_{pn} \end{bmatrix}$$

Figure 10.1b : Etape 1

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & b_{21} & \dots & b_{2n} \\ 0 & 0 & c_{33} & \dots & c_{3n} \\ 0 & 0 & \dots & \dots & \dots \\ 0 & 0 & c_{p3} & \dots & c_{pn} \end{bmatrix}$$

Figure 10.1c : Etape 2

Les modèles issus des combinaisons de k+1 modèles sont tous utiles car ils ont des environnements différents ; ils permettent donc la détection d'ensembles de conflits différents.

Nous nous intéressons maintenant à l'élimination des grandeurs observables.

Elimination des grandeurs observables :

Le principe est identique à celui de l'élimination des grandeurs non observables. La combinaison de i modèles permet d'éliminer i - 1 grandeurs. Toutefois, l'utilisation de cette règle est différente.

- Les modèles issus de recombinaisons autorisent la détection d'ensembles de conflits avec un nombre plus réduit d'observations. C'est pourquoi ils sont intéressants.

1. Cette extension suppose qu'une seule grandeur non observable est éliminée à chaque combinaison. Afin de faciliter l'explication de cette démarche, cette hypothèse simplificatrice sera supposée.

- Les grandeurs éliminées correspondent aux grandeurs qui n'auront pas besoin d'être observées. En réalisant les combinaisons de i modèles de manière différente, les $i - 1$ grandeurs éliminées sont différentes. Il faut donc réaliser les combinaisons de i modèles de toutes les manières possibles (i.e. considérer tous les pivots de Gauss).
- Combiner les modèles augmente la taille des environnements associés aux modèles obtenus et donc des ensembles de conflits détectables. Il faut donc réaliser toutes les combinaisons de i modèles parmi les p modèles obtenus en fin de première étape. Chaque combinaison permettant l'élimination d'une grandeur mais augmentant la taille de l'environnement associé au modèle.

En résumé, les modèles générés à partir de k modèles contenant uniquement des grandeurs observables sont ceux obtenus par combinaison de ces modèles 2 par 2, 3 par 3, ... où la combinaison de p modèles consiste à éliminer $p - 1$ grandeurs de toutes les manières possibles.

Par exemple, pour 3 modèles contenant 4 grandeurs observables, les combinaisons des lignes 1

et 2, 2 et 3, 1 et 3 donnent des matrices de la forme

$$\begin{bmatrix} 0 & a_{12} & a_{13} & a_{14} \\ a_{21} & 0 & a_{23} & a_{24} \\ a_{31} & a_{32} & 0 & a_{34} \\ a_{41} & a_{42} & a_{43} & 0 \end{bmatrix}, \text{ et les}$$

combinaisons des lignes 1, 2 et 3 donnent une matrice de la forme

$$\begin{bmatrix} 0 & 0 & b_{13} & b_{14} \\ 0 & b_{22} & 0 & b_{24} \\ 0 & b_{32} & b_{33} & 0 \\ b_{41} & 0 & 0 & b_{44} \\ b_{51} & 0 & b_{53} & 0 \\ b_{61} & b_{62} & 0 & 0 \end{bmatrix}$$

Nous proposons un algorithme permettant de générer les modèles.

MAT est la matrice initiale

ALNP est la liste des lignes qui n'ont pas servi comme pivot (toutes les lignes au départ)

ACP est la liste des colonnes qui ont servi comme pivot (l'ensemble vide au départ)

Propager(ALNP, ACP, MAT)

Tant que ALNP $\neq \emptyset$ faire

si il existe une ligne *lig* de ALNP contenant une grandeur non observable *col* alors

supprimer *lig* de ALNP

substituer(*lig*, *col*, MAT, ATUCH, MATSUB)

enregistrer(ATUCH, MATSUB)

propager(ATUCH, ACP, MATSUB)

sinon

prendre et supprimer *lig* de ALNP

Pour toutes les grandeurs observables *col* de *lig* n'appartenant pas à ACP faire

substituer(*lig*, *col*, MAT, ATUCH, ASUB)

enregistrer(ATUCH, MATSUB)

NACP = ACP \cup {*col*}

propager(ATUCH, NACP, ASUB)

fin pour

fsi

simplifier(ALNP, ACP, MAT)

ftq

avec :

substituer(lig, col, MAT, ATUCH, MATSUB) est une procédure supprimant la grandeur col de toutes les lignes de MAT. ATUCH est la liste des lignes substituées et MATSUB est la matrice obtenu après substitution.

enregistrer(LISTE, MAT) est une procédure enregistrant les lignes de LISTE lorsque celles ci ne contiennent aucune grandeur observable.

simplifier(ALNP, ACP, MAT) est une procédure qui

- supprime les modèles contenant une grandeur non observable appartenant à aucun des autres modèles.
- substitue et supprime 2 modèles contenant une grandeur non observable appartenant à aucun autre modèle.
- enregistre les grandeurs observables appartenant à un unique modèle comme des grandeurs propagées.

Nous présentons maintenant un exemple d'application de cet algorithme.

Soit le circuit illustré figure 10.2.

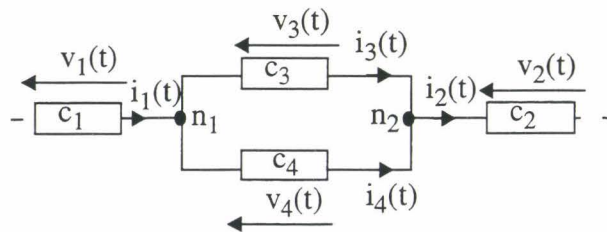


Figure 10.2 : Pont de composants

Nous supposons que les grandeurs observables sont les tensions (i.e. $v_1 \dots v_4$).

Les modèles de ce circuit permettent de créer la matrice suivante :

No	label	i_1	i_2	i_3	i_4	v_1	v_2	v_3	v_4
(1)	$\{r_1\}$	r_1				-1			
(2)	$\{r_2\}$		r_2				-1		
(3)	$\{r_3\}$			r_3				-1	
(4)	$\{r_4\}$				r_4				-1
(5)	$\{n_1\}$	1		-1	-1				
(6)	$\{n_2\}$		1	-1	-1				

Après simplification de cette matrice avec les règles ci dessus, on obtient :¹

No	label	i_3	i_4	v_1	v_2	v_3	v_4
(3)	{ r_3 }	r_3				-1	
(4)	{ r_4 }		r_4				-1
(5)	{ n_1, r_1 }	-1	-1	$1 / r_1$			
(6)	{ n_2, r_2 }	-1	-1		$1 / r_2$		

Aucune grandeur observable ne sera propagée (i.e. $ACP = \{v_1, \dots, v_4\}$)

Seules les principales étapes de l'algorithme sont détaillées :

Propager({3, 4, 5, 6})

choix : lig = 3, col = i_3

lignes touchées : {5, 6}

Propager({5, 6})

Choix : lig = 5, col = i_4

lignes touchées {4, 6}

Ces lignes sont enregistrées. Les modèles obtenus sont

(a) $\{n_1, r_1, r_3, n_2, r_2\} \quad v_1 / r_1 - v_2 / r_2 = 0$

(b) $\{n_1, r_1, r_3, r_4\} \quad -v_1 / r_1 + v_3 / r_3 + v_4 / r_4 = 0.$

La matrice suivante est alors obtenue.

No	label	i_3	i_4	v_1	v_2	v_3	v_4
(3)	{ r_3 }	r_3				-1	
(4)	{ n_1, r_1, r_3, r_4 }		0	$-1 / r_1$		$1 / r_3$	$1 / r_4$
(5)	{ n_1, r_1, r_3 }	0	-1	$1 / r_1$		$-1 / r_3$	
(6)	{ n_1, r_1, r_3, n_2, r_2 }	0	0	$1 / r_1$	$-1 / r_2$		

(Suite de l'algorithme)

Propager({4, 6})

Ces deux lignes ne peuvent pas être recombinaées car elles n'invoquent plus de grandeurs non observables. L'algorithme revient donc à la deuxième boucle de

Propager({5, 6}).

Choix : lig = 6, col = i_4

lignes touchées {4} (la ligne 5 n'est pas substituée car elle a été propagée).

Cette ligne est enregistrée, le modèle est :

(c) $\{n_2, r_2, r_3, r_4\} \quad -v_2 / r_2 + v_3 / r_3 + v_4 / r_4 = 0.$

1. La simplification de la matrice n'introduit aucune imprécision. Par conséquent, cette opération peut aussi être réalisée dans le cadre d'un propagateur et permet de diminuer de manière significative le nombre de modèles initiaux et le nombre de grandeurs.

La matrice suivante est alors obtenue.

No	label	i_3	i_4	v_1	v_2	v_3	v_4
(3)	{ r_3 }	r_3				-1	
(4)	{ n_2, r_2, r_3, r_4 }		0		$-1/r_2$	$1/r_3$	$1/r_4$
(6)	{ n_2, r_2, r_3 }	0	-1		$1/r_2$	$-1/r_3$	

(Suite de l'algorithme)

Propager({4})

Cette ligne n'est pas propagée. L'algorithme revient à la deuxième boucle de

Propager({3, 4, 5, 6})

choix : lig = 4, col = i_4 (deuxième ligne de **propager**({3, 4, 5, 6}))

La suppression de la ligne 3 engendre une simplification de la matrice.

No	label	i_3	i_4	v_1	v_2	v_3	v_4
(5)	{ n_1, r_1, n_2, r_2 }	0	0	$-1/r_1$	$1/r_2$		

La ligne (5) est enregistrée. On obtient le modèle

(d) { n_1, r_1, n_2, r_2 } $-v_1/r_1 + v_2/r_2 = 0$

On remarque que l'on obtient deux fois l'équation $v_1/r_1 - v_2/r_2 = 0$ (les modèles (a) et (d)). Ce modèle est obtenu directement par combinaison des lignes 5 et 6. Or, on peut aussi l'obtenir en recombinaison la ligne 3 avec les lignes 5 et 6, puis en recombinaison les lignes 5 et 6. Le label obtenu n'est alors pas minimal.

Il semble que l'algorithme de compilation des modèles proposé soit complet¹ (ceci n'est pas prouvé). Le problème de cet algorithme est qu'il fournit un nombre trop important de modèles (notamment lorsque deux grandeurs sont éliminées en même temps). Ce problème peut être évité en notant les justifications des combinaisons, c'est à dire les modèles impliqués dans la combinaison. La génération de modèles inutiles est évitée en n'enregistrant pas et en ne propageant pas les modèles lorsqu'une justification a été supprimée. Ceci permet par exemple de ne pas enregistrer le modèle (a).

Nous étendons maintenant cet algorithme aux modèles d'ordre supérieur.

10.2.2 Extension aux modèles d'ordre supérieur

Pour noter les modèles d'ordre supérieur dans la matrice, nous proposons de noter $d(a, i)$ le coefficient à associer à la dérivée ième d'une grandeur. Par exemple, le modèle d'une capacité

1. La complétude étant donné vis-à-vis de l'objectif fixé, c'est-à-dire d'éliminer les grandeurs des équations de toutes les façons possibles.

(i.e. $i_c(t) = c v_c'(t)$) est noté

$i_c(t)$	$v_c(t)$
-1	$d(c, 1)$

Cette notation ne permet pas d'écrire les équations invoquant une grandeur et sa dérivée. Toutefois, ce type d'équation est une occurrence rare.

Nous introduisons maintenant l'opération de dérivation d'un modèle : dériver un modèle consiste à incrémenter l'ordre des coefficients, c'est à dire à remplacer le coefficient a par $d(a, 1)$ et $d(a, i)$ en $d(a, i + 1)$. Par exemple, la dérivée de l'équation de la capacité est

$i_c(t)$	$v_c(t)$
$d(-1, 1)$	$d(c, 2)$

Etendre l'algorithme de combinaison des modèles d'ordre 0 consiste à étendre la procédure substituer(*lig, col, MAT, ATUCH, MATSUB*). Combiner deux lignes consiste à dériver les lignes de manière à obtenir des ordres identiques pour les coefficients du pivot (i.e. de la grandeur supprimée), puis à substituer les lignes.

Par exemple, la suppression de la grandeur $g1$ des lignes (1) et (2) de la matrice ci dessous consiste à dériver la ligne (2) (dont le résultat est à la ligne 3) et à substituer les lignes 1 et 3 (résultat à la ligne 4).

No	label	$g1$	$g2$	$g3$
(1)	{ h_1 }	$d(a_1, 1)$	$d(a_2, 1)$	a_3
(2)	{ h_2 }	b_1	b_2	0
(3)	{ h_1 }	$d(b_1, 1)$	$d(b_2, 1)$	0
(4)	{ h_1, h_2 }	0	$d(a_1 b_2 - a_2 b_1, 1)$	$- b_1 a_3$

Etant donné le compilateur proposé, nous proposons de présenter les avantages de la compilation vis-à-vis de la propagation.

10.3 Avantages et inconvénients d'un compilateur

Dans le chapitre précédent, quelques inconvénients des propageurs ont été présentés. Nous allons montrer que la compilation apporte une solution à certains de ces inconvénients, notamment au nombre de confrontations obtenues.

Il est clair que le nombre de confrontations obtenues dépend de notre capacité à générer des modèles. Le modèle obtenu en éliminant les grandeurs de p modèles correspond à la confrontation de l'observation et d'une prédiction d'une grandeur. La prédiction est obtenue en propageant l'observation d'une autre grandeur à travers les p modèles. Par exemple, pour un circuit comprenant deux résistances en série (cf. figure 10.3), l'observation de $v_{r_1}(t)$ (i.e. la SBT SB_{r_1}) permet de prédire une valeur pour $v_{r_2}(t)$: $R_2 / R_1 SB_{r_1}$. Confronter cette prédiction avec l'observation de $v_{r_2}(t)$ revient à vérifier la consistance du modèle $v_{r_1}(t) / r_1 - v_{r_2}(t) / r_2 = 0$

pour la SBT $(1/R_1) * SB_{r_1} - (1/R_2) * SB_{r_2}$.

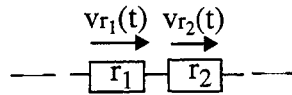


Figure 10.3 : Résistances en série

La consistance de chaque modèle n'étant vérifiée qu'une seule fois, le problème des confrontations identiques est naturellement résolu. Dans le cas des résistances en série, les confrontations des observations de $v_{r_1}(t)$ et de $v_{r_2}(t)$ avec les prédictions $R_1 / R_2 SB_{r_2}$ et $R_2 / R_1 SB_{r_1}$ sont identiques. Ces confrontations sont réalisées une seule fois en vérifiant la consistance du modèle $v_{r_1}(t) / r_1 - v_{r_2}(t) / r_2 = 0$.

La propagation d'une grandeur nécessite de connaître les valeurs de toutes les grandeurs du modèle. Cette particularité aboutit parfois à la non détection de certaines confrontations. Par contre, les modèles correspondant à ces confrontations seront générés car la compilation est réalisée indépendamment de la connaissance des grandeurs. Deux cas de non détection d'une confrontation ont été isolés :

- apparition d'une expression de grandeurs inconnues dans plusieurs modèles

Lorsque plusieurs grandeurs non observables apparaissent dans l'expression d'un modèle, alors celui ci est inutile dans le cadre de la propagation. En effet, aucune valeur ne pourra être prédite avec ce modèle. Par contre, une simple substitution de ces grandeurs, telle que celle réalisée par un compilateur, peut permettre la génération d'un modèle. Par exemple, les courants $i_3(t)$ et $i_4(t)$ (grandeurs non observables) apparaissent dans les modèles des noeuds n_1 et n_2 (i.e. $i_1(t) = i_3(t) + i_4(t)$ et $i_2(t) = i_3(t) + i_4(t)$) du circuit illustré figure 10.4. Une simple substitution de ces modèles permet d'obtenir la relation $i_1(t) = i_2(t)$. Cette relation ne peut être obtenue avec un propagateur.

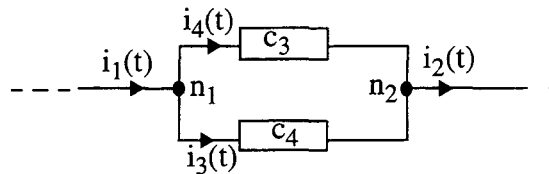


Figure 10.4 : Pont de composants

- ambiguïté entre la propagation et la confrontation des dérivées des grandeurs

La propagation de la dérivée d'une grandeur nécessite la connaissance de cette dérivée. Par contre, la confrontation de la dérivée d'une grandeur nécessite uniquement la valeur de cette grandeur. Cette ambiguïté peut aboutir à la non obtention de certaines confrontations.

Par exemple, les modèles décrivant le comportement de deux capacités en série (cf. figure 10.5) sont $i_{c_1}(t) = c_1 v_{c_1}'(t)$, $i_{c_2}(t) = i_{c_1}(t)$ et $i_{c_2}(t) = c_2 v_{c_2}'(t)$. En éliminant (par substitution) les grandeurs $i_{c_1}(t)$ et $i_{c_2}(t)$ de ces équations, on obtient le modèle $c_1 v_{c_1}'(t) = c_2 v_{c_2}'(t)$. L'obtention de cette confrontation avec un propagateur nécessite la connaissance des dérivées des tensions, alors que la confrontation peut être réalisée à partir des valeurs des tensions.

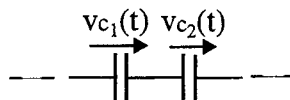


Figure 10.5 : Capacités en série

Ce problème se pose d'une manière plus générale pour obtenir les confrontations dans le cadre des systèmes décrits par des modèles d'ordre supérieur ou égal à 2. La règle d'intégration permet de surmonter certains problèmes, mais pas celui des deux capacités en série. De plus, elle semble difficile à mettre en oeuvre.

Exhaustivité des modèles générés

Vérifier la consistance d'un modèle revient à confronter deux valeurs d'une grandeur. Ceci est suffisant dans le cadre des modèles d'ordre 0, mais malheureusement pas dans celui des modèles d'ordre supérieur. Par conséquent, vérifier la consistance des modèles obtenus en éliminant les grandeurs n'est pas suffisant pour détecter tous les ensembles de conflits. Par exemple, pour un circuit composé de deux résistances et d'une capacité en série (cf. figure 10.6), le compilateur va générer 3 modèles (seules les tensions sont observables) :

$$v_{r_1}(t) / r_1 - v_{r_2}(t) / r_2 = 0, v_{r_1}(t) - r_1 c v_c'(t) = 0 \text{ et } v_{r_2}(t) - r_2 c v_c'(t) = 0$$

Vérifier la consistance des deux derniers modèles revient à confronter l'observation de $v_c(t)$ avec les prédictions de $v_c'(t)$ obtenues à partir des observations de $v_{r_1}(t)$ et de $v_{r_2}(t)$ (i.e. les SBT représentant $v_{r_1}(t) / r_1 c$ et $v_{r_2}(t) / r_2 c$). Or, une inconsistance peut être uniquement détectable en confrontant l'observation de $v_c(t)$ et l'intersection des prédictions. Le modèle correspondant à cette confrontation n'étant pas généré, cette confrontation n'est pas réalisée par DOGS.

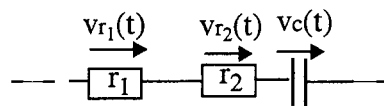


Figure 10.6 : Insuffisance des modèles générés

Confrontations erronées :

Ce problème est provoqué par la perte du lien existant entre les variables réelles dès lors que la substitution par les observations (des intervalles ou des SBT) est réalisée. Ce problème n'a pas de raison d'être pour un compilateur puisque cette substitution n'est pas réalisée.

Convergence lente :

Ce problème est typiquement lié aux propagateurs d'intervalles (et au propagateur de SBT par conséquent). En analysant le problème, on remarque qu'une convergence lente arrive lorsque le système de contraintes formé par les équations et les observations est consistant. La convergence lente n'est donc pas liée à la vérification de la consistance du système mais plutôt au calcul des grandeurs. Or, dans le cadre du diagnostic, seule la consistance est une information utile. Ce problème est évité avec la compilation car seule la consistance du système est vérifiée. Ceci nous fait dire qu'un compilateur ne se résume pas à un propagateur formel.

Par exemple, le système de contraintes suivant

- (a) $y = x + 1$
- (b) $y = 2x$
- $x \in [0, 10]$ et $y \in [1, 11]$.

aboutit à une convergence lente (cf. chapitre précédent).

Le compilateur permet de générer deux modèles : $x = 1$ et $y = 2$. Vérifier la consistance des quatre modèles (les deux générés et les deux initiaux) est immédiat et suffisant¹.

Propagations inutiles :

Etant données les observations, certaines prédictions ne sont pas propagées. Par conséquent, certaines confrontations ne sont pas réalisées. Par exemple, de Kleer propose de ne pas propager une nouvelle prédiction lorsque celle ci a déjà été réalisée. Seuls les environnements des prédictions déjà existantes sont recalculés. Ainsi, dans le cas du full adder, les observations $A = 3, B = 3, C = 2, D = 2, E = 3$ permettent de prédire $[F = 12, \{A_1, M_1, M_2\}]$ et $[G = 12, \{A_2, M_2, M_3\}]$. L'observation de $G = 12$ n'est pas propagée car cette valeur existe déjà. Par contre, le label de la prédiction $F = 12$ est recalculée : $[F = 12, \{A_1, M_1, M_2\}, \{A_1, A_2, M_1, M_3\}]$. L'observation de $F = 10$ permet alors de détecter deux ensembles de conflits.

Vérifier la consistance d'un modèle revient à confronter deux prédictions. La non propagation d'une prédiction revient donc à ne pas vérifier la consistance de certains modèles car, étant données les observations, les consistances de ces modèles peuvent être déduites de la consistance des autres modèles. La génération des modèles correspondant aux différentes confrontations non réalisées est donc inutile. Or, comme la compilation est faite indépendamment des observations, les modèles correspondant aux différentes confrontations sont tous générés. Par exemple, lorsque X, Y et Z ne sont pas observables, 3 modèles sont générés :

$$\begin{array}{lll} \{A_1, M_1, M_2\} & F = AC + BD & (k) \\ \{M_2, M_3, A_2\} & G = BD + EC & (m) \\ \{A_1, A_2, M_1, M_3\} & F - G = AC - EC & (n) \end{array}$$

L'observation $G = 12$ permet d'assurer la consistance de (m). Le recalcul des environnements revient à remarquer que les modèles (k) et (n) sont tous les deux inconsistants ou tous les deux consistants. La génération d'un de ces modèles est donc inutile. D'une manière générale, deux relations entre la consistance de ces modèles peuvent être établies :

- la consistance de 2 de ces 3 modèles engendre la consistance du troisième,
- la consistance d'un des modèles et l'inconsistance d'un autre engendre l'inconsistance du troisième. Il découle directement de cette relation que l'existence d'un unique ensemble de conflits est impossible. Il existe nécessairement 0, 2 ou 3 ensembles de conflits.

En conclusion, ne pas propager une prédiction déjà réalisée permet de ne pas générer certains modèles et donc de ne pas vérifier leur consistance. Le recalcul des environnements des prédictions permet de déduire l'inconsistance des modèles non générés. Toutefois, le coût de cette opération n'est que légèrement inférieur à celui de la génération des modèles dont la consistance peut être déduite. Le véritable gain n'est donc pas la non génération de ces modèles, mais plutôt la déduction de leur inconsistance. En d'autres termes, le problème de DOGS n'est pas de générer trop de modèles, il est plutôt de déduire automatiquement l'inconsistance des modèles lorsque c'est possible. Etant donné que ceci est réalisé par un propageur, on peut penser que de tels systèmes seront nécessairement plus rapides que DOGS. Ceci n'est pas nécessairement le cas. D'abord parce qu'un propageur réalise beaucoup d'opérations inutiles qui ne sont pas faites par un compilateur. Par exemple, les environnements des prédictions des grandeurs A, B, C, D et E sont aussi recalculés. Ces opérations n'ont pas été décrites car elles sont inutiles. Toutefois, elles contribuent pour une large part à la lenteur des propageurs. Ensuite, parce que l'inefficacité de DOGS dépend du nombre de modèles dont la consistance peut être déduite. Or, ce nombre dépend de la

1. Pour cet exemple, il est clair que la vérification de la consistance des deux modèles générés est suffisante. Dans le cadre du diagnostic, il faut aussi vérifier la consistance des deux modèles initiaux car cela peut permettre la détection d'ensembles de conflits plus petits.

modélisation des observations. Il est clair qu'il est important lorsque les observations sont des réels. Par contre, il semble que ce ne soit plus le cas lorsque les observations sont des intervalles ou des SBT. Une explication intuitive est immédiate. Dans le cas des observations réelles, la vérification de la consistance d'un modèle est inutile lorsque deux prédictions d'une grandeur sont identiques. Or, ce cas est rare lorsque les observations sont des intervalles.

Nous proposons maintenant une caractérisation théorique de l'ensemble des modèles à générer. Quelques propositions permettant d'établir les liens entre les modèles sont ensuite conjecturées.

10.4 Approche théorique

Nous proposons d'abord une définition de l'ensemble des modèles à générer en terme de consistance. Quelques voies de recherche sont alors proposées pour définir les modèles appartenant à cet ensemble lorsque les modèles initiaux sont d'ordre 0.

Dans le formalisme de Reiter, la description du système est un ensemble de formules du premier ordre, noté SD. Chaque formule correspond à un modèle de composant(s). Nous noterons MSD (Model System Description) l'ensemble des formules à générer pour trouver les ensembles de conflits minimaux en vérifiant la consistance des modèles un par un (et indépendamment de l'ordre d'arrivée des observations). Intuitivement, MSD est un ensemble de modèles tel que pour tout ensemble d'observations, tout conflit détecté avec SD doit être détecté avec MSD. D'une manière formelle, ceci peut s'écrire :

$$\begin{aligned} & \text{MSD est un ensemble de formules du premier ordre tel que} \\ & \forall \text{OBS} \subseteq \text{OBS}_{\text{tot}}, \forall \text{C} \subseteq \text{COMP}, \\ & \text{SD} \cup \text{OBS} \cup \{\neg \text{AB}(c) \mid c \in \text{C}\} \text{ inconsistant} \\ & \Leftrightarrow \exists \text{M} \in \text{MSD} \text{ tel que } \{\text{M}\} \cup \text{OBS} \cup \{\neg \text{AB}(c) \mid c \in \text{C}\} \text{ inconsistant} \end{aligned}$$

où OBS_{tot} est l'ensemble de toutes les observations possibles de toutes les grandeurs du système¹ et COMP est l'ensemble des composants.

Cette définition a un intérêt tout relatif puisqu'elle permet uniquement de caractériser MSD en terme de consistance. Or, le problème est de caractériser les modèles de cet ensemble. Le développement que nous proposons est basé sur une approche intuitive du problème.

Condition nécessaire :

MSD doit contenir toutes les combinaisons des modèles initiaux permettant l'élimination de grandeurs. Intuitivement, cette proposition est logique car pris séparément, chaque modèle de MSD a un intérêt ; soit il permet la détection d'un ensemble de conflits avec un nombre restreint d'observations, soit il permet la détection d'un ensemble de conflits plus petit. Cette condition nécessaire est d'ailleurs la base de notre algorithme.

Condition suffisante :

L'existence d'une condition suffisante, c'est à dire l'existence de MSD, dépend des modélisations des observations et du comportement des composants. Rien n'assure qu'une

1. Par exemple, si l'observation d'une grandeur est un réel alors l'ensemble des observations possibles de cette grandeur est l'ensemble des réels.

telle condition existe toujours. Toutefois, il semble que l'existence d'une condition suffisante dépende surtout de la manière dont sont compilés les modèles. Par exemple, les modèles obtenus en éliminant les grandeurs correspondent aux confrontations 2 à 2 des prédictions réalisées par un propagateur. Or, la vérification de la cohérence d'une SBT représentant une grandeur et de deux SBT représentant la dérivée de cette grandeur nécessite de confronter les prédictions 3 par 3. L'existence de MSD dépend alors de notre capacité à générer des modèles correspondant à la confrontation de 3 prédictions.

Un aspect intéressant du problème serait de définir les cas où la condition nécessaire est aussi suffisante. Nous pensons que c'est le cas lorsque les observations sont des SBT continues et que les équations des modèles sont linéaires (i.e. sont de la forme $\sum a_i g_i(t)$ où les a_i sont des réels). Nous exprimons ceci avec la propriété suivante dont la preuve n'a malheureusement pas pu être établie.

Un système décrit par p équations linéaires est consistant si et seulement si les modèles obtenus en éliminant au moins $p - 1$ grandeurs sont tous consistants.

Cette proposition est logique. La valeur des grandeurs n'étant pas connue, la présence d'une grandeur dans une équation correspond à l'introduction d'une imprécision. En minimisant le nombre de grandeurs dans les équations, on minimise aussi l'imprécision introduite.

Comme l'addition de deux SBT et la multiplication d'une SBT par un intervalle dégénéré n'introduisent pas d'imprécision, il suffit de montrer cette propriété pour des grandeurs encadrées par des intervalles. D'une manière formelle, cette proposition peut alors s'écrire :

Soient les équations $f_j(x_1, \dots, x_n) = \sum_{i=1}^n a_{ji} x_i = 0$ ($j = 1 \dots p$) où les coefficients a_{ji} sont des réels.

Soient $g_k(x_1, \dots, x_n) = 0$ ($k = 1 \dots C_n^{p-1}$) les équations obtenues en éliminant $p - 1$ grandeurs.

Montrer la proposition revient à montrer que

$$\begin{aligned} & \exists x_1 \in I_1, \dots, \exists x_n \in I_n \text{ tel que} \\ & \quad f_1(x_1, \dots, x_n) = 0 \\ & \quad \wedge \dots \\ & \quad \wedge f_p(x_1, \dots, x_n) = 0 \\ \Leftrightarrow & \exists x_1 \in I_1, \dots, \exists x_n \in I_n \text{ tel que } g_1(x_1, \dots, x_n) = 0 \\ & \quad \wedge \exists x_1 \in I_1, \dots, \exists x_n \in I_n \text{ tel que } g_2(x_1, \dots, x_n) = 0 \\ & \quad \wedge \dots \\ & \quad \wedge \exists x_1 \in I_1, \dots, \exists x_n \in I_n \text{ tel que } g_q(x_1, \dots, x_n) = 0 \quad (q = C_{p-1}^n) \end{aligned}$$

Pour $n-1$ équations et n variables, la preuve est immédiate. La solution du système d'équations (lorsque le rang est $n-1$) est de la forme $k_1x_1 = k_2x_2 = \dots = k_nx_n$ où k_1, \dots, k_n sont des réels. Vérifier la consistance de ces égalités consiste à vérifier ces égalités deux par deux (propriété 8.3). Or, ces vérifications correspondent aux modèles obtenus en éliminant $n - 2$ variables.

Nous n'avons pas réussi à étendre à un nombre quelconque d'équations.

Une deuxième proposition découle directement de la première et permet d'établir les liens existant entre les modèles de MSD.

Soit un système décrit par p modèles.

La consistance de tous les modèles obtenus en éliminant $i - 1$ ($i \in [1, p]$) grandeurs implique la consistance de tous les modèles obtenus en éliminant $i - 2$ grandeurs.

La preuve de cette proposition est immédiate lorsque la première proposition est démontrée. En effet, cette proposition permet d'assurer que les modèles obtenus en éliminant $i - 2$ grandeurs sont consistants si et seulement si tous les systèmes de $i - 1$ équations sont consistants. Or, la consistance de tous les modèles obtenus en éliminant $i - 1$ grandeurs permet d'assurer la consistance des systèmes contenant i équations, ce qui engendre celle des systèmes contenant $i - 1$ équations.

En théorie, il existe donc un grand nombre de modèles dont la consistance peut être déduite. En effet, la condition nécessaire impose que MSD soit composé des modèles obtenus en éliminant $p - 1$ grandeurs, $p - 2$ grandeurs, etc... (les modèles obtenus en éliminant $p - 2$ grandeurs doivent être conservés car ils permettent la détection d'ensembles de conflits plus petits). La consistance des modèles obtenus en éliminant $p - 2$ grandeurs, etc... peut être déduite de celles des modèles obtenus en éliminant $p - 1$ grandeurs.

En pratique, il semble que cette proposition n'ait pas beaucoup d'intérêt. Ceci découle directement de la faible connectivité des processus considérés. Ainsi, pour un processus dont le comportement est décrit par p équations, il semble que tous les modèles générés permettent d'éliminer au moins $p - 1$ grandeurs même si ces modèles sont issus de la recombinaison de deux équations.

Le seul intérêt de cette proposition est qu'elle permet de donner un fondement à une heuristique permettant de limiter le nombre de modèles générés. En effet, il faut interpréter cette proposition : "plus le nombre de grandeurs invoquées dans l'équation d'un modèle est restreint, plus on a de chances de détecter des ensembles de conflits avec ce modèle", et non pas : "plus on recombine, plus on a de chances de détecter des ensembles de conflits". Or, les modèles contenant le moins de grandeurs sont en règle générale ceux qui sont obtenus à partir d'un nombre restreint de combinaisons. Ces modèles correspondent aussi aux ensembles de conflits les plus petits !! En d'autres termes, plus l'environnement associé à un modèle est petit, plus on a de chance de détecter un ensemble de conflits avec ce modèle. Cette constatation avait déjà été réalisée par [Dague et al, 91] lorsqu'ils proposent comme heuristique de choix de la mesure suivante de prendre une grandeur proche de celle mesurée précédemment. Ce choix correspond à vérifier la consistance des modèles issus d'un nombre réduit de recombinaisons. Une heuristique permettant de limiter le nombre de modèles générés est donc immédiate : soit on limite le nombre de recombinaisons, soit on limite le nombre de grandeurs invoquées dans un modèle.

10.5 Conclusion

Nous avons tenté de présenter DOGS comme un système de diagnostic à part entière, tel que GDE. Nous voulons maintenant atténuer ces propos. D'abord, DOGS n'est qu'une alternative à la procédure d'inférence proposée dans GDE. Les procédures de génération des candidats et de choix de la meilleure mesure n'ont pas été étudiées. L'idée originelle de la compilation est venue de notre incapacité à réaliser un propagateur de SBT. Nous ne sommes d'ailleurs toujours pas convaincus qu'un tel propagateur est réalisable dans le cadre des modèles d'ordre supérieur ou égal à 2. La manière de compiler les modèles (i.e. en éliminant des grandeurs) est

directement inspirée du fonctionnement d'un propagateur. Toutefois, il s'avère que la compilation a de nombreux avantages par rapport à la propagation. Ceux-ci se résument à :

- une confrontation n'est jamais réalisée plusieurs fois,
- certaines confrontations impossibles à réaliser avec un propagateur sont obtenues,
- le problème de la convergence lente disparaît.

Etant donné ces avantages, la compilation semble donc une meilleure solution que la propagation. Toutefois, cette méthode semble difficilement applicable aux systèmes qui ne sont pas décrits par des équations linéaires. De plus, le nombre de modèles à générer est théoriquement très important. Les résultats exposés dans le chapitre suivant montrent que ceci n'est pas le cas en pratique tant au point de vue du nombre de modèles qu'au point de vue du temps de compilation (en comparaison avec celui d'un propagateur).

Chapitre 11

Implémentation et résultats

Ce chapitre est consacré aux résultats obtenus sur le monostable, problème de référence pour les systèmes de diagnostic analogique [Dague et al, 96]. La détection des pannes dépend de notre capacité à générer des suites de bandes temporelles de "bonne qualité". C'est pourquoi nous proposons d'abord une procédure de création des SBT basée sur un critère de qualité des bandes temporelles générées.

11.1 Création des Suites de Bandes Temporelles

Nous proposons dans ce paragraphe un algorithme de calcul de SBT à partir des mesures échantillonnées d'un signal. Nous décrivons ensuite une extension de cet algorithme proposée par [St Marc, 96] et permettant d'éviter l'augmentation de la taille des SBT au cours de l'opération de synchronisation.

11.1.1 Algorithme de création des SBT

Les mesures que nous considérons sont des tableaux de réels issus de l'échantillonnage d'un signal. Nous supposons l'existence d'une procédure permettant de prendre en compte l'imprécision des appareils de mesure. Cette opération nous permet d'obtenir des tableaux d'intervalles (cf. figure 11.1) où chaque intervalle correspond à la valeur du signal en un instant.

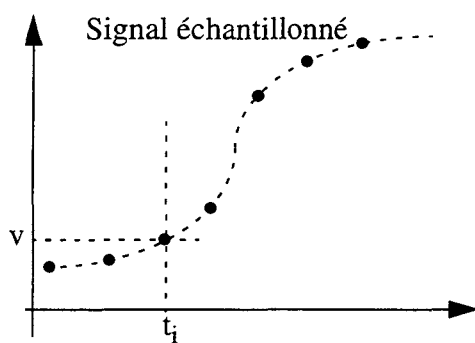


Figure 11.1a : Tableau de réels

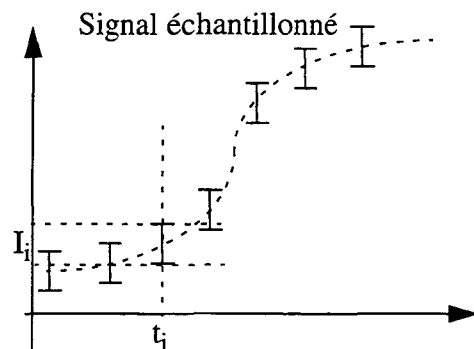


Figure 11.1b : tableau d'intervalles

Nous présentons d'abord une procédure de calcul d'une bande temporelle d'ordre 1, c'est-à-dire une bande temporelle dont les polynômes sont de la forme $p_i * t + o_i$ et $p_s * t + o_s$, à partir d'un tableau d'intervalles.

- Calcul des pentes p_i et p_s

Ce calcul est réalisé par l'intermédiaire de la méthode des moindres carrés¹.

Etant donné un tableau d'intervalles $[I_1, \dots, I_n]$ correspondant aux instants $[t_1, \dots, t_n]$, le calcul de la pente est réalisé avec la formule suivante (directement issue de la méthode des moindres



carrés) :

$$P_i = P_s = \frac{\sum t_i \times y_i - \frac{\sum t_i \times \sum y_i}{n}}{\sum t_i^2 - \frac{\sum t_i \times \sum t_i}{n}}$$

où y_i est le milieu de l'intervalle I_i (i.e. $y_i = (a_i + b_i) / 2$ lorsque $I_i = [a_i, b_i]$)
et t_i est l'instant correspondant à I_i .

C'est l'extension de ce calcul qui peut permettre l'obtention de bandes temporelles d'ordre supérieur. Par exemple, calculer une bande temporelle d'ordre 2 (i.e. dont les polynômes sont de la forme $a t^2 + b t + c$) consiste à étendre cette méthode afin de calculer les réels a et b .

- Calcul des ordonnées à l'origine

Ce calcul est réalisé à partir de la pente p calculée précédemment. Les ordonnées sont calculées de manière à obtenir une bande temporelle contenant tous les échantillons. Soit :

$$p t_j + o_i \leq a_j \leq b_j \leq p t_j + o_s \text{ pour } j = 1 \dots n$$

D'où les formules

$$o_i = \min(a_1 - p t_1, a_2 - p t_2, \dots, a_n - p t_n)$$

$$o_s = \max(b_1 - p t_1, b_2 - p t_2, \dots, b_n - p t_n)$$

Etant donné un tableau d'intervalles, les formules précédentes permettent de créer une bande temporelle d'ordre 1. Nous proposons maintenant un critère de qualité pour une bande temporelle.

- Critère de largeur

Le critère de qualité d'une bande temporelle est exprimé en pourcentage d'imprécision introduite par rapport aux intervalles. Ce critère est appelé le critère de largeur CR_LA.

Une bande temporelle est dite de bonne qualité si et seulement si la largeur de cette bande temporelle (i.e. $o_s - o_i$) est inférieur au critère de largeur (fixé par l'utilisateur) multiplié par la plus petite largeur des intervalles considérés.

D'une manière formelle :

Soit un tableau d'intervalles $[I_1, \dots, I_n]$ ($I_i = [a_i, b_i]$) et une bande temporelle de polynômes $p t + o_i$ et $p t + o_s$ créée à partir de ce tableau.

La bande temporelle est de bonne qualité si et seulement si

$$o_s - o_i < CR_LA * \min(b_1 - a_1, \dots, b_n - a_n)$$

L'hypothèse fondamentale est que la bande temporelle créée contient le signal réel non

1. Cette méthode est détaillée dans de nombreux ouvrages de méthodes de calcul numérique. Nous en citons un : [Dahlquist et al, 74].

complètement observable. Nous discutons maintenant de la validité de notre procédure par rapport à cette hypothèse.

Hypothèses de création des bandes temporelles

Lorsque le signal est correctement échantillonné, c'est à dire que le théorème de Shannon est respecté, l'allure du signal entre deux échantillons consécutifs peut être approximée par une droite reliant deux des points des intervalles. Comme toutes ces droites appartiennent à la bande temporelle créée (cf. figure 11.2), l'hypothèse que le signal est encadré par la bande temporelle est correcte.

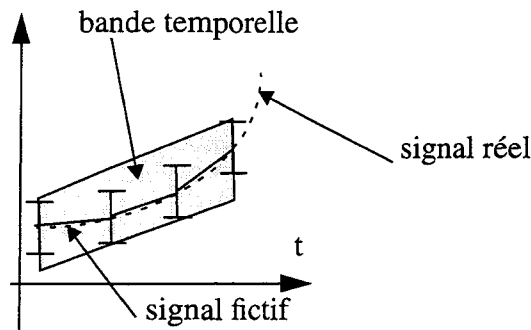


Figure 11.2 : Signal correctement échantillonné

Malheureusement, il existe de nombreux cas où le théorème de Shannon ne peut pas être respecté. Ceci est le cas lorsqu'un signal varie rapidement par rapport à la fréquence du système considéré. Le signal ne peut alors plus être considéré comme une droite et n'est donc pas nécessairement représenté par la bande temporelle créée (cf. figure 11.3).

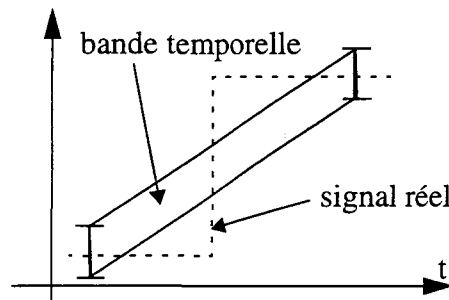


Figure 11.3 : Signal non correctement échantillonné

Nous proposons un critère de bon échantillonnage des signaux.

Un signal sera considéré correctement échantillonné lorsque les intervalles consécutifs s'intersectent.

Les signaux mal échantillonnés (i.e. le critère n'est pas respecté) seront représentés par des bandes temporelles de pente nulle (cf. figure 11.4)

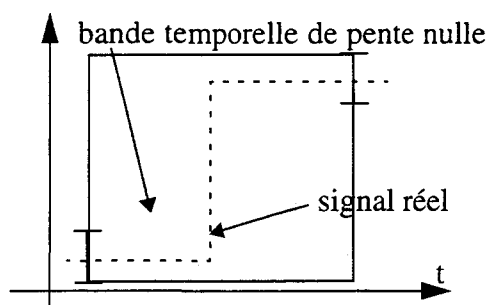


Figure 11.4 : Signal mal échantillonné

Il est clair que le critère de bon échantillonnage est subjectif et ne représente pas une garantie totale. Un exemple d'échantillonnage où ce critère n'est pas suffisant est illustré figure 11.5.

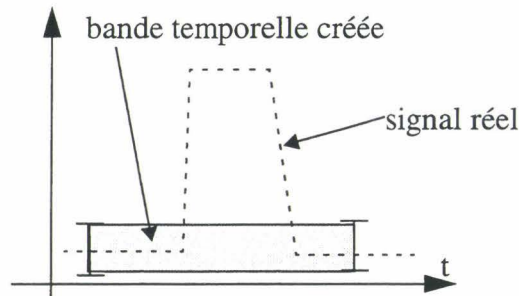


Figure 11.5 : Critère insuffisant

Nous admettons que le critère de bon échantillonnage est suffisant en pratique (i.e. le cas présenté ci dessus n'arrive jamais).

Nous avons présenté une méthode permettant de calculer une bande temporelle à partir d'un tableau d'intervalles. Nous étendons maintenant cette méthode afin de calculer une SBT contenant uniquement des bandes temporelles de bonne qualité (pour le critère de largeur). Le principe de cet algorithme est de calculer une bande temporelle avec deux échantillons puis d'élargir l'extension temporelle.

Soit le tableau d'intervalles $[I_1, \dots, I_n]$ (avec $I_i = [a_i, b_i]$ et $y_i = (a_i + b_i) / 2$) correspondant aux instants $[t_1, \dots, t_n]$.

Nous notons

- TAB : le tableau d'intervalles à encadrer par des bandes temporelles (TAB est initialisé à $[I_1, \dots, I_n]$)
- TAB_ACT : le tableau des intervalles encadrés par la bande temporelle en cours de calcul

L'algorithme consiste à :

étape 1 : initialisation

si les deux premiers éléments de TAB ne s'intersectent pas alors
 enregistrer la bande de pente 0 et contenant ces deux éléments
 supprimer le premier élément de TAB
 retourner à l'étape 1

sinon

initialiser TAB_ACT avec les deux premiers éléments de TAB
 supprimer les deux premiers éléments de TAB
 aller à l'étape 2

fsi

étape 2 : extension de la bande temporelle (on essaie de faire rentrer le premier élément de TAB dans la bande temporelle)

si le dernier élément de TAB_ACT n'intersecte pas le premier élément de TAB alors
 enregistrer la bande correspondant à TAB_ACT (les instants de début et de fin sont les instants correspondant aux premier et dernier élément de TAB_ACT, la bande et les ordonnées à l'origine sont calculées avec la méthode des moindres carrés)

enregistrer la bande de pente 0 et contenant dernier élément de TAB_ACT et le premier élément de TAB

retourner à l'étape 1

sinon

calculer les paramètres de la bande correspondant à TAB_ACT + le premier élément de TAB

si critère de largeur est vérifié alors

supprimer le premier élément de TAB et l'ajouter dans TAB_ACT

retourner à l'étape 2

sinon

enregistrer la bande correspondant à TAB_ACT

retourner à l'étape 1

fsi

fsi

Remarque :

La minimalité du nombre de bandes temporelles des SBT dépend de la procédure de création d'une bande temporelle à partir des intervalles. On dira que la procédure de création d'une bande temporelle est idéale si et seulement si lorsqu'une bande temporelle créée à partir d'un tableau d'intervalles correspondant à un intervalle de temps T ne respecte pas le critère de largeur alors toute bande temporelle créée à partir d'un tableau d'intervalles correspondant à un intervalle de temps contenant T ne respecte pas non plus le critère de largeur. Lorsque la procédure de création d'une bande temporelle est idéale alors la SBT créée avec notre algorithme est de taille minimale.

L'instant de fin des bandes temporelles est indirectement fixé par le critère de largeur. Celui ci a donc une influence sur le nombre de bandes temporelles des SBT. Ce phénomène est illustré figure 11.6. Le signal échantillonné est une sinusoïde.

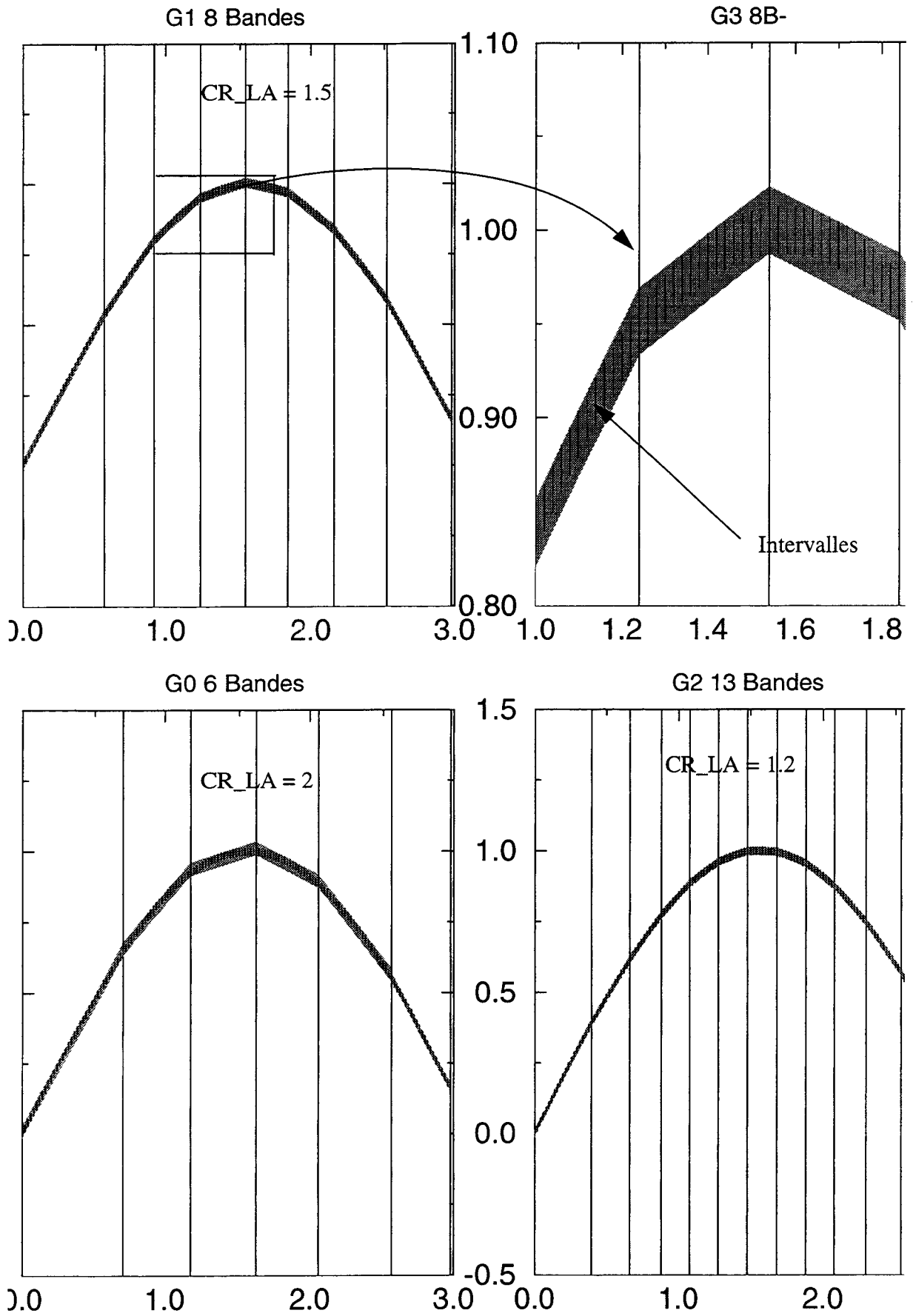


Figure 11.6 :Influence du critère de largeur

Nous proposons maintenant une extension de l'algorithme de création des SBT permettant d'éviter l'augmentation de la taille des SBT au cours de l'opération de synchronisation. Cette extension a été proposée par [St Marc, 96].

11.1.2 Extension de l'algorithme

Dans les circuits électroniques analogiques, les formes de certains signaux d'un système sont en règle générale fortement liées. Par exemple, la croissance d'un signal engendre souvent la décroissance ou la croissance d'autres signaux. Par conséquent, il semble logique que les différentes SBT représentant les différents signaux liés soient synchrones. Or, ceci n'est pas le cas avec l'algorithme que nous avons proposé car les instants de fin des bandes temporelles sont indirectement déterminés par le critère de largeur. Un exemple caractéristique de ce phénomène est que les SBT calculées avec notre algorithme en partant du premier ou du dernier élément du tableau d'intervalles sont en règle générale différentes. Toutefois, ces deux SBT sont de tailles identiques lorsque la procédure de création d'une bande temporelle est idéale. Par contre, les instants de coupure et les bandes temporelles de ces SBT sont différents.

Le principe de la synchronisation des SBT découle directement de cette remarque. Nous supposons que notre procédure de création d'une bande temporelle est idéale. Par conséquent, les SBT calculées à partir du premier et du dernier élément du tableau d'intervalles, notées SB_1 et SB_2 , sont de tailles identiques. De plus, les instants de coupure de SB_1 et de SB_2 , notées $[x_1, \dots, x_n]$ et $[y_1, \dots, y_n]$, correspondent respectivement aux instants de coupure au plus tard et au plus tôt, c'est à dire que $y_i < x_i < y_{i+1}$ et que toute SBT dont l'instant de coupure de la i ème bande temporelle n'est pas compris entre y_i et x_i ne sera pas de taille minimale (w.r.t. le critère de largeur). Par contre, l'inverse n'est pas vrai : toute SBT dont les instants de coupure sont compris entre y_i et x_i ne respectera pas nécessairement le critère de largeur (i.e. une des bandes temporelles de cette SBT ne respectera pas le critère de largeur).

L'algorithme de synchronisation consiste à calculer les instants de coupure avant de créer les SBT.

Pour la première SBT :

Calculer les SBT à partir du premier et du dernier échantillon. Soient $[x_1, \dots, x_n]$ et $[y_1, \dots, y_n]$ les instants de coupure de ces deux SBT.

Calculer la SBT dont les instants de coupure sont $(x_i + y_i) / 2$.

Créer la liste des instants de coupure, c'est à dire la liste des instants $[(x_1 + y_1) / 2, \dots, (x_n + y_n) / 2]$

Pour les SBT suivantes :

Calculer les SBT à partir du premier et du dernier échantillon. Soient $[v_1, \dots, v_p]$ et $[z_1, \dots, z_p]$ les instants de coupure de ces deux SBT.

Calculer la SBT dont l'instant de coupure de la i ème bande est

- l'instant t_i appartenant à la liste des instants de coupure et à $[z_i, v_i]$,
- $(z_i + v_i) / 2$ sinon (dans ce cas, rajouter $(z_i + v_i) / 2$ à la liste des instants de coupure).

Nous présentons maintenant les résultats obtenus sur le monostable. Le critère de largeur choisi pour cette expérimentation est 1.5.

11.2 DOGS et le monostable

Nous présentons d'abord brièvement le circuit et le fonctionnement du monostable (une explication plus complète est dans [Dague et al, 96]). Nous présentons ensuite les modèles générés par notre compilateur et enfin le diagnostic de quelques pannes.

11.2.1 Le monostable

Le schéma du monostable est illustré figure 11.7.

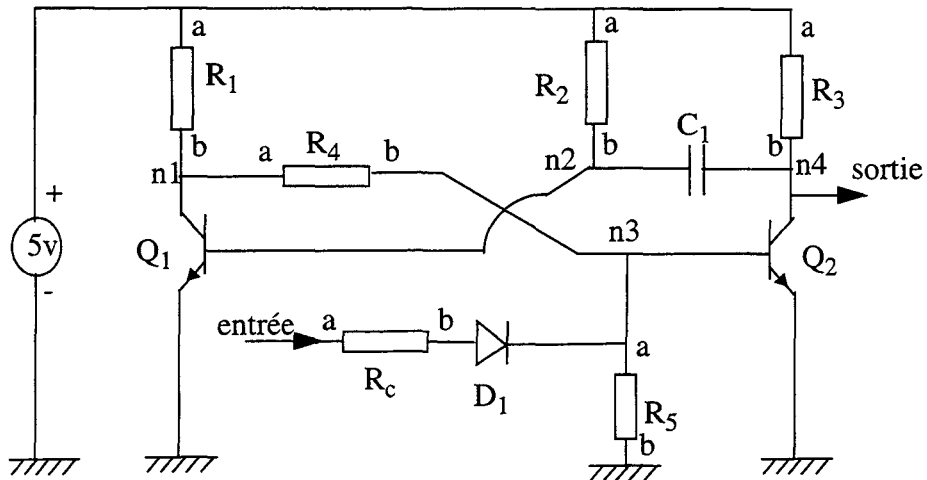


Figure 11.7 : Le monostable

But du circuit :

Le but du monostable est de générer un signal carré dont la durée (i.e. l'intervalle de temps θ) est complètement déterminée par la valeur des composants du circuit. Typiquement, les entrée et sortie de ce circuit sont de la forme (cf. figure 11.8)

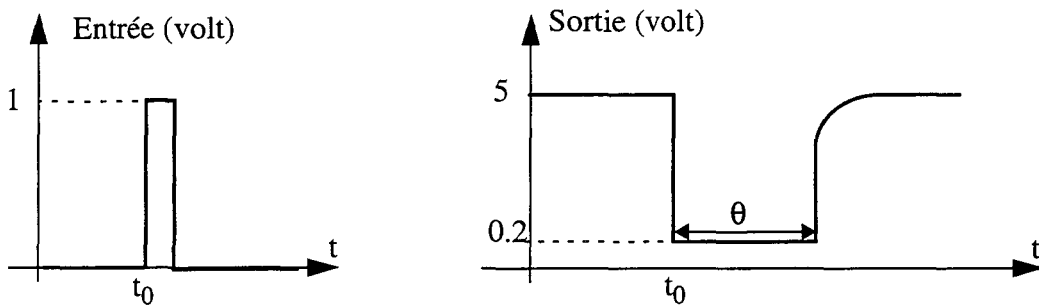


Figure 11.8 : Signaux d'entrée et de sortie du monostable

Fonctionnement du circuit :

Etat initial :

Lorsque l'entrée est à 0 volt (jusqu'à l'instant t_0), la base du transistor q_2 est aussi à 0 volt. Ce transistor est bloqué. Aucun courant ne passe dans R_3 et la sortie est à 5 volt (fixé par l'alimentation).

Le régime du transistor q_1 est fixé par les résistances r_1 et r_2 : ce transistor est saturé ; sa base est à 0.6v et la tension collecteur-émetteur est à 0.2v.

La capacité c_1 est chargée. La tension à ses bornes est $5 - 0.6 = 4.4v$

Changement à t_0 :

A l'instant t_0 , la tension fournie à l'entrée permet au transistor q2 de devenir saturé. La tension de sortie est donc 0.2 volt. La capacité étant chargée à 4.4 v, la base du transistor q1 devient $0.6 - (5 - 0.2) = -4.2$ volt. Le transistor q1 est donc bloqué. La saturation de q2 est assurée par les résistances r1, r3 et r4.

Etat intermédiaire :

La capacité c1 va se décharger à travers r2 durant la période $r2 * c1$ (c'est cette période qui fixe la durée θ). La tension de la base de q1 va augmenter exponentiellement.

Retour à l'état stable :

A l'instant $t_0 + \theta$, la tension de la base de q1 devient 0.6 volt et q1 redevient saturé. La tension de base de q2 devient alors quasi nulle car la tension collecteur de q1 est à 0.2v et la tension d'entrée est nulle. q2 redevient donc bloqué. La capacité c1 se charge alors à travers la résistance r3 en un temps très court car la résistance r3 est beaucoup plus faible que r2. Le système se retrouve alors dans son état stable.

Nous présentons maintenant les modèles des composants et les modèles générés par le compilateur.

11.2.2 Compilation des modèles

Modèles initiaux

(1) label : [r1]	$i(r1,a) + i(r1,b) = 0$
(2) label : [r1]	$v(r1,a,b) + r1 * i(r1,b) = 0$
(3) label : [r2]	$i(r2,a) + i(r2,b) = 0$
(4) label : [r2]	$v(r2,a,b) + r2 * i(r2,b) = 0$
(5) label : [r3]	$i(r3,a) + i(r3,b) = 0$
(6) label : [r3]	$v(r3,a,b) + r3 * i(r3,b) = 0$
(7) label : [r4]	$i(r4,a) + i(r4,b) = 0$
(8) label : [r4]	$v(r4,a,b) + r4 * i(r4,b) = 0$
(9) label : [r5]	$i(r5,a) + i(r5,b) = 0$
(10) label : [r5]	$v(r5,a,b) + r5 * i(r5,b) = 0$
(11) label : [rc]	$i(rc,a) + i(rc,b) = 0$
(12) label : [rc]	$v(rc,a,b) + rc * i(rc,b) = 0$
(13) label : [q1]	$i(q1,b) + i(q1,c) + i(q1,e) = 0$
(14) label : [q1]	$v(q1,b,e) - v_{be}(q1) = 0$
(15) label : [q1]	$i(q1,c) - val_{ic}(q1) = 0^1$

1. Ce modèle permet de définir le courant traversant le transistor. L'observation de $val_{ic}(q1)$ est connue à priori. Cette valeur dépend de conditions sur les tensions aux bornes du transistor, tensions qui permettent de déterminer son mode de fonctionnement : bloqué, passant ou saturé.

(16) label : [q1]	$i(q1,b) - val_ib(q1) = 0$
(17) label : [q1]	$- beta(q1) * i(q1,b) + i(q1,c) = 0$
(18) label : [q2]	$i(q2,b) + i(q2,c) + i(q2,e) = 0$
(19) label : [q2]	$v(q2,b,e) - vbe(q2) = 0$
(20) label : [q2]	$i(q2,c) - val_ic(q2) = 0$
(21) label : [q2]	$i(q2,b) - val_ib(q2) = 0$
(22) label : [q2]	$- beta(q2) * i(q2,b) + i(q2,c) = 0$
(23) label : [d1]	$v(d1,a,c) - vac(d1) = 0$
(24) label : [d1]	$i(d1,a) - i(d1) = 0$
(25) label : [d1]	$i(d1,c) + i(d1,a) = 0$
(26) label : [g1]	$v(g1,a,b) - tension(g1) = 0$
(27) label : [g1]	$i(g1,a) + i(g1,b) = 0$
(28) label : [c1]	$i(c1,a) + i(c1,b) = 0$
(29) label : [c1]	$i(c1,b) + c1 * derivee(v(c1,a,b)) = 0$
(30) label : [n1]	$i(q1,c) + i(r4,a) + i(r1,b) = 0$
(31) label : [n2]	$i(c1,a) + i(q1,b) + i(r2,b) = 0$
(32) label : [n3]	$i(d1,c) + i(q2,b) + i(r5,a) + i(r4,b) = 0$
(33) label : [n4]	$i(c1,b) + i(q2,c) + i(r3,b) = 0$
(34) label : [n5]	$i(d1,a) + i(rc,b) = 0$
(35) label : [n7]	$i(g1,a) + i(r3,a) + i(r2,a) + i(r1,a) = 0$

Les observations des grandeurs $val_ib(q1)$, $val_ic(q1)$, $val_ib(q2)$, $val_ic(q2)$, $vbe(q1)$, $vbe(q2)$, $vac(d1)$, $i(d1)$ et $tension(g1)$ sont prédéfinies. La valeur de ces observations peut dépendre de contraintes (i.e. les préconditions).

Equations générées (i.e. élimination des grandeurs non mesurables)

(1) label : [q1]	$v(q1,b,e) - vbe(q1) = 0$
(2) label : [q1]	$- val_ic(q1) / beta(q1) + val_ib(q1) = 0$
(3) label : [q2]	$v(q2,b,e) - vbe(q2) = 0$
(4) label : [q2]	$- val_ic(q2) / beta(q2) + val_ib(q2) = 0$
(5) label : [d1]	$v(d1,a,c) - vac(d1) = 0$
(6) label : [g1]	$v(g1,a,b) - tension(g1) = 0$
(7) label : [n1,q1,r4,r1]	$- v(r1,a,b) / r1 + v(r4,a,b) / r4 + val_ic(q1) = 0$
(8) label : [n3,d1,q2,r5,r4]	$v(r5,a,b) / r5 - v(r4,a,b) / r4 + val_ib(q2) - i(d1) = 0$
(9) label : [n4,n2,c1,q2,q1,r3,r2]	$- v(r2,a,b) / r2 - v(r3,a,b) / r3 + val_ib(q1) + val_ic(q2) = 0$
(10) label : [n5,d1,rc]	$- v(rc,a,b) / rc + i(d1) = 0$
(11) label : [n5,n3,d1,q2,rc,r5,r4]	$- v(rc,a,b) / rc + v(r5,a,b) / r5 - v(r4,a,b) / r4 + val_ib(q2) = 0$
(12) label : [n3,d1,q2,r5,r4]	$v(r5,a,b) / r5 - v(r4,a,b) / r4 + val_ic(q2) / beta(q2) - i(d1) = 0$
(13) label : [n4,n2,c1,q2,q1,r3,r2]	$- v(r2,a,b) / r2 - v(r3,a,b) / r3 + val_ib(q1) + beta(q2) * val_ib(q2) = 0$
(14) label : [n4,n2,c1,q2,q1,r3,r2]	$- v(r2,a,b) / r2 - v(r3,a,b) / r3 + val_ic(q1) / beta(q1) + val_ic(q2) = 0$
(15) label : [n1,q1,r4,r1]	$- v(r1,a,b) / r1 + v(r4,a,b) / r4 + beta(q1) * val_ib(q1) = 0$
(16) label : [n3,n1,d1,q2,q1,r5,r4,r1]	

- (17) label : [n5,n3,d1,q2,rc,r5,r4]
 $-v(r5,a,b)/r5 + v(r1,a,b)/r1 - \text{val_ic}(q1) - \text{val_ib}(q2) + i(d1) = 0$
- (18) label : [n5,n3,n1,d1,q2,q1,rc,r5,r4,r1]
 $-v(rc,a,b)/rc + v(r5,a,b)/r5 - v(r4,a,b)/r4 + \text{val_ic}(q2)/\text{beta}(q2) = 0$
- (19) label : [n4,n3,n2,c1,d1,q2,q1,r5,r4,r3,r2]
 $v(rc,a,b)/rc - v(r5,a,b)/r5 + v(r1,a,b)/r1 - \text{val_ic}(q1) - \text{val_ib}(q2) = 0$
- (20) label : [n3,n1,d1,q2,q1,r5,r4,r1]
 $v(r2,a,b)/r2 + v(r3,a,b)/r3 + \text{beta}(q2) * v(r5,a,b)/r5 - (\text{beta}(q2)/r4) * v(r4,a,b) - \text{val_ib}(q1) - \text{beta}(q2) * i(d1) = 0$
- (21) label : [n4,n2,c1,q2,q1,r3,r2]
 $-v(r5,a,b)/r5 + v(r1,a,b)/r1 - \text{val_ic}(q1) - \text{val_ic}(q2)/\text{beta}(q2) + i(d1) = 0$
- (22) label : [n4,n2,n1,c1,q2,q1,r4,r3,r2,r1]
 $-v(r2,a,b)/r2 - v(r3,a,b)/r3 + \text{val_ic}(q1)/\text{beta}(q1) + \text{beta}(q2) * \text{val_ib}(q2) = 0$
- (23) label : [n4,n2,n1,c1,q2,q1,r4,r3,r2,r1]
 $\text{beta}(q1) * (-v(r2,a,b)/r2 - v(r3,a,b)/r3) + v(r1,a,b)/r1 - v(r4,a,b)/r4 + \text{beta}(q1) * \text{val_ic}(q2) = 0$
- (24) label : [n3,n1,d1,q2,q1,r5,r4,r1]
 $v(r5,a,b)/r5 - v(r1,a,b)/r1 + \text{beta}(q1) * \text{val_ib}(q1) + \text{val_ib}(q2) - i(d1) = 0$
- (25) label : [n3,n1,d1,q2,q1,r5,r4,r1]
 $-v(r5,a,b)/r5 + v(r1,a,b)/r1 - \text{val_ic}(q1) - \text{val_ib}(q2) + i(d1) = 0$
- (26) label : [n5,n3,n1,d1,q2,q1,rc,r5,r4,r1]
 $v(rc,a,b)/rc - v(r5,a,b)/r5 + v(r1,a,b)/r1 - \text{val_ic}(q1) - \text{val_ib}(q2) = 0$
- (27) label : [n3,n1,d1,q2,q1,r5,r4,r1]
 $-v(r5,a,b)/r5 + v(r1,a,b)/r1 - \text{val_ic}(q1) - \text{val_ic}(q2)/\text{beta}(q2) + i(d1) = 0$
- (28) label : [n4,n2,c1,q2,q1,r3,r2]
 $-v(r2,a,b)/r2 - v(r3,a,b)/r3 + \text{val_ic}(q1)/\text{beta}(q1) + \text{beta}(q2) * \text{val_ib}(q2) = 0$
- (29) label : [n3,n1,d1,q2,q1,r5,r4,r1]
 $v(r5,a,b)/r5 - v(r1,a,b)/r1 + \text{beta}(q1) * \text{val_ib}(q1) + \text{val_ib}(q2) - i(d1) = 0$
- (30) label : [n5,n4,n3,n2,c1,d1,q2,q1,rc,r5,r4,r3,r2]
 $\text{beta}(q2) * (-v(rc,a,b)/rc + v(r2,a,b)/r2 + v(r3,a,b)/r3 + (\text{beta}(q2)/r5) * v(r5,a,b) - (\text{beta}(q2)/r4) * v(r4,a,b) - \text{val_ib}(q1) = 0$
- (31) label : [n5,n3,n1,d1,q2,q1,rc,r5,r4,r1]
 $v(rc,a,b)/rc - v(r5,a,b)/r5 + v(r1,a,b)/r1 - \text{val_ic}(q1) - \text{val_ic}(q2)/\text{beta}(q2) = 0$
- (32) label : [n5,n3,n1,d1,q2,q1,rc,r5,r4,r1]
 $-v(rc,a,b)/rc + v(r5,a,b)/r5 - v(r1,a,b)/r1 + \text{beta}(q1) * \text{val_ib}(q1) + \text{val_ib}(q2) = 0$
- (33) label : [n4,n3,n2,c1,d1,q2,q1,r5,r4,r3,r2]
 $-v(r2,a,b)/r2 - v(r3,a,b)/r3 - (\text{beta}(q2) * v(r5,a,b)/r5) + (\text{beta}(q2)/r4) * v(r4,a,b) + \text{val_ic}(q1)/\text{beta}(q1) + \text{beta}(q2) * i(d1) = 0$
- (34) label : [n4,n3,n2,n1,c1,d1,q2,q1,r5,r4,r3,r2,r1]
 $1/\text{beta}(q2) * (-v(r2,a,b)/r2 - v(r3,a,b)/r3) - v(r5,a,b)/r5 + v(r1,a,b)/r1 - \text{val_ic}(q1) + (1/\text{beta}(q2)) * \text{val_ib}(q1) + i(d1) = 0$
- (35) label : [n3,n1,d1,q2,q1,r5,r4,r1]
 $v(r5,a,b)/r5 - v(r1,a,b)/r1 + \text{beta}(q1) * \text{val_ib}(q1) + \text{val_ic}(q2)/\text{beta}(q2) - i(d1) = 0$
- (36) label : [n4,n2,n1,c1,q2,q1,r4,r3,r2,r1]
 $\text{beta}(q1) * (-v(r2,a,b)/r2 - v(r3,a,b)/r3) + v(r1,a,b)/r1 - v(r4,a,b)/r4 + \text{beta}(q2) * \text{beta}(q1) * \text{val_ib}(q2) = 0$
- (37) label : [n4,n3,n2,n1,c1,d1,q2,q1,r5,r4,r3,r2,r1]
 $\text{beta}(q1) * (-v(r2,a,b)/r2 - v(r3,a,b)/r3) - v(r5,a,b)/r5 + v(r1,a,b)/r1 + \text{beta}(q1) * \text{val_ic}(q2) - \text{val_ib}(q2) + i(d1) = 0$
- (38) label : [n5,n4,n3,n2,c1,d1,q2,q1,rc,r5,r4,r3,r2]
 $\text{beta}(q2) * v(rc,a,b)/rc - v(r2,a,b)/r2 - v(r3,a,b)/r3 - (\text{beta}(q2)/r5) * v(r5,a,b)$

- (38) label : [n5,n4,n3,n2,n1,c1,d1,q2,q1,rc,r5,r4,r3,r2,r1]

$$+ (\text{beta}(q2) / r4) * v(r4,a,b) + \text{val_ic}(q1) / \text{beta}(q1) = 0$$

$$v(rc,a,b) / rc + 1 / \text{beta}(q2) * (-v(r2,a,b) / r2 - v(r3,a,b) / r3) - v(r5,a,b) / r5$$

$$+ v(r1,a,b) / r1 - \text{val_ic}(q1) + \text{val_ib}(q1) / \text{beta}(q2) = 0$$
- (39) label : [n5,n3,n1,d1,q2,q1,rc,r5,r4,r1]

$$-v(rc,a,b)/rc + v(r5,a,b)/r5 - v(r1,a,b)/r1 + \text{beta}(q1) * \text{val_ib}(q1) + \text{val_ic}(q2)/\text{beta}(q2) = 0$$
- (40) label : [n5,n4,n3,n2,n1,c1,d1,q2,q1,rc,r5,r4,r3,r2,r1]

$$-v(rc,a,b) / (rc * \text{beta}(q1)) + v(r2,a,b) / r2 + v(r3,a,b) / r3 + v(r5,a,b) / (r5 * \text{beta}(q1))$$

$$- v(r1,a,b) / (r1 * \text{beta}(q1)) - \text{val_ic}(q2) + \text{val_ib}(q2) / \text{beta}(q1) = 0$$
- (41) label : [n4,n3,n2,n1,c1,d1,q2,q1,r5,r4,r3,r2,r1]

$$\text{beta}(q1) * (-v(r2,a,b) / r2 - v(r3,a,b) / r3) - \text{beta}(q2) * \text{beta}(q1) * (v(r5,a,b) / r5)$$

$$+ v(r1,a,b) / r1 + (\text{beta}(q2) * \text{beta}(q1) - 1) * v(r4,a,b) / r4 + \text{beta}(q2) * \text{beta}(q1) * i(d1) = 0$$
- (42) label : [n4,n3,n2,n1,c1,d1,q2,q1,r5,r4,r3,r2,r1]

$$1 / \text{beta}(q2) * (-v(r2,a,b) / r2 - v(r3,a,b) / r3) - v(r5,a,b) / r5 + v(r1,a,b) / r1$$

$$+ (1 / (\text{beta}(q1) * \text{beta}(q2)) - 1) * \text{val_ic}(q1) + i(d1) = 0$$
- (43) label : [n4,n3,n2,n1,c1,d1,q2,q1,r5,r4,r3,r2,r1]

$$(-v(r2,a,b) / r2 - v(r3,a,b) / r3) / \text{beta}(q2) - v(r5,a,b) / r5 + v(r1,a,b) / r1$$

$$+ (1 / \text{beta}(q2) - \text{beta}(q1)) * \text{val_ib}(q1) + i(d1) = 0$$
- (44) label : [n4,n3,n2,n1,c1,d1,q2,q1,r5,r4,r3,r2,r1]

$$v(r2,a,b) / r2 + v(r3,a,b) / r3 + v(r5,a,b) / (r5 * \text{beta}(q1)) - v(r1,a,b) / (r1 * \text{beta}(q1))$$

$$+ ((1 / (\text{beta}(q2) * \text{beta}(q1)) - 1) * \text{val_ic}(q2) + -i(d1) / \text{beta}(q1)) = 0$$
- (45) label : [n4,n3,n2,n1,c1,d1,q2,q1,r5,r4,r3,r2,r1]

$$\text{beta}(q1) * (-v(r2,a,b) / r2 - v(r3,a,b) / r3) - v(r5,a,b) / r5 + v(r1,a,b) / r1$$

$$+ ((\text{beta}(q2) * \text{beta}(q1) - 1) * \text{val_ib}(q2) + i(d1)) = 0$$
- (46) label : [n5,n4,n3,n2,n1,c1,d1,q2,q1,rc,r5,r4,r3,r2,r1]

$$\text{beta}(q2) * \text{beta}(q1) * v(rc,a,b) / rc + \text{beta}(q1) * (-v(r2,a,b) / r2 - v(r3,a,b) / r3) + v(r1,a,b) / r1$$

$$- \text{beta}(q2) * \text{beta}(q1) * v(r5,a,b) / r5 + (\text{beta}(q2) * \text{beta}(q1) - 1) * v(r4,a,b) / r4 = 0$$
- (47) label : [n5,n4,n3,n2,n1,c1,d1,q2,q1,rc,r5,r4,r3,r2,r1]

$$v(rc,a,b) / rc + 1 / \text{beta}(q2) * (-v(r2,a,b) / r2 - v(r3,a,b) / r3) - v(r5,a,b) / r5$$

$$+ v(r1,a,b) / r1 + (1 / (\text{beta}(q1) * \text{beta}(q2)) - 1) * \text{val_ic}(q1) = 0$$
- (48) label : [n5,n4,n3,n2,n1,c1,d1,q2,q1,rc,r5,r4,r3,r2,r1]

$$v(rc,a,b) / rc + (-v(r2,a,b) / r2 - v(r3,a,b) / r3) / \text{beta}(q2) - v(r5,a,b) / r5 + v(r1,a,b) / r1$$

$$+ (1 / \text{beta}(q2) - \text{beta}(q1)) * \text{val_ib}(q1) = 0$$
- (49) label : [n5,n4,n3,n2,n1,c1,d1,q2,q1,rc,r5,r4,r3,r2,r1]

$$-v(rc,a,b) / (rc * \text{beta}(q1)) + v(r2,a,b) / r2 + v(r3,a,b) / r3 + v(r5,a,b) / (r5 * \text{beta}(q1))$$

$$- v(r1,a,b) / (r1 * \text{beta}(q1)) + (1 / (\text{beta}(q2) * \text{beta}(q1)) - 1) * \text{val_ic}(q2) = 0$$
- (50) label : [n5,n4,n3,n2,n1,c1,d1,q2,q1,rc,r5,r4,r3,r2,r1]

$$1 / \text{beta}(q1) * v(rc,a,b) / rc - v(r2,a,b) / r2 - v(r3,a,b) / r3 - v(r5,a,b) / (r5 * \text{beta}(q1))$$

$$+ v(r1,a,b) / (r1 * \text{beta}(q1)) - (1 / \text{beta}(q1) - \text{beta}(q2)) * \text{val_ib}(q2) = 0$$
- (51) label : [c1, q1, d1,q2,r5,n3,r4,r2,n2,r1,n1]

$$-i(d1) / \text{beta}(q1) + \text{val_ic}(q2) / (\text{beta}(q2) * \text{beta}(q1)) + v(r5,a,b) / r5 * \text{beta}(q1)$$

$$+ v(r2,a,b) / r2 - v(r1,a,b) / (r1 * \text{beta}(q1)) + c1 * \text{derivee}(v(c1,a,b)) = 0$$
- (52) label : [c1,q2, d1,r5,n3,r4,r3,n4]

$$-i(d1) + v(r5,a,b) / r5 - v(r4,a,b) / r4 + v(r3,a,b) / (r3 * \text{beta}(q2))$$

$$- (c1 / \text{beta}(q2)) * \text{derivee}(v(c1,a,b)) = 0$$
- (53) label : [c1,d1,q2,q1,r5,n3,r4,r3,n4,r1,n1]

$$-i(d1) + \text{beta}(q1) * \text{val_ib}(q1) + v(r5,a,b) / r5 + v(r3,a,b) / (r3 * \text{beta}(q2))$$

$$- v(r1,a,b) / r1 - (c1 / \text{beta}(q2)) * \text{derivee}(v(c1,a,b)) = 0$$
- (54) label : [c1,d1,q1,q2,r5,n3,r4,r3,n4,r1,n1]

$$-i(d1) + \text{val_ic}(q1) + v(r5,a,b) / r5 + v(r3,a,b) / (r3 * \text{beta}(q2)) - v(r1,a,b) / r1$$

- (55) label : [c1,d1,q1,q2,r5,n3,r4,r2,n2,r1,n1]

$$-(c1 / \beta(q2)) * \text{derivee}(v(c1,a,b)) = 0$$

$$-i(d1) / \beta(q1) + \text{val_ib}(q2) / \beta(q1) + v(r5,a,b) / (r5*\beta(q1)) + v(r2,a,b) / r2$$

$$- v(r1,a,b) / (r1*\beta(q1)) + c1 * \text{derivee}(v(c1,a,b)) = 0$$
- (56) label : [c1,q2,r3,n4]

$$\beta(q2) * \text{val_ib}(q2) - v(r3,a,b) / r3 + c1 * \text{derivee}(v(c1,a,b)) = 0$$
- (57) label : [c1,d1,q1,q2,rc,n5,r5,n3,r4,r2,n2,r1,n1]

$$\text{val_ic}(q2) / (\beta(q2)*\beta(q1)) - v(rc,a,b) / (rc*\beta(q1)) + v(r5,a,b) / (r5*\beta(q1))$$

$$+ v(r2,a,b) / r2 - v(r1,a,b) / (r1*\beta(q1)) + c1 * \text{derivee}(v(c1,a,b)) = 0$$
- (58) label : [c1,rc,d1,n5,r5,n3,r4,r3,n4]

$$- v(rc,a,b) / rc + v(r5,a,b) / r5 - v(r4,a,b) / r4 + v(r3,a,b) / (r3*\beta(q2))$$

$$- (c1 / \beta(q2)) * \text{derivee}(v(c1,a,b)) = 0$$
- (59) label : [c1,q1,d1,q2,rc,n5,r5,n3,r4,r3,n4,r1,n1]

$$\text{val_ib}(q1) - v(rc,a,b) / (rc*\beta(q1)) + v(r5,a,b) / (r5*\beta(q1)) + v(r3,a,b) /$$

$$(r3*\beta(q2)*\beta(q1)) - v(r1,a,b) / (r1*\beta(q1)) - (c1 / (\beta(q2)*\beta(q1))) * \text{derivee}(v(c1,a,b)) = 0$$
- (60) (64) label : [c1,d1,q1,rc,n5,r5,n3,r4,r3,n4,r1,n1]

$$\text{val_ic}(q1) - v(rc,a,b) / rc + v(r5,a,b) / r5 + v(r3,a,b) / (r3*\beta(q2)) - v(r1,a,b) / r1$$

$$-(c1/\beta(q2)) * \text{derivee}(v(c1,a,b)) = 0$$
- (61) label : [c1,d1,q1,q2,rc,n5,r5,n3,r4,r2,n2,r1,n1]

$$\text{val_ib}(q2) / \beta(q1) - v(rc,a,b) / (rc*\beta(q1)) + v(r5,a,b) / (r5*\beta(q1)) + v(r2,a,b) / r2$$

$$- v(r1,a,b) / (r1*\beta(q1)) + c1 * \text{derivee}(v(c1,a,b)) = 0$$
- (62) label : [c1,q2,r3,n4]

$$\text{val_ic}(q2) - v(r3,a,b) / r3 + c1 * \text{derivee}(v(c1,a,b)) = 0$$
- (63) label : [c1,q1,r2,n2]

$$\text{val_ic}(q1) / \beta(q1) - v(r2,a,b) / r2 + c1 * \text{derivee}(v(c1,a,b)) = 0$$
- (64) label : [c1,q1,r1,r2,r4,n1,n2]

$$- v(r4,a,b) / (r4*\beta(q1)) + v(r1,a,b) / (r1*\beta(q1)) - v(r2,a,b) / r2$$

$$+ c1 * \text{derivee}(v(c1,a,b)) = 0$$
- (65) label : [c1,q1,r2,n2]

$$- \text{val_ib}(q1) + v(r2,a,b) / r2 - c1 * \text{derivee}(v(c1,a,b)) = 0$$

Nous présentons maintenant quelques cas de pannes.

11.2.3 Résultats

Nous présentons uniquement 2 pannes et le circuit fonctionnant normalement.

Pas de panne :

Tous les modèles sont consistants. Ce résultat est logique.

Résistance r1 très faible :

Les modèles inconsistants sont les suivants :

(64) label : [c1,q1,r1,r2,r4,n1,n2]

$$- v(r4,a,b) / (r4*\beta(q1)) + v(r1,a,b) / (r1*\beta(q1)) - v(r2,a,b) / r2$$

$$+ c1 * \text{derivee}(v(c1,a,b)) = 0$$

(7) label : [n1,q1,r4,r1]
$$- v(r1,a,b) / r1 + v(r4,a,b) / r4 + \text{val_ic}(q1) = 0$$

$$(49) \text{ label : [n5,n4,n3,n2,n1,c1,d1,q2,q1,rc,r5,r4,r3,r2,r1]} \\ - v(rc,a,b) / (rc * beta(q1)) + v(r2,a,b) / r2 + v(r3,a,b) / r3 + v(r5,a,b) / (r5 * beta(q1)) \\ - v(r1,a,b) / (r1 * beta(q1)) + (1 / (beta(q2) * beta(q1)) - 1) * val_ic(q2) = 0$$

$$(57) \text{ label : [c1,d1,q1,q2,rc,n5,r5,n3,r4,r2,n2,r1,n1]} \\ val_ic(q2) / (beta(q2)*beta(q1)) - v(rc,a,b) / (rc*beta(q1)) + v(r5,a,b) / (r5*beta(q1)) \\ + v(r2,a,b) / r2 - v(r1,a,b) / (r1*beta(q1)) + c1 * derivee(v(c1,a,b)) = 0$$

$$(51) \text{ label : [c1,d1,q1,q2,r5,n3,r4,r2,n2,r1,n1]} \\ - i(d1) / beta(q1) + val_ic(q2) / (beta(q2)*beta(q1)) + v(r5,a,b) / r5*beta(q1) \\ + v(r2,a,b) / r2 - v(r1,a,b) / (r1*beta(q1)) + c1 * derivee(v(c1,a,b)) = 0$$

Le conflit minimal est donné par l'équation (7). Etant donné ce conflit, il est inutile de vérifier les autres équations. Toutefois, ceci montre que des ensembles de conflits peuvent être détectés avec des équations assez complexes. Les candidats uniques sont $\{r1\}$, $\{r4\}$, $\{n1\}$ et $\{q1\}$.

Capacité c1 coupée :

Les modèles inconsistants sont¹ :

$$(64) \text{ label : [c1,q1,r1,r2,r4,n1,n2]} \\ - v(r4,a,b) / (r4*beta(q1)) + v(r1,a,b) / (r1*beta(q1)) - v(r2,a,b) / r2 \\ + c1 * derivee(v(c1,a,b)) = 0$$

$$(62) \text{ label : [c1,q2,r3,n4]} \\ val_ic(q2) - v(r3,a,b) / r3 + c1 * derivee(v(c1,a,b)) = 0$$

Un seul candidat unique est détecté : $\{c1\}$!!

Il est difficile de juger de la pertinence de ces résultats avec le faible nombre de tests réalisés. Toutefois, les SBT ont été introduites afin de diagnostiquer les systèmes sans calculer les dérivées. Cette modélisation permet (entre autres choses) de détecter la capacité c1 coupée, panne qui n'est pas trouvée par CATS. Elle permet aussi de détecter l'ensemble de conflits $\{c1, r1, r2, r4, n1, n2\}$ dans le cas de la résistance r1 très faible, ensemble de conflits non trouvé par CATS. L'autre ensemble de conflits est détecté par CATS, ce qui est logique car il ne fait pas intervenir la dérivée d'une grandeur.

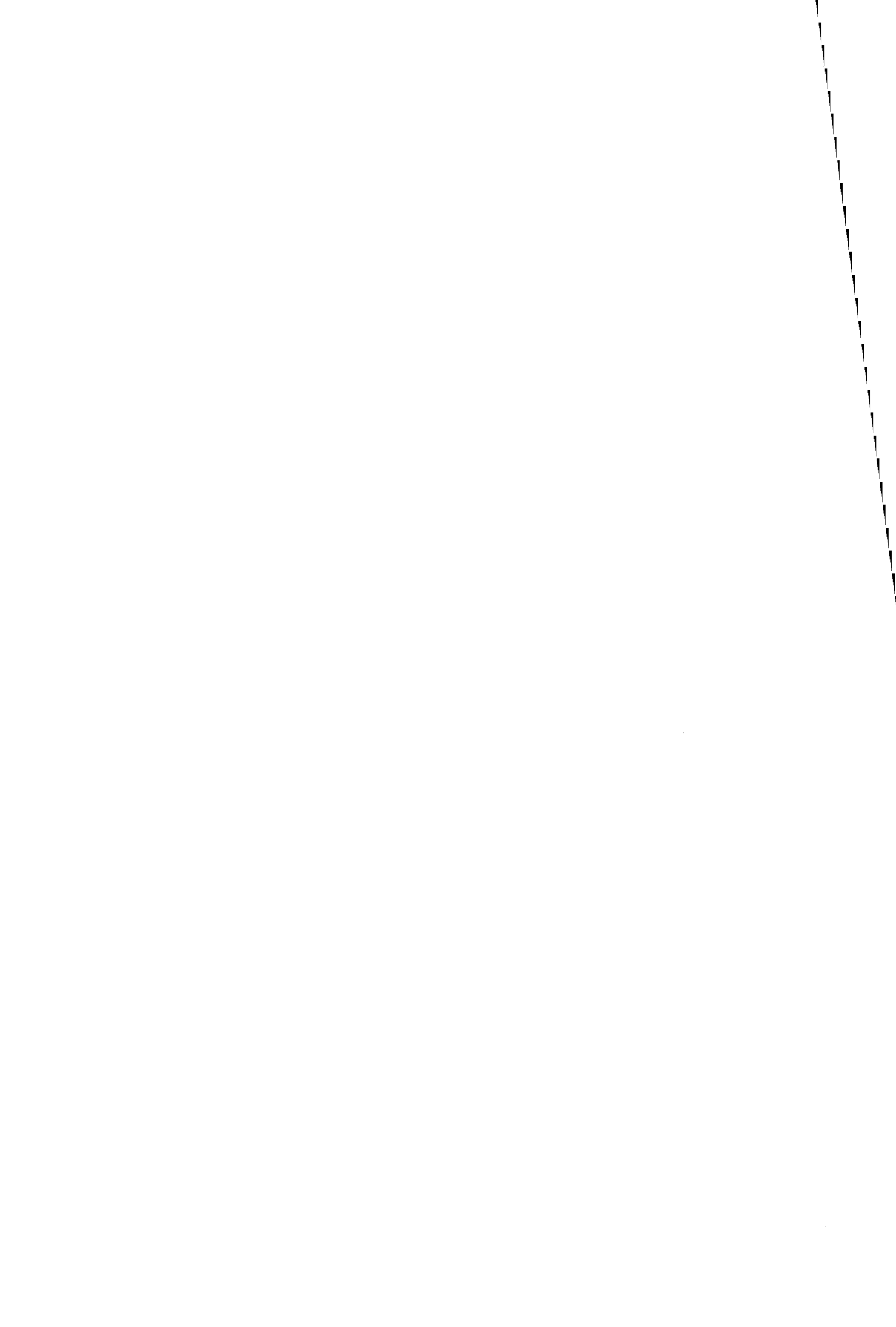
Bien que très partiels, ces résultats ont été jugés satisfaisants et ont abouti au dépôt d'un brevet Dassault Electronique.

1. Le transistor q1 étant saturé, aucun modèle ne permet de définir les courants de ce transistor. C'est pourquoi tous les modèles mentionnant les courants de q1 sont consistants.

Conclusion

Cette partie présente les deux principaux apports de cette thèse : le système de diagnostic et les suites de bandes temporelles. Le premier permet d'obtenir un diagnostic rapide (une vingtaine de secondes pour la compilation et la vérification des modèles) et le deuxième permet de diagnostiquer des systèmes sans calculer la dérivée des grandeurs, ce qui, dans le cas de la capacité coupée, autorise un diagnostic de meilleure qualité. Le principe du système de diagnostic est de remplacer la procédure d'inférence de GDE, composée d'un propagateur et d'un ATMS, par une procédure de compilation des modèles. Etant donné la faible connectivité des processus diagnostiqués et le faible nombre de grandeurs observables vis-à-vis du nombre total de grandeurs, le nombre de modèles générés est faible par rapport au nombre théorique de modèles. Ceci nous assure la viabilité de cette méthode. Pour en connaître la pertinence, il faudrait la comparer directement avec GDE. Or, ceci n'a pas été fait.

CONCLUSION



1 Contributions

La principale contribution de cette thèse est sans aucun doute la modélisation proposée, c'est à dire les suites de bandes temporelles. Cette modélisation permet de faciliter le diagnostic de systèmes analogiques car

- les systèmes peuvent être modélisés par des équations différentielles d'ordre quelconque,
- la vérification de la cohérence d'une équation avec les observations est quantitative et ne nécessite pas le calcul de la dérivée d'une grandeur. Cette vérification est réalisée en utilisant l'opération d'intégration.

L'étude des SBT a permis de montrer que les opérations sur les SBT héritent des propriétés des opérations sur les intervalles et ont certaines propriétés liées à la prise en compte du temps. Ceci a mis en évidence le fait que différencier les grandeurs constantes au cours du temps de celles qui ne le sont pas, peut permettre la détection d'ensembles de conflits. Cette étude a abouti à l'établissement d'une famille d'algorithmes de vérification de la cohérence d'une équation différentielle avec les SBT représentant les grandeurs invoquées dans cette équation.

Le deuxième apport de cette thèse est l'établissement d'une nouvelle procédure de diagnostic basée sur une compilation d'un ensemble d'équations. Son intérêt principal par rapport aux procédures basées sur la simulation telles que G.D.E. [de Kleer et al, 87] réside dans la rapidité d'exécution. Toutefois, le cadre est beaucoup moins ambitieux que celui de G.D.E. : une limite naturelle à la compilation semble être le diagnostic de systèmes modélisés par des équations linéaires (par morceaux). Cette limitation, peu contraignante dans le domaine de l'électronique, peut se révéler plus gênante dans d'autres domaines.

Ces travaux ont aboutis à la création d'un système de diagnostic : DOGS.

2 Perspectives

La validation de DOGS passe par une expérimentation plus importante que celle effectuée dans la quatrième partie. Dans ce cadre, une implémentation de DOGS pour des SBT d'ordre quelconque est actuellement en cours de réalisation par D. Gest de l'I.S.E.N.

Le concept des suites de bandes temporelles a été inventé pour surmonter les limites de la procédure de détection d'incohérence de CATS [Dague et al, 91] qui est basée sur un encadrement de la dérivée des grandeurs. Les résultats obtenus montrent que les algorithmes de DOGS peuvent aussi s'appliquer dans le cadre de CATS. En effet, on remarque que 2 intervalles consécutifs permettent de créer une bande temporelle¹. Il découle de cette constatation que la comparaison d'une grandeur et de sa dérivée peut être réalisée

- soit en calculant un encadrement de la dérivée à partir de l'encadrement de la grandeur (algorithme actuel de CATS),
- soit en calculant un encadrement de la grandeur à partir de l'encadrement de la dérivée (algorithme de DOGS).

1. Ceci revient à considérer qu'un tableau d'intervalles est une suite de bandes temporelles d'ordre 0.

Il est aussi possible de prendre en compte le caractère constant des grandeurs (i.e. les paramètres des composants) dans CATS. Ceci consiste à étendre la procédure de propagation afin de calculer des tableaux d'intervalles représentant les paramètres des composants. L'intervalle encadrant les grandeurs constantes résulte de l'intersection des intervalles du tableau.

L'implémentation de ces deux extensions permettrait de juger la pertinence des suites de bandes temporelles (i.e. des résultats de cette thèse). Sans trop d'effort, CATS peut aussi permettre de comparer les procédures de diagnostic basées sur la compilation ou sur la simulation. En effet, l'algorithme de détection d'inconsistance des modèles est immédiat : il consiste à calculer un tableau d'intervalles encadrant une équation et à vérifier que chaque intervalle contient 0.

Certains aspects des systèmes de diagnostic, tel que le choix de la mesure suivante, n'ont pas été abordés dans cette thèse. La compilation des équations offre de nombreuses possibilités. Il est par exemple possible d'intégrer les approches probabiliste ou heuristique proposées respectivement par de Kleer [de Kleer, 91] et par Dague [Dague et al, 91]. Une autre solution consiste à choisir la grandeur dont la mesure va permettre la vérification du plus grand nombre d'équations. L'approche la plus prometteuse nous semble être l'intégration de notre système dans l'algorithme DIAGNOSE de [Reiter, 87]¹. Le *théorème prover* est constitué des algorithmes de vérification de la consistance des modèles. Un aspect intéressant est que le théorème prover ne réalise jamais deux fois les mêmes opérations puisque les modèles ne sont vérifiés qu'une seule fois. Le choix de la meilleure mesure peut être réalisé en utilisant directement les procédures proposées dans [Hou, 94].

La caractérisation des modèles à générer lors de la compilation n'a pu être achevée. Une telle caractérisation a principalement 3 intérêts. Elle permet :

- de définir la complétude des modèles à générer,
- d'étudier les liens entre les différents modèles générés et notamment de différencier les modèles dont la vérification permet d'assurer la consistance du système de ceux permettant uniquement la détection d'ensemble de conflits plus petits.
- d'étudier les liens entre la connectivité des systèmes et le nombre de modèles générés, c'est-à-dire les liens entre la structure du système et la complexité du diagnostic. Des recherches dans cette direction ont déjà été entreprises, notamment par [Darwiche, 95].

1. Avec les corrections proposées par [Greiner et al, 89]

Références bibliographiques

- [Abu-Hanna, 96] A. Abu-Hanna, "Model-Based Systems and Engineering", *Invited Session on MBS, CESA'96 IMACS-IEEE/SMC Multiconference, "Computational Engineering In Systems Applications", Juillet 9-12, 1996, Lille.*
- [Alarcon et al, 94] I. Alarcon et al, "Heterogenous Integration Architecture for Intelligent Control Systems," *Int. Journal of Artificial Intelligence in Engineering Special Issue, 1994.*
- [Ayeb et al, 90] B. El. Ayeb, P. Marquis, M. Rusinovitch, "Deductive / Abductive Diagnosis: The DA-principles", *Proceedings of ECAI, Stockholm, 1990.*
- [Biasizzo et al, 96] A. Biasizzo, F. Novak, "A Methodology for Model-Based Diagnosis of Analog Circuits", *Invited Session on MBS, CESA'96 IMACS-IEEE/SMC Multiconference, "Computational Engineering In Systems Applications", pp. 344-349, Lille, Juillet 1996.*
- [Böttcher, 95] C. Böttcher, "No Faults in Structure? How to diagnose Hidden Interactions", *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 95), Montréal, pp. 1728 - 1733, Août 1995.*
- [Böttcher et al, 96] C. Böttcher, P. Dague, P. Taillibert, "Hidden interactions in analog circuits", *Seventh International Workshop on Principles of Diagnosis, Octobre 1996.*
- [Brusoni et al, 96] V. Brusoni, L. Console, P. Terenziani, D.T. Dupré, "A spectrum of definitions for temporal model-based diagnosis", *Seventh International Workshop on Principles of Diagnosis, Octobre 1996.*
- [Brown, 76] A. Brown, "Qualitative knowledge, causal reasoning, and the localization of failures", Technical Report 362, MIT Artificial Intelligence Lab, Cambridge, MA, 1976.
- [Brown et al, 82] J.S. Brown, R. Burton, J. de Kleer, "Pedagogical, natural language, and knowledge engineering issues in SOPHIE I, II and III", *In D. Sleeman and J.S. Brown, editors, Intelligent Tutoring Systems, pp. 227-282. Academic Press, New York, 1982.*

- [Cantone et al, 83] R.R. Cantone, F. Pipitone, W. Lander, M. Marrone, "Model-based probabilistic reasoning for electronics troubleshooting", in *Proceedings IJCAI-83*, pp. 207-211, Karlsruhe, Août 1983.
- [Chittaro et al, 96] L. Chittaro, R. Ranon, J.L. Cortés, "Ship over troubled water: functional reasoning with influences applied to the diagnosis of a marine technical system", *Seventh International Workshop on Principles of Diagnosis*, Octobre 1996.
- [Console et al, 90] L. Console, P. Torasso, "Integrating models of correct behaviour into abductive diagnosis", *Proceedings of ECAI 90*, pp. 160-166.
- [Console et al, 91a] L. Console, D.T. Dupré, P. Torasso, "On the relationship between abduction and deduction", *Journal of Logic and Computation*, 1(5), pp. 661-690, 1991.
- [Console et al, 91b] L. Console, P. Torasso, "A spectrum of logical definitions of model-based diagnosis", *Computational Intelligence*, 7(3), 133-141, 1991.
- [Dague et al, 87] P. Dague, P. Devès, O. Raiman, "Troubleshooting: when Modeling is the Trouble," *Proceedings of AAI Conference, Seattle, Juillet 1987*.
- [Dague et al, 90a] P. Dague, P. Devès, P. Luciani, P. Taillibert, "Analog Systems Diagnosis," *Proceedings of the Ninth ECAI, Stockholm, Août 1990*.
- [Dague et al, 90b] P. Dague, O. Jehl, P. Taillibert, "An interval propagation and conflict recognition engine for diagnosing continuous dynamic systems", *International workshop on expert systems in engineering, Vienne, Septembre 1990*.
- [Dague et al, 90c] P. Dague, P. Devès, O. Jehl, P. Luciani, P. Taillibert, "A conflict Recognition Engine for Model-based Diagnosis of Continuous Dynamical Physical Systems", *Proceedings of the 8th Nat. Conf. on Artificial Intelligence (2nd AAI Workshop on Model-based Reasoning), Boston, Juillet 1990*.
- [Dague et al, 91] P. Dague, P. Devès, P. Luciani, O. Jehl, P. Taillibert, "When oscillators stop oscillating," *Proceedings of the twelfth IJCAI, Sydney, Août 1991*.
- [Dague et al, 96] P. Dague, P. Taillibert, "The monostable: a reference problem for analog diagnosis", *Seventh International Workshop on Principles of Diagnosis, Octobre 1996*.
- [Dahlquist et al, 74] G. Dahlquist, A. Björk, "Numerical methods", CWK Llerup, Bokfoerlag, Sweden.
- [Darwiche, 95] A. Darwiche, "Model-based diagnosis using causal network", *Proceedings of the 14th IJCAI, Montréal, pp. 211-217, 1995*.
- [Davis, 84] R. Davis, "Diagnostic reasoning based on structure and behavior", *Artificial Intelligence, vol 24(1)*, pp. 347-410, 1984. Aussi dans [Hamscher et al, 92].
- [DeCoste, 90] D. DeCoste, "Dynamic across-time measurement interpretation", *Proceedings of the 8th Nat. Conf. on Artificial Intelligence*, pp. 373-379, Boston, 1990, aussi dans [Hamscher et al, 92].

- [Dechter, 96] R. Dechter, A. Dechter, "Structure-driven algorithms for truth maintenance", *Artificial Intelligence*, vol. 82, pp. 1-20, 1996.
- [Devès et al, 92] P. Devès, C. Fischer, P. Taillibert, "An Alternative to Expert Systems for Electrical Diagnosis", in *Future Generation Computer Systems*, Vol 7 1991/1992.
- [Dubois et al, 90] D. Dubois, J. Lang, H. Prade, "A Possibilistic Assumption-based Truth Maintenance System with Uncertain Justifications and its Application to Belief Revision Truth Maintenance System", eds J.P.Martin, M.Reinfrank Springer Verlag 1991 (ECAI Stockholm 1990)
- [Dvorak et al, 89] D. Dvorak, B. Kuipers, "Model-based monitoring of dynamic systems", *Proceedings of the 11th IJCAI*, pp. 1238-1243, Detroit, 1989, aussi dans [Hamscher, 92].
- [First, 82] M.B. First, B.J. Weimer, S. McLinden, R.A. Miller, "LOCALIZE: Computer-assisted localization of peripheral nervous system lesions", *Computers and Biomedical Research*, vol. 15(6), pp. 525-543, Decembre 1982.
- [Friedrich et al, 90a] G. Friedrich, W. Nejd, "MOMO- Model-based diagnosis for everybody", *Proceedings of the IEEE Conference on Artificial Intelligence Applications*, Santa Barbara, Mars 1990.
- [Friedrich et al, 90b] G. Friedrich, G. Gottlob, W. Nejd, "Physical impossibility instead of fault models", *Proceedings of the Nat. Conf. on Artificial Intelligence (AAAI 90)*, pp. 331-336, Menlo Park, 1990. Aussi dans [Hamscher et al, 92].
- [Geffner et al, 87] H. Geffner, J. Pearl, "An improved Constraint algorithm for diagnosis", in *Proceedings of the 10th Int. Joint Conf. on Artificial Intelligence*, pp. 1105-1111, Milan, 1987.
- [Genereseth, 84] M.R. Genereseth, "The use of design descriptions in automated diagnosis", *Artificial Intelligence*, vol. 24, pp. 411 - 436, 1984.
- [Goldstone, 91] D.J. Goldstone, "Controlling inequality reasoning in a TMS-based analog diagnosis system", *Proceedings of the 9th Nat. Conf. on AI*, pp. 512-517, Juillet 1991.
- [Greiner et al, 89] R. Greiner, B.A. Smith, R.W. Wilkerson, "A correction to the algorithm in Reiter's theory of diagnosis ", *Artificial Intelligence*, vol 41(1), pp. 79-88, 1989.
- [Guckenbiehl et al, 90] T. Guckenbiehl, G. Schäfer-Richter, "SIDIA: Extending prediction based diagnosis to dynamic models", in *Working Notes of the 1st Int. Workshop on Principles of Diagnosis*, pp. 74-82, Juillet 90.
- [Hamscher et al, 84] W. Hamscher, R. Davis, "Diagnosing circuits with state: An inherently underconstrained problem", *Proceedings of the 4th Nat. Conf. on Artificial Intelligence*, pp. 142 - 147, Austin, 1984.
- [Hamscher, 88] W. Hamscher, "Model-based reasoning: Troubleshooting", in *Exploring Artificial Intelligence: Survey Talks from the National Conferences on Artificial Intelligence*, pp. 297-346, San Mateo, 1988.

Références bibliographiques

- [Hamscher, 89] W. Hamscher, "Temporally Coarse Representation of behaviour for Model-based Troubleshooting of Digital Circuits," *Proceedings of the Eleventh IJCAI, Detroit, Août 1989*.
- [Hamscher, 90a] W. Hamscher, "XDE: Diagnosing devices with hierarchic structure and known component failure modes", in *proceedings of the 6th IEEE Conference on AI applications, pp. 139-143, San Diego, Mars 1990*.
- [Hamscher, 90b] W. Hamscher, "Explaining unexpected financial results", in *Working Notes of the Spring Symposium on Automated Abduction, pp. 96-100, Mars 1990*.
- [Hamscher, 91a] W. Hamscher, "ACP: Reason maintenance and inference control for constraint propagation over intervals", *Proceedings of the 9th Nat. Conf. on AI, pp. 506-511, July 1991*.
- [Hamscher, 91b] W. Hamscher, "Modelind digital circuits for troubleshooting", *Artificial Intelligence, vol. 51, pp. 223-271, 1991*.
- [Hamscher et al, 92] W. Hamscher, L. Console, J. de Kleer, *Readings in Model-Based diagnosis, Morgan Kauffman, San Mateo, CA, 1992*.
- [Holtzblatt, 88] L.J. Holtzblatt, "Diagnosing mutiple failures using knowledge of component states", in *proceedings of 4th IEEE Conference on AI Applications, pp. 139-143, San Diego, Mars 1988*.
- [Hou, 94] A. Hou, "A theory of measurement in diagnosis from first principles", *Artificial Intelligence, vol 65, pp. 281- 328, 1994*.
- [de Kleer, 76] J. de Kleer, "Local methods for localizing faults in electronic circuits", Memo 394, MIT Artificial Intelligence Lab, 1976.
- [de Kleer, 86] J. de Kleer, "An Assumption-based TMS", *Artificial Intelligence 28(2), pp. 127-162, 1986*
- [de Kleer et al, 87] J. de Kleer, B.C. Williams, "Diagnosing Multiple Faults", *Artificial Intelligence, vol 32(1), pp. 97 - 130, 1987*.
- [de Kleer et al, 89] J. de Kleer, B.C. Williams, "Diagnosis with Behavioral Modes", *Proceedings of the Eleventh IJCAI, Detroit, Août 1989*.
- [de Kleer, 90] J. de Kleer, "Using crude Probabilities estimates to guide diagnosis", *Artificial Intelligence, vol. 45, pp. 381- 392, 1990*.
- [de Kleer, 91] J. de Kleer, "Focusing on probable diagnosis", *Proceedings of AAAI Conference, Anaheim, pp. 842-848, 1991*.
- [de Kleer et al, 92a] J. de Kleer, J.S. Brown, "Model-based diagnosis in SOPHIE III", in .
- [de Kleer et al, 92b] J. de Kleer, A.K. Mackworth, R. Reiter, "Characterizing diagnoses and systems", *Artificial Intelligence, 52, pp. 197-222, 1992*.
- [Kuipers, 84] B. Kuipers, "Commonsense Reasoning about Causality: Deriving Behaviour from Structure", *Artificial Intelligence, vol 24, pp. 169-203, 1984*.
- [Kuipers, 86] B. Kuipers, "Qualitative Simulation", *Artificial Intelligence, vol 24,*

- pp. 289-388, 1986.
- [Leyval et al, 91] L. Leyval, S. Gentil, "On line event based simulation through a causal graph", *IMACS workshop on qualitative reasoning and decision support systems*, Toulouse, pp. 209-214, 1991.
- [Lhomme, 94] O. Lhomme, "Contribution à la résolution de contraintes sur les réels par propagation d'intervalles", Thèse de doctorat, Université de Nice - Sophia Antipolis, 1994.
- [Loiez, 95] E. Loiez, "Programme de diagnostic de circuits électroniques", *IVèmes Journées Francophones de Programmation Logique*, Mai 1995.
- [Loiez et al, 96a] E. Loiez, P. Vanheeghe, "Modélisation à l'aide de Bandes Temporelles", Note Interne LAIL, [NI/96/2].
- [Loiez et al, 96b] E. Loiez, P. Taillibert, "Analog Systems Diagnosis: Modeling with Temporal Bands," *Invited Session on MBS, CESA'96 IMACS-IEEE/SMC Multiconference, "Computational Engineering In Systems Applications"*, pp. 552-557, Lille, Juillet 1996.
- [Loiez et al, 96c] E. Loiez, P. Taillibert, "Polynomial temporal band sequences for analog diagnosis", in *Working Notes of the 7th International Workshop on Principles of Diagnosis*, Octobre 1996.
- [Malik et al, 96] A. Malik, P. Struss, "Diagnosis of dynamic systems does not necessarily require simulation", *Seventh International Workshop on Principles of Diagnosis*, Octobre 1996.
- [Moore, 66] R.E. Moore, "Interval Analysis", Prentice Hall, New Jersey, 1966.
- [Mozetic, 91] I. Mozetic, "Hierarchical Model-based diagnosis", *International Journal of Man-Machines Studies*, vol 35(3), pp. 329-362, 1991.
- [Ng, 90] H.T. Ng, "Model-based Multiple Fault Diagnoses of Time Varying, Continuous Physical Devices," *Proceedings of the sixth IEEE Conference on AI Applications*, Santa Barbara, Mars 1990.
- [Oyeleye et al, 90] O.O. Oyeleye, F.E. Finch, M.A. Kramer, "Qualitative modeling and fault diagnosis of dynamic processes by MIDAS", *Chemical Engineering Communications*, 96:205-228, 1990.
- [Pan, 84] Y. Pan, "Qualitative reasoning with deep-level mechanism models for diagnoses of mechanism failures," *Proceedings of the first IEEE Conference on AI Applications*, Denver, 1984.
- [Bos-Plachez, 96] C. Bos-Plachez, "A Possibilistic ATMS Contribution To Diagnose Analog Electronic Circuits", *Proceedings of the 4th European Congress on Intelligent Techniques and Soft Computing*, Aachen, pp. 2:612-617, Septembre 1996.
- [Poole, 88] D. Poole, "Representing knowledge for logic-based diagnosis", *Proceedings of the International Conference on Fifth Generation Computer Systems*, pp. 1282-1290, ICOT, 1988.

- [Poole, 90] D. Poole, "Normality and Faults in logic-based diagnosis", *Proceedings of the 11th IJCAI*, pp. 1304-1310, Detroit, Août 1989.
- [Poole, 91] D. Poole, "Representing diagnostic knowledge for probabilistic Horn abduction", in *Proceedings of the 12th Int. Joint Conf. on Artificial Intelligence*, pp. 1129-1135, Sydney, Août 1991.
- [Raiman, 86] O. Raiman, "Order of magnitude reasoning", *Proceedings of the 5th Nat. Conf. on Artificial Intelligence*, pp. 100-104, Philadelphia, 1986.
- [Raiman, 89] O. Raiman, "Diagnosis as a trial: the alibi principle", *IBM Workshop on Model-Based Diagnosis*, Paris, Juillet 1989.
- [Reiter, 87] R. Reiter, "A Theory of Diagnosis from first Principles," *Artificial Intelligence* 32, 1987.
- [Reggia et al, 83] J.A. Reggia, D.S. Nau, Y. Wang, "Diagnostic expert systems based on a set-covering model", *International Journal of Man-Machines Studies*, vol 19, pp. 437-460, 1983.
- [Scarl et al, 85] E.A. Scarl, J.R. Jamieson, C. I. Delaune, "A fault detection and isolation method applied to liquid oxygen loading for the space shuttle", in *Proceedings of the 9th Int. Joint Conf. on Artificial Intelligence*, pp. 414-416, Los Angeles, 1985.
- [Shortliffe, 76] E.H. Shortliffe, "Computer-based Medical Consultations: MYCIN". New York, Elsevier, 1976.
- [Singh, 86] N. Singh, "SATURN: An automatic test generation system for digital circuits", *Proceedings of the 5th Nat. Conf. on Artificial Intelligence*, pp. 778-783, Philadelphia, 1986.
- [St Marc, 96] T. St Marc, "Rapport de stage de fin d'étude: Diagnostic Automatique", Document Dassault Electronique, Ne 768603, St Cloud, Juillet 1996.
- [Struss, 89] P. Struss, "Diagnosis as a process", *International Workshop on Model-Based Diagnosis*, Paris, July 1989.
- [Struss, 91] P. Struss, "What's in SD? Towards a theory of modelling for diagnosis", *Working Notes of the 2nd Int. Workshop on Principles of Diagnosis*, pp. 41-51, 1991.
- [Struss et al, 89] P. Struss, O. Dressler, "Physical Negation - Integrating Fault Models into the General Diagnostic Engine", *Proceedings of the Eleventh IJCAI*, Detroit, pp. 1318- 1323, Août 1989
- [Tatar, 96] M. Tatar, "Diagnosing with Cascading Defects", *Proceedings of the 12th European Conference on Artificial Intelligence (ECAI - 96)*. Budapest, pp. 511 - 515, Août 1996.
- [ten Teije et al, 96] A. ten Teije, F. van Harmelen, "Examples of approximations in diagnosis based on approximate entailment", *Seventh International Workshop on Principles of Diagnosis*, Octobre 1996.

[Williams, 86]

B.C. Williams, "Doing Time: Putting Qualitative Reasoning on Firmer Ground," *Proceedings of AAAI Conference, Philadelphia, Août 1986*.

