

Table des matières

1 Contexte industriel	7
1.1 Présentation de l'industrie des semiconducteurs	7
1.2 Fabrication des semiconducteurs	9
1.3 Planification de la production dans une entreprise de semiconducteurs	12
1.3.1 Les différentes échelles de planification	12
1.3.2 Complexité de la planification de production	13
1.3.3 Planification de la production dans l'entreprise STMicroelectronics	14
1.4 Contrôle de production	14
1.4.1 Principe du contrôle de production dans la littérature	15
1.4.2 Principe du contrôle de production dans l'usine de Crolles	16
1.5 Synthèse du chapitre	16
2 Retard de la production de l'usine par rapport à la planification	19
2.1 Performance de l'outil de planification	19
2.1.1 Présentation du l'outil de planification	19
2.1.2 Les données disponibles dans le cadre de nos travaux	22
2.1.3 Utilisation de l'outil de planification selon différents paramètres	22
2.2 Caractérisation de la variabilité de la production	25
2.2.1 Définition de la variabilité de la production dans l'usine	25
2.2.2 Analyse de la variabilité dans l'usine Crolles 300	26
2.3 Détection des retards entre la production et la planification	29
2.3.1 Comparaison cumulative entre la planification et l'historique	30
2.3.2 Indicateur du retard critique entre la planification et l'historique	32
2.4 Synthèse du chapitre	37
3 État de l'art du diagnostic dans les chaînes de production	39
3.1 Définition général du problème de diagnostic	39
3.1.1 Méthodes basées sur la connaissance	39
3.1.2 Méthodes basées sur le traitement des données	41
3.1.3 Méthodes à base de modèle	43
3.2 Diagnostic des chaînes de production	48
3.2.1 Diagnostic à base de connaissance	48
3.2.2 Diagnostic à base de données	49
3.2.3 Diagnostic à base de modèle	50
3.3 Synthèse du chapitre	51
4 Modélisation du problème	53
4.1 Détection de dérives temporelles dans les Graphes d'Événements Temporisés	54
4.1.1 Graphes d'Événements Temporisés (GET)	54
4.1.2 Détection de dérives temporelles dans un GET	56
4.2 Modélisation d'une étape de fabrication par des GET	57
4.2.1 Formalisation du procédé de fabrication	57
4.2.2 Focus sur une étape d'une opération	58
4.2.3 Modélisation d'une étape par des GET	59
4.3 Synthèse du chapitre	62

5 Outils algébriques	65
5.1 Rappels mathématiques	65
5.2 Théorie des dioïdes	66
5.2.1 Dioïdes	67
5.2.2 Dioïde de séries formelles et dioïde quotient	70
5.2.3 Définition et propriétés de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$	71
5.3 Théorie de la résiduation appliquée aux dioïdes	75
5.3.1 Théorie de la résiduation	75
5.3.2 Résolution d'application par la théorie de la résiduation	77
5.4 Modélisation des systèmes $(\max, +)$ linéaires	78
5.5 Mesure des décalages temporels entre les séries $\mathcal{M}_{in}^{ax}[\gamma, \delta]$	80
5.6 Synthèse du chapitre	82
6 Détection et localisation de dérives temporelles par des observateurs	83
6.1 Synthèse d'un observateur de systèmes $(\max, +)$ -linéaires avec perturbations	84
6.1.1 Système avec perturbations	84
6.1.2 Observateur	85
6.2 Détection de dérives temporelles	90
6.2.1 Caractérisation des dérives temporelles par des perturbations	90
6.2.2 Indicateur	91
6.2.3 Raffinement de l'état estimé	93
6.2.4 Indicateur avec les états estimés SAP	96
6.3 Localisation de dérives temporelles	98
6.3.1 Tandem	98
6.3.2 Parallélisme	99
6.3.3 Synchronisation	100
6.3.4 Boucle sur une même transition	103
6.3.5 Boucle entre deux transitions	104
6.3.6 Exemples d'associations de structures élémentaires	106
6.4 Synthèse du chapitre	111
7 Détection de dérives temporelles dans des systèmes intervalles	113
7.1 Théorie des dioïdes intervalles	113
7.2 Modélisation des systèmes $(\max, +)$ -linéaires à temps intervalles	115
7.3 Etude des sorties d'un système $(\max, +)$ -linéaire à temps intervalle	116
7.3.1 Mesure de la distance temporelle du système	117
7.3.2 Sorties acceptables d'un système	118
7.4 Détection de dérives temporelles dans un système $(\max, +)$ -linéaire à temps intervalles	120
7.4.1 Indicateur dans les cas d'une sortie certaine	121
7.4.2 Indicateur dans les cas d'une sortie incertaine	123
7.5 Synthèse du chapitre	126
8 Exemple de diagnostic de dérives temporelles pour une opération de l'usine	127
8.1 Présentation générale de l'exemple étudié	127
8.2 Transformation des données de l'usine en séries de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$	128
8.2.1 Transformation des données de l'historique en entrée u	128
8.2.2 Transformation des données de l'historique en sortie y_o	129
8.3 Modélisation de l'opération	130
8.3.1 Modèles GET de l'opération	130
8.3.2 Vérification de la modélisation	131
8.3.3 Calcul des sorties y_s des GET de l'opération avec l'entrée u	132
8.4 Diagnostic de dérives temporelles	133
8.4.1 Par intervalles	133
8.4.2 Par observateur	135
8.5 Synthèse du chapitre	138
Bibliographie	145

Introduction

Les entreprises de semiconducteurs sont en fort essor depuis quelques années, elles produisent en effet de plus en plus de puces électroniques nécessaires à nos appareils du quotidien comme les ordinateurs, les automobiles ou bien les téléphones ... Dans le cadre des entreprises de semiconducteurs la productivité est toujours plus importante pour pouvoir optimiser la production dans les usines. Elles ont un coût de production élevé car ces usines possèdent des équipements onéreux dont l'utilisation doit être optimisée.

Les usines de semiconducteurs font partie des chaînes de production les plus complexes au monde. Avec une concurrence toujours plus importante, la production de semiconducteurs à moindre coût, en grande quantité, mais sans en détériorer la qualité, est un enjeu majeur pour ces entreprises. L'optimisation des flux de production devient un élément central dans ces usines. Une bonne gestion des flux permet par exemple de limiter l'impact économique des goulots d'étranglement et d'optimiser l'utilisation de tous les équipements d'une usine. Dans ces entreprises, la planification des chaînes de production est complexe, elle s'effectue généralement à l'aide d'un logiciel qui permet de faire un plan global de production par semaine, procurant un ensemble de consignes permettant de respecter l'avancement de fabrication des produits dans l'usine.

Pour une bonne gestion des flux de production, il faut comparer les flux réels de l'usine aux plans issus de la planification et veiller à ce qu'ils se correspondent. Si ces flux se contredisent cela peut engendrer des problèmes dans la production des semiconducteurs. Pour faire cette comparaison plusieurs méthodes sont mises en place en observant les consignes de production sur des ateliers ou sur certaines technologies. Néanmoins ces observations sont canalisées sur ces ateliers et technologies spécifiques sans proposer une comparaison globale des différents plans de planification à l'échelle de l'usine.

L'idée dans ces travaux est d'explorer une méthode pour effectuer une comparaison complète des plans de planification avec la réalité de l'usine. La problématique se pose dans un cadre où les plans de planification de l'usine sont au niveau des problèmes de la planification opérationnelle. Ces travaux de thèse s'effectue dans le cadre d'une convention CIFRE entre la société STMicroelectronics (site de Crolles) et le LAAS (Laboratoire d'Architecture et d'Analyse de Systèmes, Toulouse).

Le chapitre 1 présente le contexte économique et industriel des entreprises de semiconducteurs. Pour cela, on décrit le processus de fabrication des puces électroniques. Puis, les grandes lignes de fonctionnement de la planification de production et celles mises en place au sein l'entreprise STMicroelectronics à Crolles (38) sont présentées. Enfin, on décrit le fonctionnement du contrôle de production qui vérifie au cours de la fabrication des semiconducteurs qu'il n'y a pas de problème dans la production. Le chapitre se conclut sur la présentation du contrôle de production et sur la problématique de détection de dérive temporelle qui est étudiée dans le cadre de ces travaux.

Une présentation des différentes outils de planification et des données de l'historique utilisés dans l'entreprise est proposée dans le chapitre 2. Ce chapitre présente aussi les différentes utilisations du logiciel de planification utilisé dans l'entreprise et comment elles sont mises en place dans celle-ci. Ce chapitre cherche à montrer que les données de l'entreprise sont sujettes aux variations du fonctionnement de l'usine. Enfin, le chapitre présente une méthode utilisant les données de la planification et de l'historique pour effectuer une première étude comparative qui nous permet d'identifier un point limitant dans l'étude de notre problématique sur les dérives temporelles.

Une revue des différentes méthodes de diagnostic est présentée dans le chapitre 3. Après avoir donné les différentes méthodes de diagnostic existant dans la littérature, un approfondissement de ces méthodes de diagnostic sur les chaînes de production est présenté. Parmi les méthodes de diagnostic des chaînes de production, on se concentre sur les méthodes de diagnostic à base de modèles pour des systèmes à événements discrets temporels.

La problématique étudiée et le positionnement de notre méthode de diagnostic sont décrits dans le chapitre 4. Dans cette thèse, nous proposons l'utilisation de systèmes (max, +)-linéaires pour définir et mettre en œuvre un ensemble de méthodes de détection et de localisation de dérives temporelles dans de tels systèmes. Les deux méthodes que nous proposons reposent sur le principe général (voir figure 1) qui consiste à proposer un moyen de comparer un flux attendu (plan issu de la planification) et le flux réel de production dans l'usine et d'en exploiter les différences pour déterminer les sources de ces dérives.

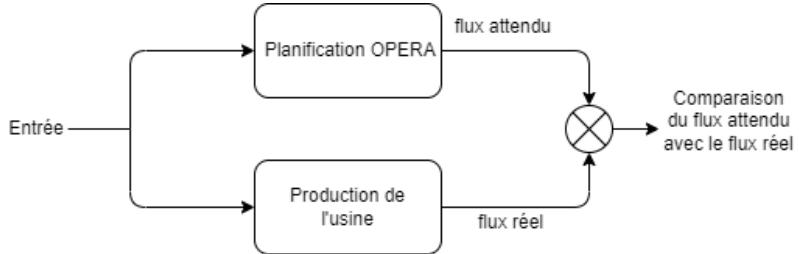


FIGURE 1 – Méthode de diagnostic : principe général

Dans ce chapitre on définit formellement le type de modèle que l'on veut exploiter, à savoir les Graphes d'Événements Temporisés (GET), la notion de dérive temporelle sur ces GETs et la façon de modéliser les informations observables du système. La deuxième partie de ce chapitre est consacrée quant à elle à la modélisation générique du cas d'étude qui a été identifié au cours de ces travaux de thèse.

Les méthodes développées s'appuient sur l'algèbre (max, +) qui est présenté dans le chapitre 5. Ce chapitre présente les définitions mathématiques utilisées notamment la théorie des dioïdes et en particulier le dioïde $\mathcal{M}_{in}^{ax}[\gamma, \delta]$. Le chapitre propose une traduction algébrique de notre modélisation en représentation d'états à l'aide du dioïde $\mathcal{M}_{in}^{ax}[\gamma, \delta]$. Les opérations mathématiques servant à comparer les flux sont alors présentées, ces opérations exploitent la représentation de ces flux dans le dioïde $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ et s'appuient sur la théorie de la résiduation.

À l'aide de l'algèbre (max, +) et des opérations définies précédemment on définit une première méthode de détection et de localisation des dérives temporelles dans le chapitre 6. Cette méthode définit une modélisation d'un système avec observateur qui permet d'estimer ses états. Dans cette méthode, une dérive temporelle est caractérisée sous la forme d'une perturbation. Puis, une extension de cette méthode est présentée dont l'objectif est d'améliorer la détection des dérives temporelles et faire une meilleure analyse des résultats. Enfin, l'analyse de ces états permet de réaliser une localisation des dérives temporelles.

Le chapitre 7 présente une deuxième méthode de détection mais qui s'applique sur des systèmes à temps intervallaire. Cette méthode s'appuie sur des dioïdes intervalles et définit deux indicateurs, l'un pour des sorties uniques et l'autre pour des sorties qui sont aussi des intervalles.

En s'appuyant sur les données réelles mises à notre disposition au sein de l'entreprise dans le cadre de cette thèse CIFRE, le chapitre 8 présente un exemple utilisant la méthode de détection des dérives temporelles par intervalles et la méthode par observateur sur le cas d'étude identifié au chapitre 4. Ce chapitre montre notamment comment traduire des données réelles de l'historique de production en séries algébriques issues du dioïde $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ et propose une analyse des résultats proposés par ces approches.

Tous les calculs des exemples des chapitres 6, 7 et 8 ont été effectués grâce à la bibliothèque MAXPLUSDIAG implémentée en C++ au sein du LAAS qui repose sur la bibliothèque MINMAXGD également écrite en C++ (LARIS, Université d'Angers, [CLHB00]) et qui rend accessible toutes les opérations formelles définies dans le dioïde $\mathcal{M}_{in}^{ax}[\gamma, \delta]$.

Chapitre 1

Contexte industriel

1.1 Présentation de l'industrie des semiconducteurs

L'industrie du semiconducteur fabrique des circuits intégrés communément appelés puces. Les puces sont des dispositifs électroniques produits pour différents domaines d'application, comme l'automobile, les télécommunications, la médecine et de nombreux autres domaines. Les puces sont disponibles dans de nombreuses applications comme les ordinateurs, les capteurs, l'automobile ou les téléphones portables. Du fait de la perpétuelle évolution de ses domaines d'application l'industrie du semiconducteur est elle-même en perpétuelle évolution. Un semiconducteur est un matériau qui a des propriétés électriques entre le conducteur et l'isolant. Le fonctionnement d'une puce est basé sur l'assemblage de plusieurs composants électroniques simples (diode, transistor, résistance, etc.). Selon l'association de ces composants électroniques simples, le fonctionnement sera plus ou moins complexe. Les puces sont généralement encapsulées dans des boîtiers pour être ensuite intégrées dans des cartes électroniques. L'évolution constante des technologies est une des caractéristiques principales de cette industrie comme le décrit la loi de Gordon Moore (1965) présentée dans l'article [Moo06] et qui montrait alors que la densité d'intégration des dispositifs devrait doubler environ tous les deux ans. Bien qu'elle ait souvent été remise en question, cette loi empirique s'est vérifiée au cours du temps avec des puces de plus en plus complexes puisque certaines intègrent aujourd'hui plusieurs milliards de transistors élémentaires.

La complexité des technologies de production et la sophistication des équipements de production requièrent des investissements extrêmement importants. La production des puces est réalisée dans les salles blanches (environnement très contrôlé avec des conditions hygrométriques, de température et de qualité d'air très surveillées). Comme le montre la figure 1.1 l'objectif de la salle blanche est le maintien de l'air propre en injectant un flux d'air unidirectionnel de manière à continuellement injecter de l'air filtré par le plafond pour évacuer par le sol toute particule qui serait générée à l'intérieur de la salle, suite aux activités humaines ou mécaniques. La précision des différents contrôles, la qualité des matériaux mis en œuvre et l'énergie nécessaire à son fonctionnement font que les coûts d'installation et d'exploitation d'une salle blanche sont très importants.

Du fait de leur complexité technique, les équipements de production représentent une part essentielle des coûts de production. Suivant sa fonction et son type, un équipement peut coûter entre quelques centaines de milliers (équipements de mesure, robots de transports) et plusieurs dizaines de millions de dollars (équipements de lithographie). Dans une salle telle que celles de Crolles, on trouve plusieurs centaines d'équipements et 70% du coût de l'installation leur est consacré d'après [MFM12]. Une salle blanche va rapidement coûter plusieurs milliards de dollars. Face à de tels investissements, les différentes ressources doivent être utilisées au maximum pour maximiser l'amortissement des installations et minimiser les coûts d'utilisation.

Pour pouvoir suivre la loi de Moore, la taille standard des plaques de silicium a augmenté au fil du temps pour garantir une meilleure homogénéité des phénomènes physiques (en diminuant l'importance des « bords » de plaque) et permettre la miniaturisation des puces. Cette croissance de la taille des plaques permettait jusqu'ici de réduire le coût individuel de chaque puce (plus de surface = plus de puces). Cette croissance de la taille standard s'est cependant arrêtée il y a une dizaine d'années du fait des coûts prohibitifs envisagés pour les usines, et donc du faible nombre d'industriels intéressés. À l'heure actuelle la réduction des coûts de production doit donc s'appuyer essentiellement sur la réduction des coûts d'exploitation à travers le développement de meilleurs

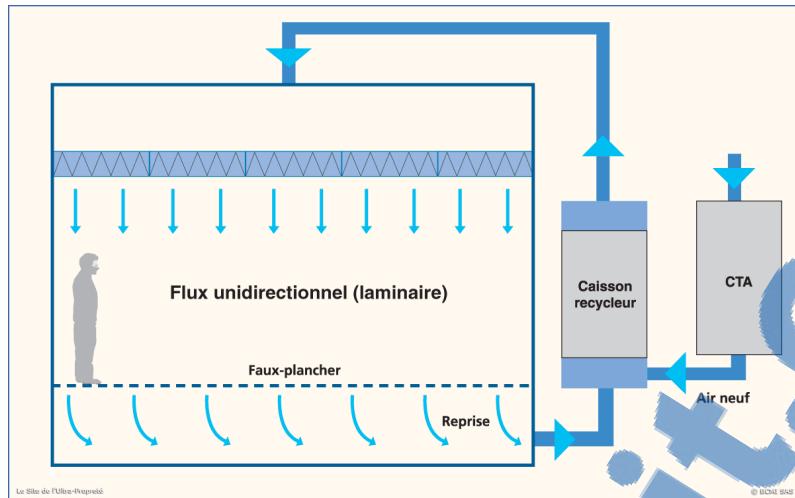


FIGURE 1.1 – Fonction d'une salle blanche

systèmes décisionnels.

Du fait du rôle central des composants électroniques dans la plupart des domaines industriels, le chiffre d'affaires global de l'industrie augmente de façon permanente comme le montre la figure 1.2 issue des données de la Semiconductor Industry Association (SIA [SIA18]). Cette croissance



FIGURE 1.2 – Chiffre d'affaires global des ventes de semiconducteurs dans le monde en billions de dollars tiré de [SIA18] entre 1999 et 2019

importante du chiffre d'affaires dans le monde se fait toutefois de façon différenciée et la part de ce chiffre réalisée par les 10 premiers acteurs mondiaux est de plus en plus importante du fait de l'importance des investissements demandés. Pour rester compétitifs face aux géants américains et asiatiques, les fabricants européens ont donc dû diversifier leur production et améliorer la flexibilité de leurs lignes de fabrication pour rester compétitifs et surtout rentables.

STMicroelectronics est une entreprise de semiconducteurs franco-italienne qui conçoit, développe, fabrique et commercialise des puces. Le groupe a 11 sites de fabrication dans le monde et des centres de recherche et de développement situés dans 10 pays. Les principales usines de fabrication de puces se trouvent à Agrate Brianza et à Catane en Italie, à Crolles, Rousset et Tours en France et à Singapour. Les sites d'assemblage et de tests des puces sont situés en Chine, en Malaisie, à Malte, au Maroc, aux Philippines et à Singapour.

Le site de Crolles intègre deux unités de production de composants : l'une pour fabriquer des puces sur des plaques de 200mm de diamètre (Crolles 200) et l'autre sur des plaques de 300mm (Crolles 300). Crolles 300 est à ce jour l'usine la plus avancée de l'entreprise. Elle est très fortement automatisée : toutes les activités de transport et de stockage des produits à l'intérieur de l'usine sont réalisées par un système automatique (Automated Material Handling System / AMHS) au plafond.

1.2 Fabrication des semiconducteurs

La figure 1.3 présente le processus de fabrication des semiconducteurs. Elle s'effectue généralement de la manière suivante :

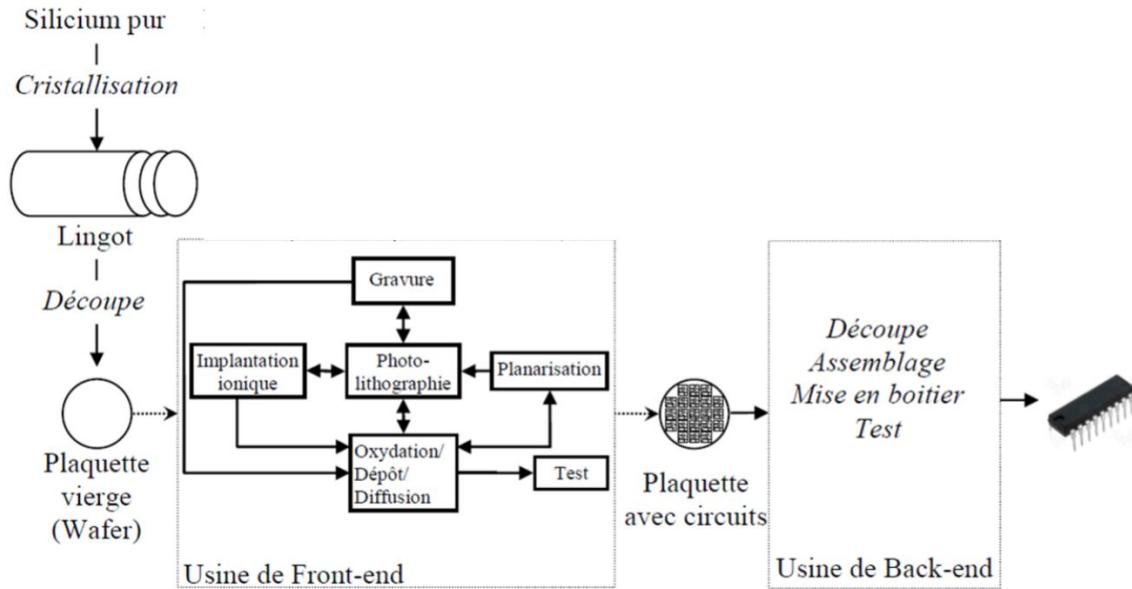


FIGURE 1.3 – Processus de fabrication de semiconducteurs [Bet12]

ment sur une plaque de silicium monocristallin de quelques centaines de microns d'épaisseur. Le diamètre des plaques dépend généralement de l'époque de construction de l'usine et les équipements de production ne peuvent traiter qu'un seul diamètre de plaques. La plupart des usines de composants sont aujourd'hui des unités 200mm et 300mm. Sur une plaque, on va venir « imprimer » des centaines à des milliers de puces, le nombre variant évidemment selon les dimensions de la puce et le diamètre de la plaque. Les puces sont réalisées en imprimant et en connectant divers composants élémentaires : résistances, isolants et principalement transistors. Les différents composants élémentaires sont fabriqués par couches qui sont successivement déposées, isolées, gravées, recuites, polies sur les plaques. Ces différentes couches sont produites en salle blanche, sous atmosphère contrôlée, dans les usines *front-end* appelées fab. À la sortie de ces usines *front-end* les plaques de silicium sont envoyées vers les sites *back-end* où les puces sont testées individuellement afin de détecter les zones non fonctionnelles. Une fois ces tests passés, les plaques sont découpées en puces et envoyées dans les usines d'assemblage où elles seront mises en boîtier avant d'être de nouveau testées pour validation finale.

La thèse se concentre sur la partie *front-end* de la fabrication des puces qui est la partie la plus complexe du processus. La suite de cette section décrit plus en détail le fonctionnement du *front-end*.

Le dépôt d'une seule particule de quelques microns pouvant endommager un grand nombre de transistors et donc le fonctionnement de plusieurs puces, la fabrication s'effectue dans des salles blanches dont la surface avoisine les 10000 m^2 .

Comme indiqué précédemment, les puces sont fabriquées couche par couche. On pourrait presque parler ici de fabrication additive. Chaque étape de fabrication s'effectue dans une zone spécifique caractérisée par le type d'équipement employé et le type de processus réalisé. Les différentes zones sont appelées ateliers. Les plaques sont régulièrement mesurées et inspectées par échantillonnage (c'est-à-dire, on ne mesure qu'un certain pourcentage de façon non destructive) pour vérifier la qualité des puces et le bon fonctionnement des équipements. Si un défaut ou une marginalité est constaté, la plaque est retraitée si c'est possible, sinon elle est détruite. Le nombre d'étapes élémentaires nécessaires à la fabrication d'un circuit allant, pour Crolles 300 de quelques centaines à plus de mille, le même type d'équipements va être utilisé plusieurs fois dans la gamme de fabrication. On va donc avoir à gérer des flux réentrants : c'est sans doute l'une des caractéristiques principales de l'industrie en termes de gestion de production. La figure 1.3 montre dans l'usine de front-end les relations entre les 5 différents ateliers qui sont définis ci-dessous :

- Diffusion/Oxydation : comme son nom l'indique, la diffusion est un processus qui va utiliser la

température pour permettre la migration d'un matériau à l'intérieur de la plaque ou encore d'homogénéiser ou de relaxer des contraintes à l'intérieur de couches. L'oxydation est un processus de dépôt de matériau qui permet la croissance d'une couche d'oxyde (de silicium à l'origine) à la surface de la plaque. Diffusion et oxydation sont des processus à haute température qui utilisent traditionnellement des fours avec des durées de process pouvant aller jusqu'à 12 heures et plus. La plupart des fours sont des équipements à batch c'est-à-dire qu'un four peut traiter plusieurs lots de plaques en même temps.

- Photolithographie : la photolithographie est un processus qui consiste à transférer des motifs sur une résine photosensible déposée sur la surface de la plaque grâce à une source lumineuse. Comme avec un pochoir, les motifs sont présents sur un masque caractéristique du produit à obtenir et de l'étape de fabrication considérée. Les équipements de photolithographie sont les plus critiques en termes de technologie et aussi les plus complexes. Certains équipements peuvent coûter plus de 50 millions d'euros. C'est pour cette raison que c'est généralement cet atelier qui va définir la capacité globale de l'usine, puisque c'est en photolithographie qu'on trouvera les principaux goulots d'étranglement de la ligne.
- Gravure : la gravure est un processus d'élimination de matière. Généralement le processus de gravure se fait après le processus de photolithographie. Il existe deux types de process : la gravure humide qui utilise une réaction principalement chimique (la plaque est plongée dans différents bains ou aspergée de différentes solutions chimiques) et la gravure sèche qui utilise le bombardement de la plaque par différents types d'ions obtenus à partir de plasmas gazeux.
- Implantation : l'implantation ionique est un processus d'injection d'ions dopants à la surface de la plaque et à l'intérieur du silicium. L'implantation des dopants (de type P ou de type N) va permettre de modifier les propriétés conductrices des zones concernées et de créer les composants élémentaires des circuits.
- Planarisation : la planarisation est un processus de polissage mécano-chimique de la plaque. C'est un processus d'aplanissement de la surface de la plaque pour réduire les écarts et les irrégularités d'épaisseur liés au cumul de différents dépôts. Ce processus est par exemple effectué au début de chaque nouvelle couche de métallisation. Cela permet d'éviter l'accumulation des erreurs topologiques sur une plaque.



FIGURE 1.4 – Image d'un FOUP contenant 25 plaques

Le transport des plaques entre les différents équipements s'effectue avec des boîtes appelées FOUP (Font Opening Unified Pod), un FOUP contient 25 plaques (voir la Figure 1.4). Traditionnellement, les plaques sont regroupées par 25, on parle de lot. Un lot correspond à un groupe de 25 plaques du même produit final. Étant donné la diversité des phénomènes physico-chimiques mis en jeu, on va trouver plusieurs types de fonctionnements différents pour les équipements de production. Certains équipements peuvent traiter plusieurs lots en même temps (fonctionnement par *batch*) alors que d'autres traitent unitairement chaque plaque. On parlera alors de fonctionnement plaque à plaque. De même, certains types d'équipement requièrent des phases de stabilisation entre recettes différentes. On parlera alors de trains ou de séries.

Pour la fabrication de semiconducteurs les termes peuvent changer selon l'usine et la compagnie. De même, le vocabulaire peut être différent entre l'industrie et la littérature. Dans cette thèse les définitions spécifiques sont introduites dans la suite de la section.

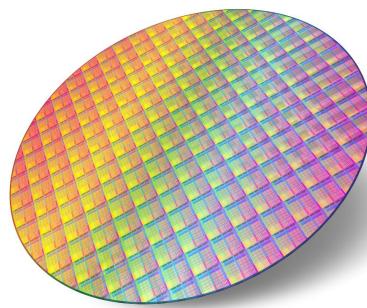


FIGURE 1.5 – Image d'une plaque ou wafer

- Plaque, plaquette, wafer : une tranche, une galette de semiconducteur, généralement en silicium. La plaque est utilisée comme support pour la fabrication des puces. Dans l'usine généralement les plaques ne sont pas vierges, elles ont déjà passé certaines étapes de fabrication. Voir la figure 1.5 pour un exemple de plaque.
- Lot : Un lot est un ensemble de plaques. Il peut contenir d'une plaque à 25 plaques. Les plaques contenues dans un lot sont physiquement transportées dans un FOUP. Dans le système d'information de Crolles 300, le lot est la référence, c'est l'objet logique et physique qui se déplace dans l'usine.
- Produit : Un produit spécifie la puce fabriquée. Le produit correspond à une référence client et est réalisé via un enchaînement particulier de couches utilisant un jeu de masques caractéristique du produit. Chaque produit correspond donc à une route – enchaînement d'étapes – et à un jeu de masques particuliers.
- Technologie : Une technologie correspond à un type générique de procédés de fabrication partageant des caractéristiques communes telles que la finesse de gravure (dimension minimale imprimée sur la plaque), le type de transistor (bipolaire, CMOS) ou de composants (capteurs d'image, mémoire, etc.)..
- Étape (step) : Une étape est une étape élémentaire de fabrication et correspond au passage d'un lot sur un équipement. On distingue généralement les étapes à valeur ajoutée (étapes dites de process, impliquant une transformation physique de la plaque) des étapes sans valeur ajoutée (tels que les nettoyages ou la mesure des plaques).
- Opération : Une opération est un regroupement de plusieurs étapes. Une opération est généralement composée d'une étape à valeur ajoutée précédée et/ou suivie par des étapes de nettoyage et/ou de mesure.
- Brique : Une brique est un terme lié au système de gestion informatique de la production (MES pour *Manufacturing Execution System*) et correspond au regroupement de plusieurs opérations qui vont constituer un ensemble logique.
- Route : Une route désigne la liste ordonnée des étapes de fabrication d'un lot (gamme de fabrication). Le lot doit passer sur chaque étape de fabrication pour pouvoir sortir de l'usine. Il existe de nombreuses routes possibles pour une technologie donnée et une route peut avoir de nombreux produits. Une route peut comprendre de 100 à 1000 étapes élémentaires suivant la complexité de la technologie et les options auxquelles elle correspond.
- Équipement : Un équipement est une machine de traitement qui permet de faire l'étape du processus de fabrication.
- Recette : Une recette est l'ensemble des paramètres qui permettent de piloter un équipement pour la réalisation d'une étape. La recette donne le type de fluides à utiliser, leur débit, la pression de chaque gaz, la durée de traitement ou la température. La recette est liée à l'étape de l'opération de la brique d'une route.

Pour mieux comprendre les relations entre les éléments cités précédemment, la figure 1.6 montre les liens existant entre les technologies, les produits et les lots. Une technologie supporte p produits, un produit correspond à un nombre de n lots. Les produits d'une même technologie vont être réalisés en enchaînant les mêmes étapes mais avec des masques différents

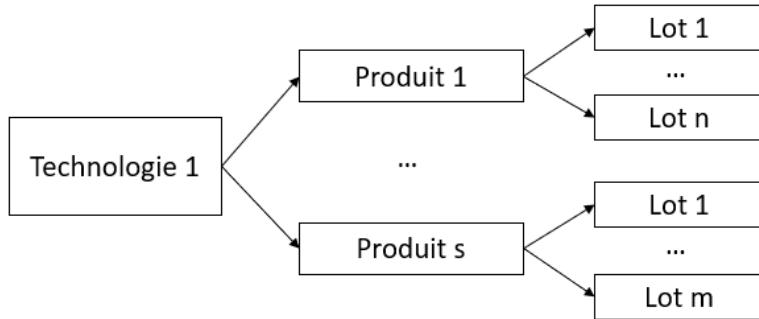


FIGURE 1.6 – Hiérarchie de la classification des produits

La figure 1.7 explique l'avancement d'un lot sur une route. Une route est composée de n briques, chaque brique est composée de m opérations et chaque opération est composée de p étapes.

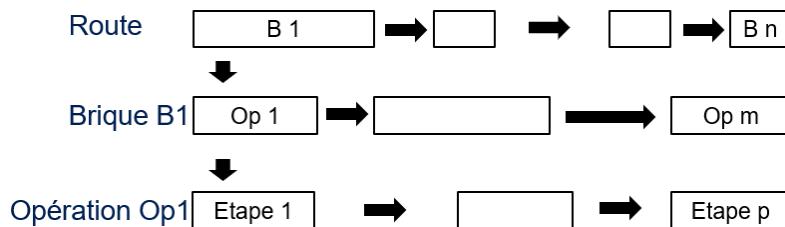


FIGURE 1.7 – Différents éléments de processus d'une route

1.3 Planification de la production dans une entreprise de semiconducteurs

La planification adresse la prise de décisions pour la gestion de la production au sein de la chaîne de fabrication. La planification recouvre des aspects aussi divers que le dimensionnement de capacité et investissement, la gestion des ordres de fabrication en ligne ou les aspects logistiques comme le stockage des produits ou la coordination des systèmes de transport.

1.3.1 Les différentes échelles de planification

La planification est généralement décomposée selon trois échelles : l'échelle stratégique, l'échelle tactique et l'échelle opérationnelle proposée par [Ant65]. Le livre [MFM12] décrit les différents niveaux de décision, au niveau de l'entreprise (long terme), l'usine (moyen terme) et sur le poste de travail (court terme). Les différents niveaux de décision sont résumés dans le tableau 1.1.

- Planification stratégique : planification sur le long terme. Les décisions qui sont prises se traduisent par une politique d'entreprise sur plusieurs années. Les décisions prises concernent les installations de production (par exemple construction d'une nouvelle usine) ou des processus de fabrication (par exemple le développement de nouvelles technologies). La planification stratégique fixe le cadre de la planification tactique.
- Planification tactique : Les décisions prises concernent la production des produits et la commande des ressources selon les prévisions commerciales. Le plan de production calcule les quantités de production par période pour pouvoir répondre aux demandes à moindre coût. La planification tactique donne le cadre de la planification opérationnelle.

Échelle de planification	Problèmes considérés	Horizon de temps
Stratégique	Recherche de nouveaux partenaires industriels Sélection des fournisseurs et sous-traitants Ouverture ou fermeture de sites de production Développement d'un nouveau produit Configuration de l'usine	3 à 5 ans
Tactique	Programme directeur de production, équilibrage de la charge des ressources	3 mois à 1 an
Opérationnelle	Ordonnancement, suivi des ateliers, règles de répartition	De quelques heures à plusieurs semaines

TABLE 1.1 – Problèmes de planification de la production selon l'échelle considérée

- Planification opérationnelle : Les décisions prises concernent le fonctionnement quotidien des ateliers de production. Les décisions prises sont opérationnelles, elles concernent les quantités les dates de passage des lots ou produits par rapport aux capacités des ateliers (par exemple la disponibilité des équipements).

1.3.2 Complexité de la planification de production

Dans la fabrication de composants semiconducteurs, de nombreux facteurs complexifient la planification de production. Concernant Crolles 300, les principaux facteurs de complexité de planification sont :

- Flux réentrants : les lots doivent passer sur le même équipement ou type d'équipement plusieurs fois au cours de leur fabrication. On retrouve donc devant le même équipement des lots (du même produit ou non) à des niveaux différents de leurs routes. Un lot peut être à quelques étapes de la sortie du produit comme au début de sa fabrication, ce qui complexifie la planification.
- Spécificité des processus de fabrication : La nature et la durée des processus de fabrication varient considérablement. Certaines étapes de processus peuvent être exécutées en quelques minutes (par exemple des étapes de nettoyage) et d'autres peuvent durer plusieurs heures (par exemple des étapes de diffusion). Le fonctionnement des équipements est aussi source de différences : un équipement par *batch* traite simultanément plusieurs lots (généralement entre 1 à 6 lots) alors qu'un autre équipement traite un seul lot à la fois plaque par plaque. Le processus par *batch* amène souvent de longues files d'attente sur les équipements ce qui complexifie la planification.
- Grande variété de produits : Les usines de fabrication de semiconducteurs peuvent être classées en deux catégories, *Low-Mix High-Volume* (LMHV) et *High-Mix Low-Volume* (HMLV). Dans le cas LHMV, l'usine produit un faible nombre de produits différents en grande quantité (microprocesseurs, mémoires, etc.). Dans ce cas, les flux de production sont plus simples à piloter car on aura moins de cas spécifiques à considérer que dans le cas d'une production à la demande. Dans le cas HMLV, l'usine produit une large variété de produits pour répondre à la demande de nombreux clients. Les flux de production sont alors plus complexes à piloter puisqu'il faut tenir compte de la demande de chaque client (livraison juste à temps) de façon à ne pas arrêter les chaînes d'assemblage du client considéré.
- Qualification des recettes sur les équipements : à l'intérieur d'un même atelier, pour un même type d'équipement, les opérations à réaliser varient suivant l'étape à laquelle se trouve le produit. De même, suivant les caractéristiques de la technologie, tous les produits ne demanderont pas la même finesse de réalisation. Enfin, suivant son degré d'usure un équipement ne présentera pas le même niveau de performance. La notion de qualification par équipement, par produit et par niveau est donc essentielle pour garantir la qualité des produits. Si un équipement est seul qualifié pour une étape d'un certain produit « mineur » une semaine, il peut très vite se retrouver limitant si le volume de ce produit augmente la semaine suivante et il faudra qualifier un autre équipement pour l'étape et le produit considérés. On parle de la *qualification d'équipement* pour une certaine recette. Ce processus peut être long, mais

aussi très fréquent (plusieurs fois dans une semaine) dans les usines HMLV. Cela complexifie la planification.

L'usine Crolles 300 est considérée comme une usine HMLV ce qui complexifie la planification des différents produits.

1.3.3 Planification de la production dans l'entreprise STMicroelectronics

La planification de production au sein de Crolles 300 est décomposée en plusieurs étapes. Les différentes étapes sont décrites sur la figure 1.8.

- La première étape est de fournir les plans de livraison de l'usine. Les plans de livraison traduisent les quantités de chaque produit à livrer chaque semaine. Les produits ne sont pas livrés directement au client mais aux usines *back end* pour le test électrique, la découpe et la mise en boîtier. Le plan de livraison est lié à une planification tactique fournie par les services centraux qui remplissent la fonction de *planification de production*.
- La deuxième étape est de fournir un *plan de production*. Pour cela le service *planning* fournit un plan de lancement des lots en production, plan défini chaque semaine en fonction des temps de cycle standard et des dates de livraison. Le plan de production est lié à une planification tactique. La définition des plans de production est définie à l'aide d'un outil d'aide à la décision qui prend en compte le plan de livraison et les encours de production de l'usine (Work In Progress WIP).
- La troisième étape est à la frontière entre la planification tactique et la planification opérationnelle. Cette étape appelée *Plan de Référence Interne* a pour but de supporter la gestion des flux de production dans l'ensemble de l'usine. Le calcul du Plan de Référence Interne (PRI) est réalisé à l'aide de l'outil *OPERA* qui en partant des plans de lancement des lots, du plan de livraison, de la position des encours et d'un modèle de capacité de la ligne de fabrication (capacité et qualification des équipements, routes des produits, temps moyen de réalisation des étapes...) va produire un plan détaillé. Ce plan ne correspond pas à un ordonnancement, mais il permet la mise en place de nombreuses actions comme la mise à jour de la priorité des lots (selon leur retard ou leur avance estimés à la livraison), la qualification d'équipements supplémentaires et certaines consignes de production pour l'ordonnancement.
- La dernière étape est de la planification opérationnelle. En fonction des différentes consignes, le système d'ordonnancement automatique va distribuer les lots sur les équipements en fonction des caractéristiques et de la disponibilité de chacun. Le but de cette étape *Ordonnancement /Distribution /Contrôle de Production* est d'optimiser localement la production, en temps réel ou sur un horizon de quelques heures. Selon la complexité des ateliers, les outils et méthodes de décision peuvent changer. Dans les ateliers critiques, les décisions sont prises par des outils complexes d'ordonnancement (*scheduler*) les autres ateliers utilisant des outils plus simples, à base de règles de pondération et de tri par exemple (Distribution/ contrôle de production).

Le Manufacturing Execution System (MES) est un système de gestion par ordinateur qui sert à gérer la production dans une usine.

A travers les différentes étapes de planification, les plans stratégiques sont peu à peu transformés par des consignes plus précises et transmises au niveau des ateliers pour aboutir à des décisions en temps réel sur le choix des lots à lancer sur chaque équipement.

1.4 Contrôle de production

Dans la figure 1.8 la vérification du respect des consignes de production est effectuée par le contrôle de production. Ce contrôle est un aspect important pour vérifier que le plan de production est respecté par rapport à la réalité et pour ne pas prendre de retard sur les livraisons.

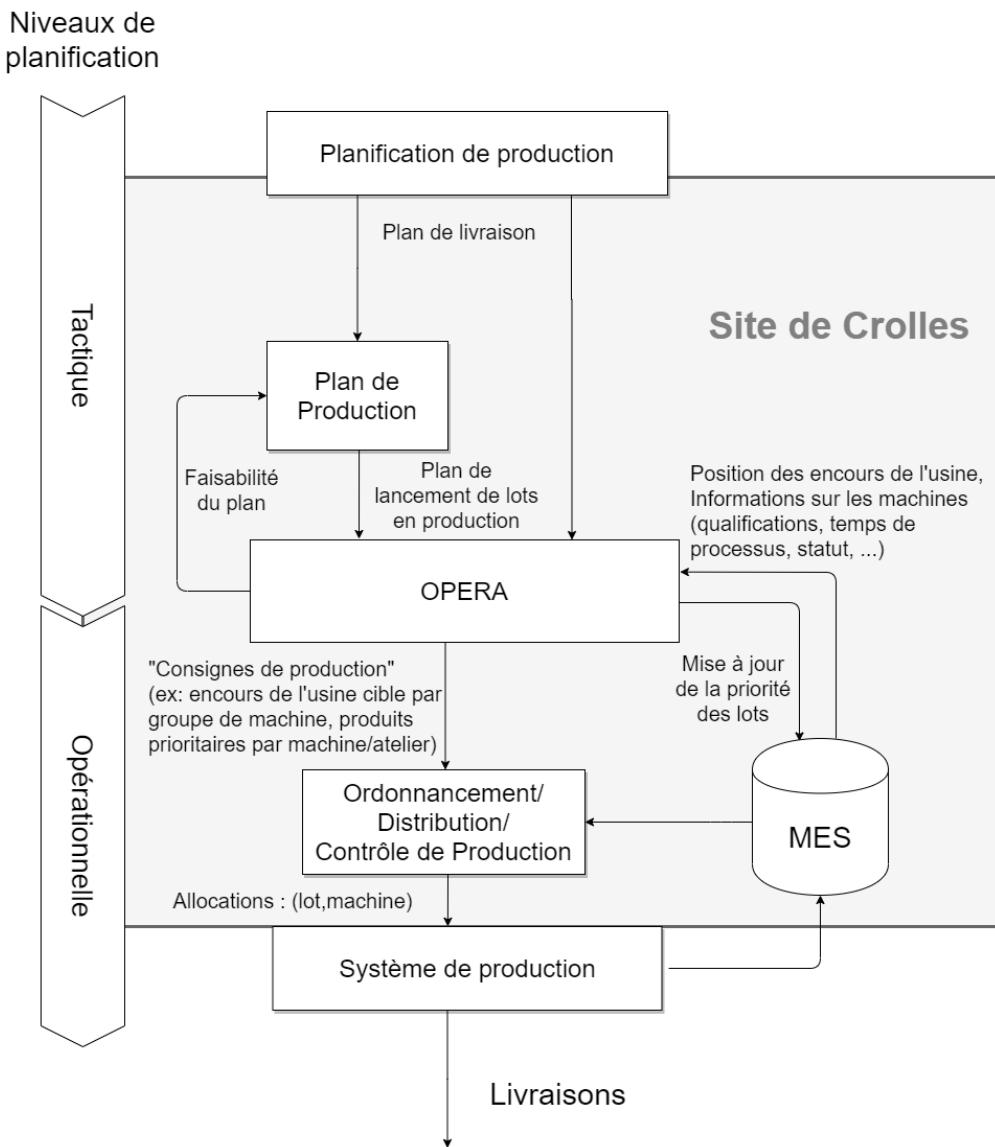


FIGURE 1.8 – Étapes de planification de production à STMicroelectronics Crolles extraites de [Chr20]

1.4.1 Principe du contrôle de production dans la littérature

Le contrôle de production est utilisé au niveau opérationnel dans la planification. L'objectif de ce contrôle est de veiller à la bonne répartition des encours dans l'usine. Les méthodes de contrôle de production peuvent être décomposées en deux approches. La première consiste à changer les consignes de pilotage de la production quand les différents indicateurs indiquent une dérive. La seconde s'appuie sur l'utilisation de points de contrôle : les goulots d'étranglement¹. Pour une ligne de produits (généralement une technologie), les goulots d'étranglement sont les points limitants la production en termes d'activité et de capacité. Il existe deux types de goulots d'étranglement. Les goulots d'étranglement structurels sont une limitation par rapport à la capacité de production des équipements. Les goulots d'étranglement fonctionnels peuvent quant à eux changer au cours du temps, puisqu'ils sont liés à des phénomènes ponctuels tels que l'indisponibilité des équipements ou la variabilité des encours. Des indicateurs qui aident à la décision sur les équipements en goulots d'étranglement pour le contrôle de production sont proposés dans [LG88]. Les goulots d'étranglement sont les équipements/ateliers qui vont être limitant pour la production. Les indicateurs proposés dans l'article sont le contrôle des temps de cycle des produits, la comparaison des diffé-

1. Dans l'entreprise STMicroelectronics les goulots d'étranglement sont notés bottlenecks

rentes politiques d'ordonnancement et le contrôle des stocks (le contrôle des plaques qui arrivent sur les équipements/ateliers en goulot d'étranglement).

Une revue des problèmes de contrôle par atelier est faite dans [ULMV94]. Une l'évolution des problèmes de planification et de contrôle opérationnel dans la production de semiconducteurs et des approches possibles pour les résoudre des approches de résolutions est présentée dans [GRFM06]. Le contrôle de production établie dans [BW81] le lien entre le rendement et le temps de passage moyen dans l'usine. Il permet le maintien des temps de passage afin de fixer le temps de fin de production et ainsi réduire la variabilité dans l'usine pour améliorer les planifications. Une autre stratégie pour le contrôle est la règle d'une boucle fermée pour les ateliers définis dans [GR88] avec comme principe fondamental le hasard des indisponibilités des équipements. Le principe de ce contrôle sur l'ordonnancement est de comparer plusieurs entrées dans l'ordonnancement sur plusieurs ateliers. Une autre méthode est d'intégrer le contrôle des usines avec le contrôle des équipements, la supervision de l'usine, le retour d'informations des équipements et le contrôle des temps de processus par des méthodes statistiques pour faire de la détection/diagnostic des défauts dans la production des semiconducteurs comme le montre [EBC⁺00]. Un autre approche est de considérer le problème de contrôle de production par la réduction du temps de cycle dans les usines de semiconducteurs LMHV, cette méthode de contrôle de production appelée DDescending Control (DEC) se base sur la simulation et l'optimisation pour améliorer les règles de contrôle de production, méthode proposée dans [GSS96].

1.4.2 Principe du contrôle de production dans l'usine de Crolles

Au sein de l'usine Crolles 300 de STMicroelectronics, une fois que le service *planning* a calculé les objectifs de pilotage pour la semaine et publié le Plan de Référence Interne (PRI), le service *WIP Management* est chargé de faire réaliser les objectifs au moyen de consignes de production. Le Service *WIP Management* observe en continu les flux de production dans les différents ateliers pour modifier les consignes de production selon l'évolution de l'activité par rapport aux objectifs. Le service *WIP Management* modifie les consignes de production au cours de la journée en fonction des événements : retard important de certains produits, blocage potentiel pour une technologie donnée ou équilibrage de la ligne. Pour ce faire le service modifie essentiellement la priorité des lots, ce qui va influer l'exécution au niveau de l'ordonnancement et de la distribution. Au cours de la journée selon l'avancement de l'activité par rapport aux objectifs détaillés, les consignes peuvent être aussi modifiées pour accélérer le traitement des lots en amont de certains ateliers avec plus de capacités disponibles.

Le mode de gestion actuel, s'il est bien adapté aux produits à fort volume, atteint néanmoins ses limites sur les produits à faible volume. En effet, comme les objectifs sont calculés sur une moyenne de 7 jours, réagir au pourcentage de déviation n'est pas possible : on serait toujours très négatif ou au contraire très positif pour tout un ensemble de lots potentiellement perturbateurs de l'ordonnancement. Pour faire le contrôle de production, le service *WIP Management* ne regarde pas la globalité de l'usine mais regarde atelier par atelier donnant des consignes pour chaque atelier. Ces consignes peuvent évoluer au cours du temps selon la situation. Pour vérifier que l'usine respecte bien la planification, le service *WIP Management* vérifie certaines étapes essentielles dans la route de chaque produit, mais est très loin de surveiller l'ensemble du système. Après, selon les conditions de l'usine, le service *WIP Management* regarde aussi l'évolution des goulets d'étranglement pour vérifier que les ateliers limitants ont toujours des encours pour alimenter leurs goulets d'étranglement respectifs et limiter leurs impacts en termes de retard.

1.5 Synthèse du chapitre

Le chapitre présente le fonctionnement de l'industrie des semiconducteurs et les principales étapes de fabrication d'un composant électronique. Ensuite, on présente la planification dans les entreprises de semiconducteurs et les différentes échelles de planification ainsi que les facteurs qui font que cette activité est complexe dans le cas de l'unité de production de Crolles300. Le contrôle de production montre qu'il est difficile d'avoir une vision globale de l'usine pour détecter les retards que l'on peut avoir sur les produits. La figure 1.9 représente le lien entre la partie opérationnelle de la planification avec le PRI qui permet de faire une validation des objectifs de pilotage de la semaine et le service de contrôle de production *WIP Management* qui donne des consignes de production au cours de la journée. L'objectif de mes travaux, comme le montre la figure 1.9, est

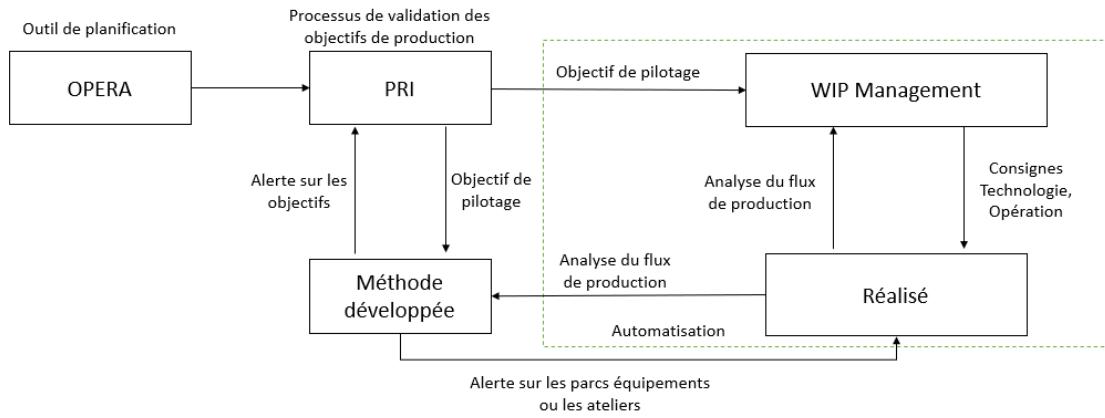


FIGURE 1.9 – Représentation du lien entre la partie opérationnelle de la planification et le contrôle de production

d'automatiser la détection de dérives potentielles dans les flux de production. Partant de la mesure de l'activité de production et des objectifs correspondants, l'objectif est de proposer une approche permettant une détection globale des dérives et d'alerter le service *WIP Management* en cas de dérives observées. Pour faire cela une connaissance des outils de planification et des données de l'activité de production est présentée dans le manuscrit.

Chapitre 2

Retard de la production de l'usine par rapport à la planification

Ce chapitre présente dans un premier temps le fonctionnement du logiciel de planification utilisé dans l'entreprise STMicroelectronics Crolles ainsi que les données de l'usine et de la planification qui ont été mises à disposition dans le cadre de cette thèse. En particulier, les différents paramétrages qui sont utilisés pour la planification sont détaillés. Dans un second temps, le chapitre discute de la notion de variabilité dans les chaînes de production et comment cette variabilité se caractérise au sein de l'usine Crolles300 par une analyse des données réelles de production. Suite à cette analyse, une méthode de comparaison par opération entre l'historique et la planification est proposée afin de déterminer la présence de retard. Enfin, le chapitre introduit la notion de retard critique et propose un indicateur qui mesure ce type de retard.

2.1 Performance de l'outil de planification

Cette section présente le fonctionnement de l'outil de planification et les différentes données qui sont à notre disposition dans le cadre de nos travaux.

2.1.1 Présentation du l'outil de planification

L'outil de planification utilisé par ST Crolles, nommé OPERA, se base sur la méthode présentée dans [MMJ⁺18]. Cette méthode heuristique est capable de résoudre des problèmes de taille réelle (avec environ 6000 lots, 500 équipements) avec un temps d'exécution de moins d'une minute. Elle se décompose en trois étapes présentées sur la figure 2.1 [CDPLV18]. La première phase est la

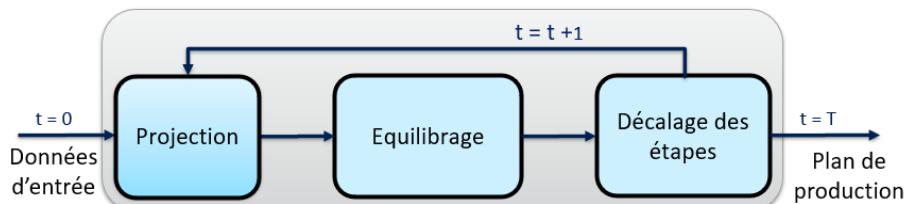
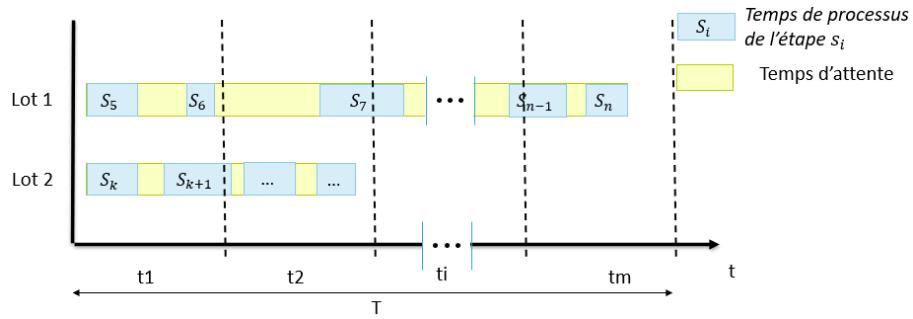


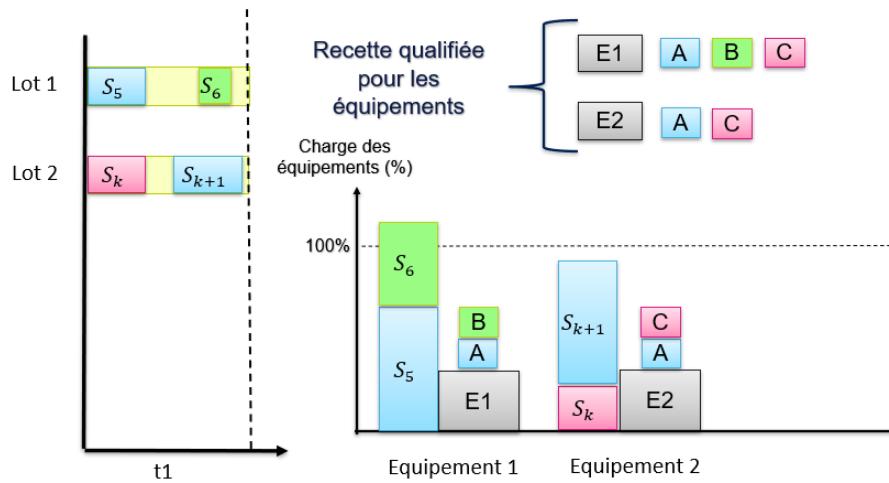
FIGURE 2.1 – Schéma de la méthode heuristique en trois phases

projection, elle permet avec les informations des données d'entrée de fixer chaque étape de processus des lots avec les différentes entrées, la date de début et de fin de chaque étape de processus de lot.

Exemple 2.1. La figure 2.2 présente la première phase de la méthode heuristique qui comme le montre cette figure consiste à projeter chaque lot selon sa date de début et le temps de chaque étape de processus. Par exemple, l'étape S_5 du lot 1 est planifiée pour un démarrage en période t_1 suivie par l'étape S_6 puis de S_7 planifiée sur la période t_2 . La projection des étapes sur le lot 1 s'effectue jusqu'à la fin de la période t_m .

FIGURE 2.2 – Projection de deux lots sur une période $T = \sum_{i=1}^m t_i$

La deuxième phase est l'équilibrage, elle a pour but de répartir les lots sur les équipements disponibles.

FIGURE 2.3 – Équilibrage des charges de deux lots sur la période t_1

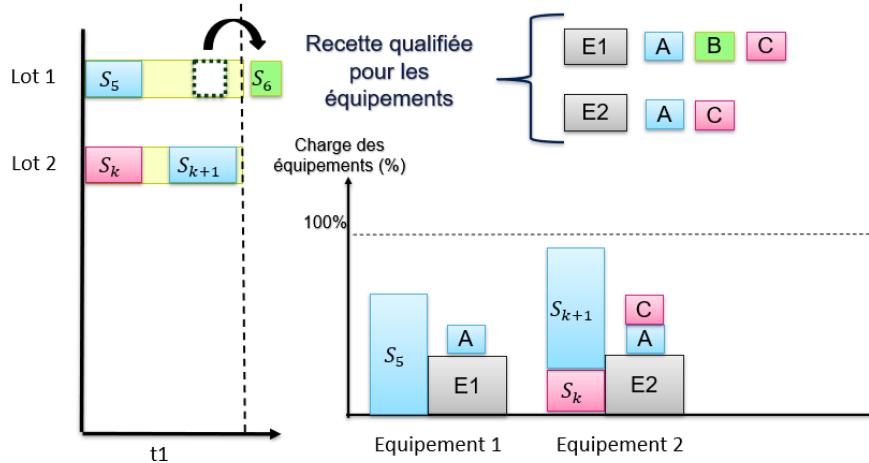
Exemple 2.2. Suite à la projection de la figure 2.2 l'algorithme se concentre sur la première période (t_1) pour faire l'équilibrage de charge pour les 4 étapes qui sont exécutées dans cette période. Le résultat de cette phase d'équilibrage est présenté sur la figure 2.3. Le lot 1 doit faire les étapes S_5 et S_6 utilisant les recettes A et B qui peuvent être effectuées toutes les deux sur l'équipement 1 (voir les recettes qualifiées pour chaque équipement en haut à droite de la figure 2.3). Le lot 2 doit effectuer les étapes S_k et S_{k+1} utilisant les recettes C et A réalisables sur les équipements 1 et 2. Lors de cette phase d'équilibrage, les étapes S_5 et S_6 sont affectées à l'équipement 1 si bien que la charge de cet équipement est supérieure à 100%.

La troisième phase est le décalage des étapes sur des périodes ultérieures (t_{i+1}, \dots) qui a pour but de lisser la charge de travail des équipements et de satisfaire les contraintes de capacité des équipements.

Exemple 2.3. Après avoir fait l'équilibrage de charge de la figure 2.3, la charge sur l'équipement 1 est trop importante par rapport à la capacité de l'équipement. La deuxième étape S_6 du lot 1 est décalée à la période suivante t_2 ce qui permet de désaturer l'équipement 1 sur la période t_1 qui n'a plus que l'étape S_5 et S_k à effectuer. Après cette étape l'algorithme effectue une nouvelle projection au temps t_2 des lots suivie d'un équilibrage et d'un décalage et ainsi de suite sur la période T .

La méthode présentée dans [CDPLV18] reprend celle de [MMJ⁺18] en améliorant les parties d'équilibrage et de décalage des étapes et prend en compte différents objectifs de planification. La méthode de l'article [CDPLV18] est mise en œuvre dans le logiciel OPERA utilisé chez ST Crolles.

OPERA est un logiciel qui est utilisé pour résoudre différents types de problèmes : notamment il permet de faire de la projection comme présentée ci-dessus mais il peut également être utilisé

FIGURE 2.4 – Décalages des étapes de deux lots sur la période t_1

pour de la simulation de plans de production de l'usine. L'outil permet également de faire une planification sur différentes périodes (par exemple sur un jour, une semaine...). La figure 2.5 présente différentes entrées de l'outil de planification.

- La configuration indique sur quelle période la planification est effectuée, si les dates de sortie doivent être respectées et si les capacités des équipements doivent être prises en compte.
- Les plans de lancement et de sortie des lots informent sur les dates de démarrage de la production de nouveaux lots et sur les dates attendues de livraison des encours.
- Le positionnement des encours permet d'identifier les lots et leur localisation dans l'usine au moment où la planification est lancée.
- La capacité des équipements précise la disponibilité des équipements et leurs qualifications courantes.
- Le temps de cycle indique le temps moyen de processus et d'attente des étapes d'une opération pour chaque route.

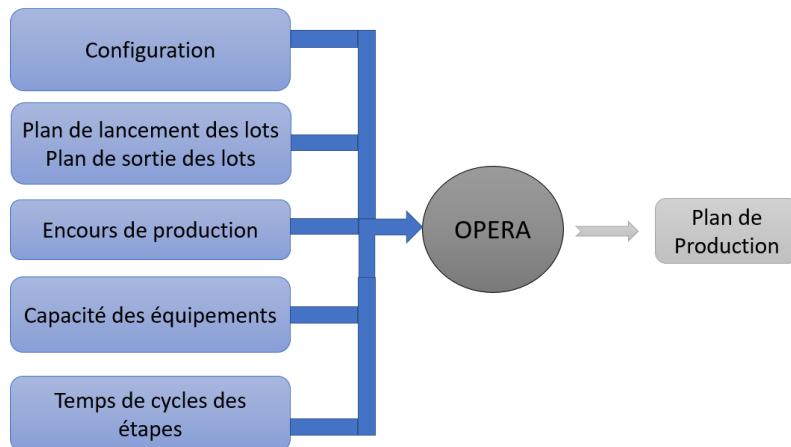


FIGURE 2.5 – Schéma des entrées et des sorties de l'outil OPERA

Les temps de cycle sont notamment calculés grâce à une méthode d'optimisation-simulation. L'article [GHL18] récapitule toutes les méthodes optimisation-simulation appliquées aux semiconducteurs. Il explique que de nombreuses méthodes permettent de faire de la planification et aussi de faire de la simulation. Mais la convergence ou non convergence des solutions entre la simulation et la planification n'est pas encore bien comprise. La méthode est très dépendante des valeurs initiales et peut ne pas converger. De plus, selon la complexité de la chaîne de production la charge

du modèle de simulation peut devenir importante. La capacité des équipements est calculée par rapport aux différentes qualifications en fonction des encours de l'usine et des maintenances qui vont être mises en place dans la semaine planifiée. Les plans de lancement des lots sont fournis par le service *planning* d'après la figure 1.8.

2.1.2 Les données disponibles dans le cadre de nos travaux

Dans le cadre de ces travaux, les données qui sont à notre disposition sont les différents fichiers de sortie de OPERA comme les fichiers de type *LOTSPROJECTION* qui décrivent les différentes étapes planifiées sur une période donnée. Un fichier *LOTSPROJECTION* contient plusieurs colonnes. Dans ce fichier on utilise ici les colonnes suivantes :

Technologie	Brique	Operation	ETAPE	STEP_OUT	Track_Qty.	TYPE
<i>Techno1</i>	<i>b₁</i>	<i>o₂</i>	<i>e₃</i>	14/10/20- 06 :39 :21	25	OPERA
<i>Techno1</i>	<i>b₁</i>	<i>o₂</i>	<i>e₃</i>	14/10/20- 07 :46 :21	25	OPERA

TABLE 2.1 – Colonnes utilisées du fichier *LOTSPROJECTION* issu de la planification

La colonne *STEP_OUT* correspond à la date de sortie planifiée d'une étape pour un lot donné. La colonne *Track_Qty* correspond au nombre de plaques du lot. La colonne *TYPE* correspond au fait que l'étape vient du tableau de la planification.

Pour les données de l'usine (l'historique), on a à notre disposition le fichier *TGVHistorique* qui est tiré du MES et qui regroupe toutes les étapes réellement effectuées sur une période donnée.

Le fichier *TGVHistorique* contient plusieurs colonnes. Parmi celles-ci, les colonnes suivantes sont utilisées ici :

Technologie	Brique	Operation	ETAPE	STEP_OUT	Track_Qty	TYPE
<i>Techno1</i>	<i>b₁</i>	<i>o₂</i>	<i>e₃</i>	14/10/20- 06 :23 :21	25	Histo
<i>Techno1</i>	<i>b₁</i>	<i>o₂</i>	<i>e₃</i>	14/10/20- 07 :10 :21	25	Histo

TABLE 2.2 – Colonnes utilisées du fichier *TGVHistorique*

La colonne *STEP_OUT* correspond à la date de fin de la réalisation de l'étape sur un lot donné. La colonne *Track_Qty* correspond au nombre de plaques qui sortent de l'étape¹. La colonne *TYPE* correspond au fait que l'étape vient du tableau de l'historique.

2.1.3 Utilisation de l'outil de planification selon différents paramètres

Comme nous l'avons vu dans la section 2.1.1, avec les différentes configurations de l'outil et les différentes entrées, le logiciel peut produire un ensemble de planifications possibles. Notre objectif dans cette section est de comparer les différents paramétrages de cette planification avec des données historiques afin de vérifier la fiabilité de l'outil de planification.

Dans un premier temps, l'idée est de vérifier que la planification effectuée par le logiciel correspond à la réalité. Pour cela on compare le plan de la planification avec les données de l'historique pour une semaine donnée, en exploitant les fichiers *LOTSPROJECTION* et *TGVHistorique* de la semaine en question. La comparaison utilise le logiciel *Spotfire* qui est une plate-forme d'analyse de données. L'avantage du logiciel *Spotfire* est de permettre de faire des associations entre différentes données et de faire des représentations graphiques de ces données. La comparaison s'effectue entre la somme du nombre de plaques en sortie d'une étape pour une journée et la somme prévue par OPERA. La planification a été réalisée avec le paramétrage *ADD_PRI_FINI* (prise en compte des dates de fin et de la capacité des équipements, voir les détails ci-dessous). La figure 2.6 présente le résultat de cette comparaison pour la semaine débutant le 11/12/2019. Alors que sur les trois premiers jours de la semaine, la planification hebdomadaire (en rouge) et l'historique (en bleu) sont plutôt en cohérence (par exemple, 30500 plaques planifiées pour 31000 plaques réellement sorties le 11/12/2019), un écart plus significatif entre l'historique et la planification se manifeste ensuite (la planification a tendance à planifier moins d'étapes en fin de semaine).

La figure 2.7 montre une deuxième comparaison entre la planification et l'historique pour la semaine 47 en 2020. La figure 2.7 montre la présence constante d'une différence entre le nombre

1. Dans l'entreprise STMicroelectronics le nombre de plaques en sortie d'étape est notée *Track_Out*.

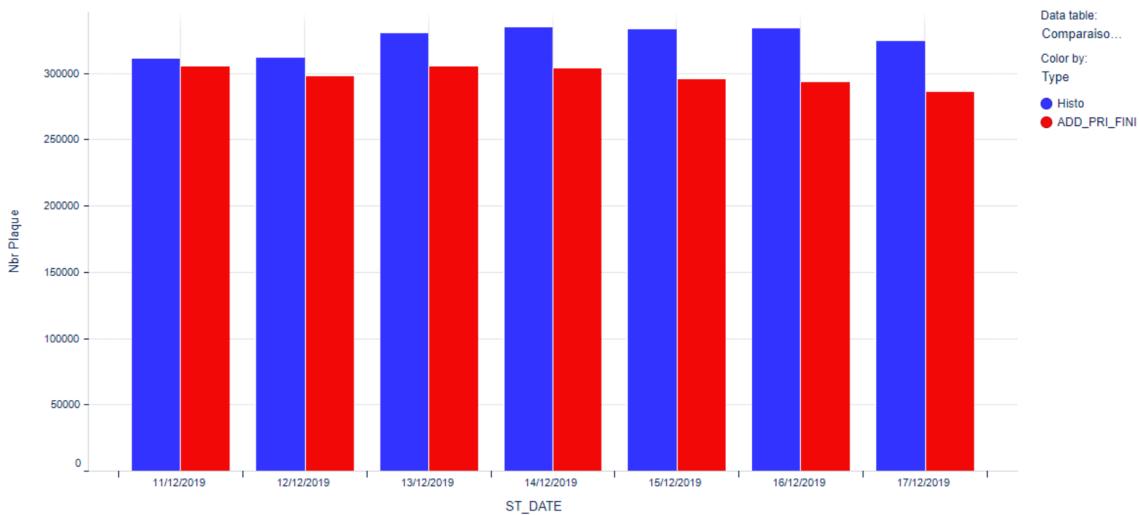


FIGURE 2.6 – Comparaison du nombre de plaques en sortie d'une étape de la semaine 50 de 2019 entre la planification et l'historique

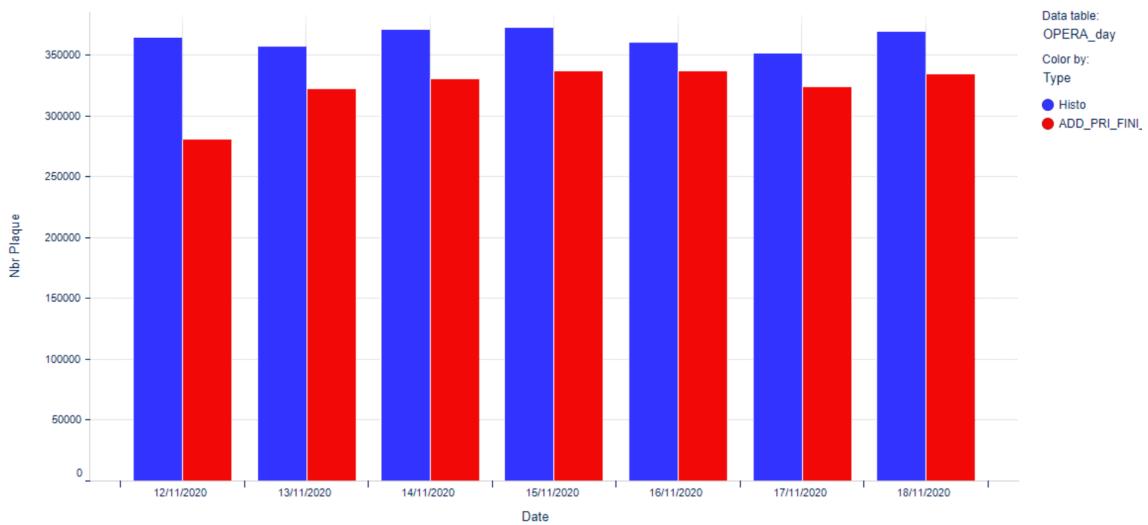


FIGURE 2.7 – Comparaison du nombre de plaques en sortie d'une étape de la semaine 47 de 2020 entre la planification et l'historique

planifié de plaques en sortie d'une étape et le nombre réel, différence importante tout au long de la semaine planifiée. Contrairement à la semaine du 11/12/2019, les écarts sont ici toujours élevés avec un écart maximal de 8000 plaques pour le 12/11/2020.

Les comparaisons précédentes s'appuient sur le paramétrage de planification réellement utilisé en exploitation. Pour aller plus loin on va maintenant comparer différents paramétrages du logiciel OPERA afin de vérifier si il n'y a pas un paramétrage qui permet un écart minimal entre la planification et l'historique sur les deux mêmes semaines.

Les différents paramètres de l'outil OPERA permettent de faire plusieurs planifications. Selon ces différents paramètres le résultat des planifications est varié, les planifications permettant d'avoir de nouvelles informations. Les paramètres de la planification sont :

- Due Dates (date de sortie d'un lot) : la planification peut s'effectuer soit en prenant en compte les dates de sortie des lots (Avec Due Date ADD) soit en ne les prenant pas en compte (Sans Due Date SDD).
- Capacité Finie : l'outil peut calculer la planification de l'usine en prenant en compte la capacité des équipements (FINI), c'est-à-dire que la planification prend en compte le temps de disponibilité des équipements.

- Capacité Infinie : l'outil peut calculer la planification de l'usine en ne prenant pas en compte cette capacité des équipements (INFINI).

Les différents paramètres de l'outil de planification permettent de faire quatre plans de production différents. Ces plans de production vont varier entre la prise en compte des dates de fin des lots et la prise en compte des capacités des équipements. Les quatre plans de production comparés ont été calculés *a posteriori* pour cette comparaison, ils débutent le même jour avec les mêmes entrées et sur la même période de planification. Pour faire la comparaison, on utilise les fichiers *LOTSPROJECTION* qui regroupent chaque étape de processus de chaque lot à la date planifiée sur la totalité de la planification. Ces fichiers sont importés et regroupés à l'aide du logiciel *Spotfire*. Les données historiques ont été également rajoutées à notre base d'étude. La comparaison s'effectue avec la somme du nombre de plaques en sortie d'une étape pour une journée et la somme planifiée. Plusieurs comparaisons ont pu être réalisées sur l'ensemble de l'historique et deux semaines sont présentées ici pour illustrer significativement la comparaison.

La figure 2.8 montre graphiquement la comparaison des quatre planifications et l'historique sur une période d'une journée avec :

- En bleu, l'historique (Histo).
- En rouge, la planification avec les dates de sorties des lots et avec les capacités des équipements (ADD_PRI_FINI).
- En jaune, la planification avec les dates de sorties des lots et sans les capacités des équipements (ADD_PRI_INFINI).
- En vert, la planification sans les dates de sorties des lots et avec les capacités des équipements (SDD_PRI_FINI).
- En rose, la planification sans les dates de sorties des lots et sans les capacités des équipements (SDD_PRI_INFINI).

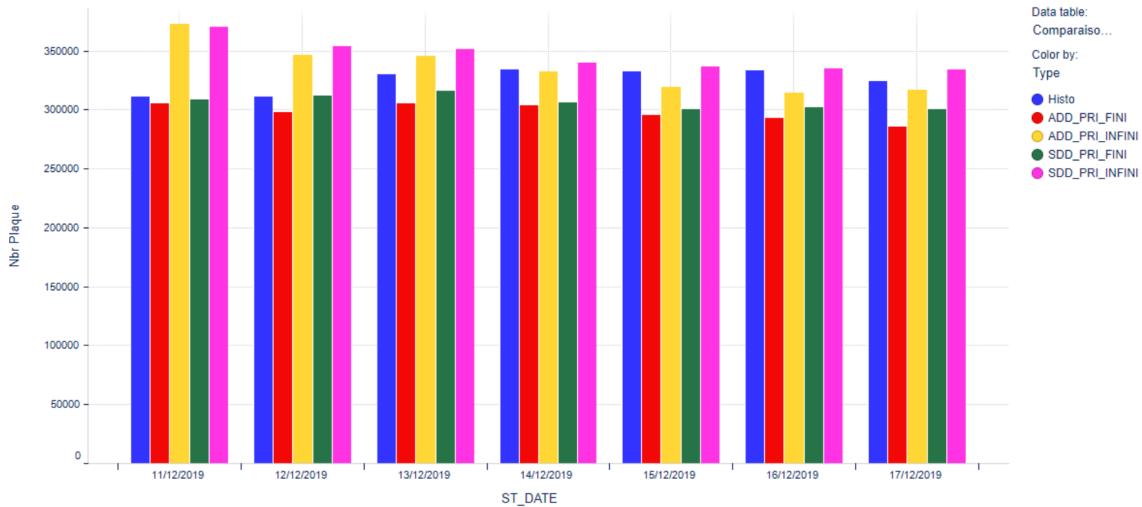


FIGURE 2.8 – Comparaison des trackouts de la semaine 50 de 2019 entre les différentes planifications et l'historique sur une période d'une journée

On observe sur la figure 2.8 que les planifications en rouge et en vert sont les plus proches de la réalité (en bleu). En revanche aucune d'entre elles n'est en parfaite cohérence avec l'historique. Sur la figure 2.9 pour la journée du 12/11/2020 les planifications en jaune et en rose sont dans ce cas les plus proches de l'historique. Ainsi, à la vue des résultats, sur le semaine 50 de 2019, la planification réellement exploitée (ADD_PRI_FINI) est la plus en cohérence avec l'historique alors que sur la semaine 47 de 2020, un autre paramétrage de la planification (ADD_PRI_INFINI) aurait été plus en cohérence avec l'historique.

L'analyse faite sur les différents paramètres de l'outil de planification montre ainsi que selon les paramètres de la planification utilisés il n'y a pas de paramétrage dominant dans la qualité

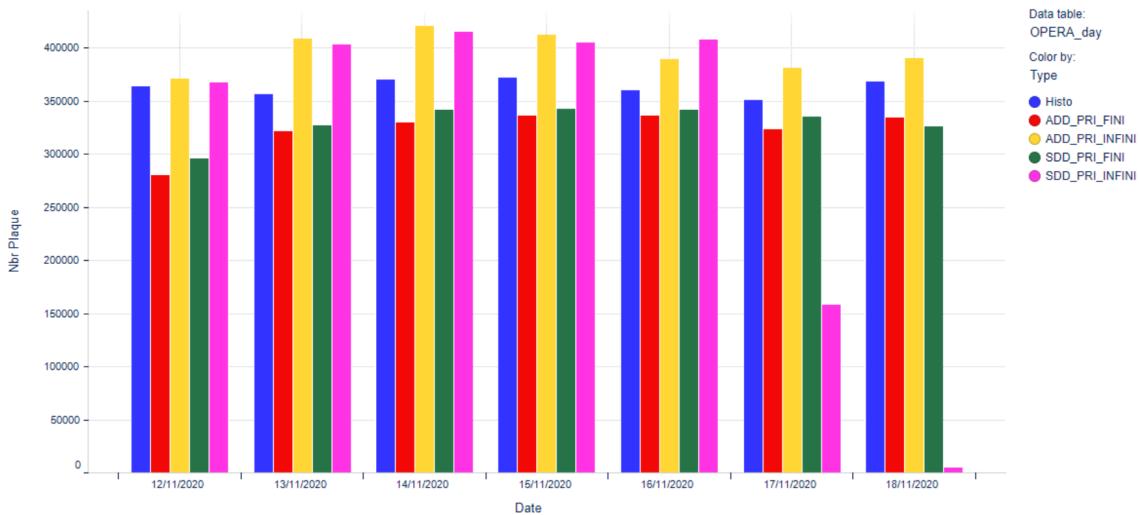


FIGURE 2.9 – Comparaison des nombres de plaques en sortie d'une étape de la semaine 47 de 2020 entre les différentes planifications et l'historique sur une période d'une journée

de la planification. Ce phénomène semble dû à la forte variabilité de l'activité et des flux entrant des ateliers. Un paramétrage unique et optimal ne peut être envisagé et doit être fonction de la situation de l'usine (capacités équipements, position des en-cours,...). En pratique dans l'usine de ST Crolles l'outil de planification est utilisé dans différents modes et pour différents niveaux de planification. Sur la figure 1.8 le service de *planification opérationnelle de la production* utilise l'outil de planification pour faire une planification sur une période d'une semaine et à capacité finie, cette planification est utilisée pour définir les objectifs de production. L'outil de planification est aussi utilisé tous les matins sur une période de 8h et à capacité infinie (ADD_PRI_INFINI) pour identifier les lots qui sont en retard par rapport à leurs *due dates*. L'outil de planification est également utilisé dans l'atelier photolithographie pour anticiper les besoins des masques et les pré-positionner sur une période de 8h et à capacité infinie (ADD_PRI_INFINI) pour savoir quels masques peuvent être sortis ou entrés dans les équipements de photolithographie.

L'analyse des différents paramètres de la planification montre qu'il n'existe pas un paramétrage optimal dans l'utilisation de l'outil dans l'usine Crolles 300. Ainsi, dans la suite de nos travaux on utilise le paramétrage qui est défini par le service *planification opérationnelle de la production* et qui est utilisé par ce service pour la planification hebdomadaire. Pour rappel, ce paramétrage est à capacité finie et avec *due dates* (ADD_PRI_FINI).

2.2 Caractérisation de la variabilité de la production

2.2.1 Définition de la variabilité de la production dans l'usine

Dans le contexte des flux de fabrication, la notion de *variabilité* fait référence à un phénomène qui affecte le flux de production et entraîne des imprécisions dans la livraison, ce qui peut engendrer des retards de livraison. Plus précisément, la variabilité implique la perte de contrôle des flux. Les usines de fabrication de semiconducteurs sont soumises à des niveaux élevés de variabilité en raison de divers facteurs, notamment les flux ré-entrants, les produits multiples, les pannes des équipements ou les processus différents en fonction des lots. Cette variabilité diminue la productivité, augmente les temps de cycle et affecte sévèrement le suivi du système.

L'article [DVLE16] fait une revue des sources de variabilité dans l'industrie des semiconducteurs et des méthodes pour les gérer. L'article présente une classification en quatre sources qui sont les équipements, les produits, l'opérationnel et le structurel. Les différentes sources de variabilité sur les équipements sont la variabilité des temps de processus selon les équipements, les produits et les niveaux considérés, le temps d'indisponibilité des équipements, le traitement des équipements par lot ou par ensemble de lots, le statut des équipements et les séquences particulières des processus comme le décrivent les articles [RFN03] et [SDZ07]. Les différentes sources de variabilité sur les produits sont la variété des produits, les retours en arrière (quand une erreur arrive sur un lot, le lot est dirigé sur une autre route ce qui engendre plus de mesures) ou les lots qui sont mis au rebut

(les lots sont alors détruits) quand ils ne sont pas conformes aux spécifications et ne peuvent être repris, la priorité des lots, la taille des lots et les contraintes temporelles (une contrainte temporelle étant le temps maximal à ne pas dépasser entre deux étapes de processus, consécutives ou non, afin de garantir la validité des plaques). Les différentes sources de variabilité sur l'opérationnel sont les règles d'ordonnancement des lots sur les équipements, la stratégie du contrôle des en-cours de l'usine, la maintenance des équipements, la mise en attente des lots, les dates de lancement des lots (plan de *start* des lots), les inspections des plaques dans un lot, les effets de fin d'équipe et les arrêts de l'usine. Enfin, pour le structurel, la variabilité se retrouve dans le manque de redondance des équipements (le nombre d'équipements parallèles qui traitent une étape donnée du processus), les flux réentrants des équipements, la spécification des équipements, l'hétérogénéité des équipements (la disparité des caractéristiques des équipements), le croisement des opérations, les ressources de production. L'identification des facteurs contribuant à la perte de capacité dans les installations de fabrication de plaques de semiconducteurs est également présentée dans [RFN03]. Par exemple, la question de l'identification des équipements qui introduisent de la variabilité dans le système et limitent la capacité du système est examinée dans [DSF06]. Pour cela, la décision d'affectation des qualifications des équipements aux opérations dans l'atelier de photolithographie pour la fabrication de plaques de semiconducteurs présentée dans l'article [Ign09] est utilisée pour illustrer le problème. Cette décision permet de déterminer la configuration et le fonctionnement d'une usine en présentant les opérations devant être exécutées, et ainsi optimiser les performances globales de l'usine. Une autre méthode d'identification consiste à examiner les applications de la théorie des files d'attente pour les systèmes de fabrication de semiconducteurs afin d'en réduire la variabilité [SDZ07].

Pour en revenir à la revue des sources de variabilité de [DVLE16], l'article définit différentes méthodes pour visualiser son influence par des "courbes opérationnelles" ou "courbes de débit de temps de cycle". Ces courbes sont utilisées dans [Sch99] qui présente deux simulations expérimentales utilisant un modèle de taille réelle d'une usine de fabrication de semiconducteurs multi-produits. L'article explique que la variabilité est causée par l'indisponibilité de certains équipements. Les courbes représentent le *X-factor* (rapport entre le temps de cycle observé et le temps de processus brut) en fonction de l'utilisation de la capacité disponible à différents niveaux de variabilité. Cette courbe permet de montrer que plus la variabilité est importante plus le temps de cycle est important. Un autre impact de la variabilité présenté dans [RFN03] est qu'elle a une influence sur la perte de la capacité des équipements, c'est-à-dire que plus la variabilité est importante plus la capacité des équipements est réduite. En effet, la capacité physique ne change pas mais par contre, si on veut obtenir le même temps de cycle, alors il faut réduire la capacité utilisée pour la planification.

Une autre méthode a été proposée pour analyser le problème de la variabilité dans la production des lignes de fabrication en série avec des équipements utilisant les statistiques de Bernoulli sur la fiabilité des équipements. Cette méthode utilise les chaînes de Markov présentées dans [LM00]. Trois problèmes sont examinés : le problème de la variance de la production (*variabilité des produits*), le problème de la satisfaction de la demande constante et le problème de la satisfaction de la demande aléatoire générée (*variabilité de la quantité*). Pour ces trois problèmes, des limites sont fixées aux mesures de variabilités respectives. Les limites présentées montrent que les longues lignes de production lissent la production et réduisent la variabilité (c'est-à-dire si pas de variabilité en entrée, on lance des quantités stables). Plus précisément, ces limites indiquent que la variabilité de la production d'une ligne comportant de nombreux équipements est plus faible que celle d'un système à un seul équipement dont le volume de production et les caractéristiques de fiabilité sont similaires.

2.2.2 Analyse de la variabilité dans l'usine Crolles 300

Dans cette section on présente la variabilité dans l'usine Crolles 300. L'objectif général est que l'usine fonctionne toujours en flux tendu au niveau de sa production. Ainsi, la production maximise l'utilisation des équipements et minimise le temps d'attente des lots. L'usine étant à 98% d'automatisation pour le transport des lots sur les équipements et pour le lancement des lots sur les équipements, dans notre analyse, on part du principe que la production n'est pas influencée par la variabilité. Pour mettre en évidence la présence ou non d'une potentielle variabilité dans la production, cette section décrit une analyse globale exploitant les données historiques de production sur plusieurs semaines.

Fonctionnement de l'usine Crolles 300

Dans un premier temps, nous décrivons brièvement le fonctionnement de l'usine Crolles 300. Il faut en effet considérer les particularités des usines de semiconducteurs comme le fonctionnement 24 heures sur 24 et 7 jours sur 7. Pour le fonctionnement de cette usine, il y a 5 équipes différentes. Les équipes 1 et 2 alternent le travail sur une durée de 8h (la durée du travail est nommée *shift*) du matin (S1) de 5h à 12h45 et l'après-midi (S2) de 12h45 à 20h30 du lundi au jeudi. L'équipe 3 qui travaille sur le *shift* de la nuit (S3) de 20h30 à 5h entre 3 et 4 nuits suivant les semaines. Le vendredi le *shift* du matin s'effectue par l'équipe 4 de 5h à 13h30 puis soit par l'équipe (selon l'équipe de l'après-midi dans la semaine) de 13h30 à 21h15 et enfin les équipes 5 ou 3 de 21h15 à 5h. Les samedi et dimanche il n'y a que 2 *shifts* celui du matin de 5h à 17h par l'équipe 4 et celui de l'après-midi de 17h à 5h. La Figure 2.10 représente la répartition des équipes sur 3 semaines.

S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J
12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
					3								4						
4	1	1	1	1	1	4	4	2	2	2	2	2	4	4	1	1	1	1	
5	2	2	2	2	2	2	5	1	1	1	1	1	1	5	2	2	2	2	
5	3	3	3	3	5	3	3	3	3	3	3	3	5	3	3	3	3	3	

FIGURE 2.10 – Calendrier de la répartition des équipes sur trois semaines

Analyse de la variabilité de l'usine avec des données historiques

Pour analyser la variabilité dans l'usine, on se propose de mettre en place une méthode exploitant les données issues du fichier *TGVHistorique* présenté dans la sous-section 2.1.2. L'objectif est de décrire le comportement global de l'usine en exploitant un grand nombre de données pour identifier si la variabilité est présente dans l'usine. Cette méthode se base l'étude de la variabilité sur une journée classique, à savoir les journées du mardi, mercredi et jeudi. En effet, les journées du lundi et vendredi ne sont pas prises en compte car ce ne sont jamais les mêmes équipes et de plus il y a une dynamique différente en début/fin de semaine. On exclut aussi le weekend où l'on fonctionne seulement en deux *shifts* plus longs. L'analyse porte sur la somme des plaques en sorties des étapes sur chaque demi-heure d'une journée sans différentiation entre mardi, mercredi et jeudi. Le nombre de plaques et leur heure de sortie sont utilisés dans notre analyse car il s'agit de l'information la plus détaillée disponible dans l'usine. L'analyse mise en place vise à étudier le comportement de l'usine sur chaque demi-heure de la journée en réalisant un lissage pour pouvoir avoir le comportement de l'usine en général. On agrège par demi-heure les données ce qui permet d'identifier l'influence des changements d'équipe tout en limitant la volumétrie des données. Pour cela on réalise la moyenne par demi-heure des plaques en sorties des étapes ainsi que le premier et le troisième quartile normalisé afin de décrire la distribution des données. La formule de calcul des moyennes des plaques en sorties des étapes pour chaque demi-heure est :

$$\bar{m}(i, j) = \frac{\sum_{e=1}^{E_{i,j}} TrackOut(i, j, e)}{E_{i,j}}$$

où $i = \{1 \dots 48\}$ correspond à l'indice de chaque demi-heure d'une journée, $TrackOut$ correspond au nombre de plaques qui sont sorties pour la demi-heure i de la journée j sur l'étape e et $E_{i,j}$ le nombre d'étape agrégées pour la demi-heure i de la journée j .

Exemple 2.4. Pour comprendre la signification de notre $E_{i,j}$ on prend comme exemple le tableau à gauche correspondant à nos données historiques. Le tableau à droite correspond à notre agrégation. Pour la deuxième ligne de ce tableau l'agrégation se fait sur la première journée le 11/12 pour la première demi-heure de la journée cela donne $E_{1,1} = 3$.

La formule de calcul des moyennes des plaques en sorties des étapes pour chaque demi-heure :

$$\bar{m}(i) = \frac{\sum_{j=1}^n \bar{m}(i, j)}{n}$$

où n nombre de jours et $\bar{m}(i, j)$ est la moyenne pour chaque demi-heure i de la journée j .

etape	jour	heure	demi-heure	track_out	jour	heure	demi-heure	track_out
e_1	11/12	5	00	25				
e_2	11/12	5	00	25	11/12	3	00	75
e_3	11/12	5	00	25	11/12	3	30	25
e_4	11/12	5	30	25				

TABLE 2.3 – Exemple de l'agrégation des différentes étapes pour une date

Le premier quartile est la valeur au-dessous de laquelle se situent 25% des données ayant les plus petites valeurs. La formule pour calculer le premier quartile est :

$$Q1(i) = Q1(\bar{m}(i, j))$$

où $Q1$ est une fonction qui ordonne les plaques en sorties des étapes en fonction de la moyenne la plus petite jusqu'à la plus grande et retourne la valeur de l'index $\frac{(n \times 48) + 3}{4}$ avec n le nombre total de jours (Nombre de semaines \times 3).

Le troisième quartile est la valeur au-dessus de laquelle se situent 75% des données ayant les plus grandes valeurs. La formule pour calculer le troisième quartile est :

$$Q3(i) = Q3(\bar{m}(i, j))$$

où $Q3$ est une fonction qui ordonne les plaques en sorties des étapes en fonction de la moyenne la plus petite jusqu'à la plus grande et retourne la valeur de l'index $\frac{3(n \times 48) + 1}{4}$ avec n le nombre total de jours (Nombre de semaines \times 3).

Une étape de normalisation des séries est nécessaire pour s'affranchir des différences de volume de plaques selon les ateliers, on utilise pour cela la moyenne des plaques en sorties des étapes sur toutes les demi-heures :

$$\bar{m} = \frac{\sum_{j=1}^n \sum_{i=1}^p \bar{m}(i, j)}{p \times n}$$

avec $p = 48$, n nombre de jours et $\bar{m}(i, j)$ est la moyenne pour chaque demi-heure i de la journée j . On définit les séries normalisées suivantes :

$$\bar{m}_N(i) = \frac{\bar{m}(i)}{\bar{m}}$$

$$Q1_N(i) = \frac{Q1(i)}{\bar{m}}$$

$$Q3_N(i) = \frac{Q3(i)}{\bar{m}}$$

La figure 2.11 représente les trois fonctions calculées ci-dessus pour nos données sur 21 semaines et sur 3 jours de la semaine. On obtient donc $n = 21 \times 3$ jours pour les différentes formules ci-dessus.

On observe sur la figure 2.11 que malgré l'agrégation par moyenne du nombre de plaques en sorties des étapes les séries évoluent en fonction de la demi-heure considérée. Sur cette figure la courbe verte correspond au premier quartile, la bleue à la moyenne et la rouge au troisième quartile. La forte variabilité de l'usine est montrée sur cette figure. Certaines chutes de production peuvent être expliquées par le fonctionnement opérationnel de l'usine comme la chute à 5h, 12h30 et 20h30 qui correspondent aux changements d'équipes. Mais certaines chutes ne peuvent pas être expliquées. Cette analyse montre également qu'il n'est pas possible de caractériser de potentielles dérives temporelles au niveau global en raison de la forte variabilité de l'usine qui masque ce type de phénomène.

On se place maintenant à l'échelle de l'atelier (voir section 1.2) en utilisant les mêmes agrégations pour chaque atelier de l'usine. La figure 2.12 présente les résultats de l'analyse sur l'atelier de photolithographie car c'est l'un des ateliers les plus critiques dans l'usine, il est celui qui génère le plus de données et il est le plus pertinent pour identifier des dérives temporelles. Comme on peut le constater sur la figure la forte variabilité est présente sur l'atelier aussi. On remarque qu'il y a aussi les mêmes chutes de production à 5h, 12h30 et 20h30 pour les changements d'équipes mais aussi des chutes de production que l'on ne peut pas expliquer.

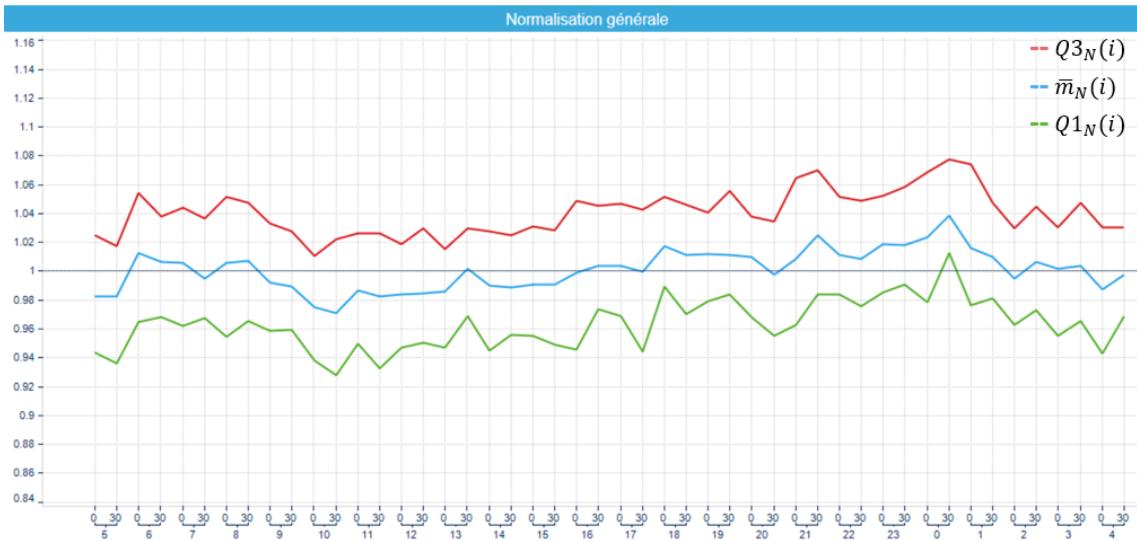


FIGURE 2.11 – Normalisation des plaques en sorties des étapes par demi-heure sur 21 semaines sur l’usine en général

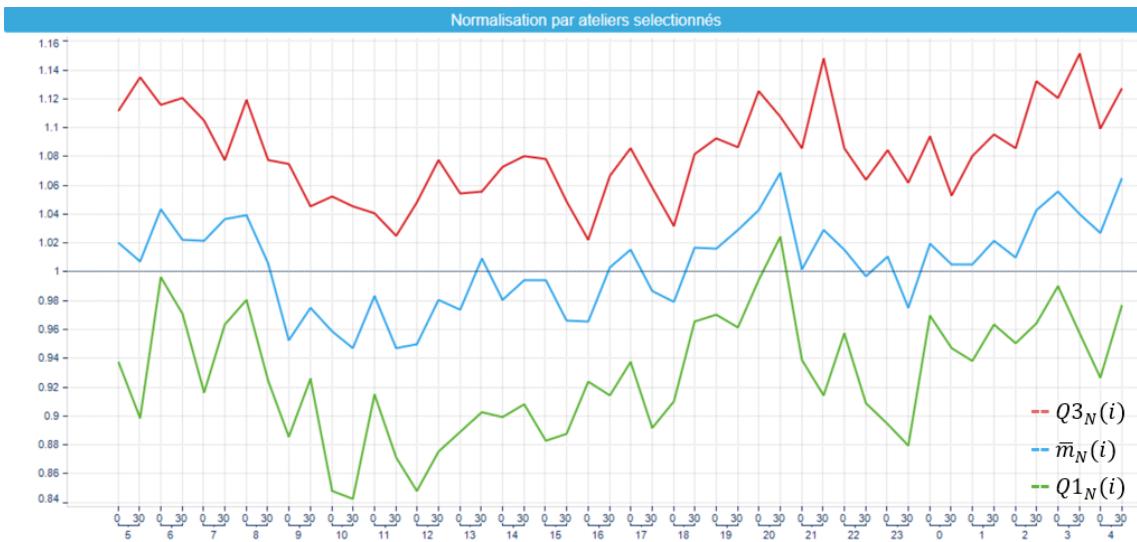


FIGURE 2.12 – Normalisation des plaques en sorties des étapes par demi-heure sur 21 semaines sur l’atelier photolithographie

Avec l’analyse effectuée au niveau des ateliers et sur la production au niveau de l’usine, on remarque que la variabilité reste trop importante pour envisager une caractérisation des dérives temporelles à ces niveaux. L’idée dans la suite de ces travaux est de ne plus se baser sur les ateliers qui sont trop généraux mais de se baser sur les opérations pour analyser les dérives temporelles. Cette granularité permet notamment d’utiliser les plans de planification pour les comparer à l’historique.

2.3 Détection des retards entre la production et la planification

La section 2.2 a présenté le problème de la variabilité dans la fabrication de semiconducteurs et montré l’un de ses effets sur la production effective de plaques dans l’usine au cours des semaines. L’étude de la section précédente montre en particulier que la variabilité est importante même si l’on considère seulement une analyse locale au niveau des ateliers. L’objectif de cette section est de proposer un autre point de vue moins global en présentant une méthode qui caractérise les dérives

temporelles non pas par atelier mais par opération d'une technologie. Pour ce faire la section présente une méthode de comparaison entre la planification et l'historique utilisant les données présentées dans la sous-section 2.1.2.

2.3.1 Comparaison cumulative entre la planification et l'historique

Deux fonctions ont été utilisées pour la comparaison Historique/planification. L'une des fonctions de comparaison est le cumul des plaques en sortie d'une étape en Historique/planification et l'autre le débit de production par rapport à Historique/planification. Les différentes fonctions sont calculées pour le quadruplet {technologie, brique, opération, étape} notée n autrement dit, n correspond ici à une étape spécifique dans une route spécifique pour la production de lots d'une technologie spécifique. La fonction cumul a pour principe le calcul de la somme des plaques en sortie d'une étape n du point de vue de la planification et de l'historique. Pour calculer la fonction cumul, on agrège les dates tous les quarts d'heure ce qui constitue notre pas d'échantillonnage.

Exemple 2.5. Par exemple pour le tableau 2.2 avec l'agrégation des dates tous les quarts d'heure donne le tableau 2.5.

n	STEP_OUT	Track_Qty	Echan	TYPE
$T_1b_1o_2e_3$	14/10/20- 06 :00 :00	0	06 :00-06 :15	
$T_1b_1o_2e_3$	14/10/20- 06 :23 :21	25	06 :15-06 :30	Histo
$T_1b_1o_2e_3$	14/10/20- 06 :30 :00	0	06 :30-06 :45	
$T_1b_1o_2e_3$	14/10/20- 06 :45 :00	0	06 :45-07 :00	
$T_1b_1o_2e_3$	14/10/20- 07 :10 :21	25	07 :00-07 :15	Histo

TABLE 2.4 – Exemple de l'agrégation des dates dans *TGVHistorique* de l'historique

La fonction cumul de production évaluée sur l'historique de lots est exprimée en fonction de $P_e^n(t)$ (où $P_e^n(t)$ est le nombre réel de plaques en sortie d'une étape n sur la période t). La fonction est définie sur l'ensemble des $T = \{1, \dots, T_{max}\}$ (où T est l'ensemble des pas d'échantillonnage). T_{max} est calculé par rapport au nombre de jours de la semaine dans l'historique, au nombre d'heures de la journée et au pas d'échantillonnage. Cela donne la formule $T_{max} = \text{Nbr Jours de la semaine} \times \text{Nbr d'heure} \times \text{Tous les quarts d'heure} = 7 \times 24 \times 4$.

On obtient la fonction $f_e^n : T \rightarrow \mathbb{N}$ $f_e^n(1) = P_e^n(1)$

$$f_e^n(t+1) = f_e^n(t) + P_e^n(t+1)$$

Exemple 2.6. Par exemple pour le tableau 2.4 avec l'agrégation des dates tous les quartes d'heure le calcul du cumul cela donne le tableau 2.5.

n	STEP_OUT	Track_Qty	Echan	TYPE	T	f_e^n
$T_1b_1o_2e_3$	14/10/20- 06 :00 :00	0	06 :00-06 :15		1	0
$T_1b_1o_2e_3$	14/10/20- 06 :23 :21	25	06 :15-06 :30	Histo	2	25
$T_1b_1o_2e_3$	14/10/20- 06 :30 :00	0	06 :30-06 :45		3	25
$T_1b_1o_2e_3$	14/10/20- 06 :45 :00	0	06 :45-07 :00		4	25
$T_1b_1o_2e_3$	14/10/20- 07 :10 :21	25	07 :00-07 :15	Histo	5	50

TABLE 2.5 – Exemple du cumul dans *TGVHistorique* de l'historique

De façon similaire, la fonction cumul de production planifiée de lots est exprimée en fonction de $P_p^n(t)$ (où $P_p^n(t)$ est le nombre de plaques en sortie d'une étape prévue par la planification). $f_p^n(1) = P_p^n(1)$

$$f_p^n(t+1) = f_p^n(t) + P_p^n(t+1)$$

La figure 2.13 représente les fonctions cumul pour la planification et pour l'historique pour une technologie issue d'une étape/opération/brique/route particulière. La représentation est faite sur les quarts d'heures à partir du début de la planification le 14/11/2020, l'horizon de comparaison dans notre exemple est sur 1 semaine.

La figure 2.13 représente les fonctions cumul pour la planification et pour l'historique :

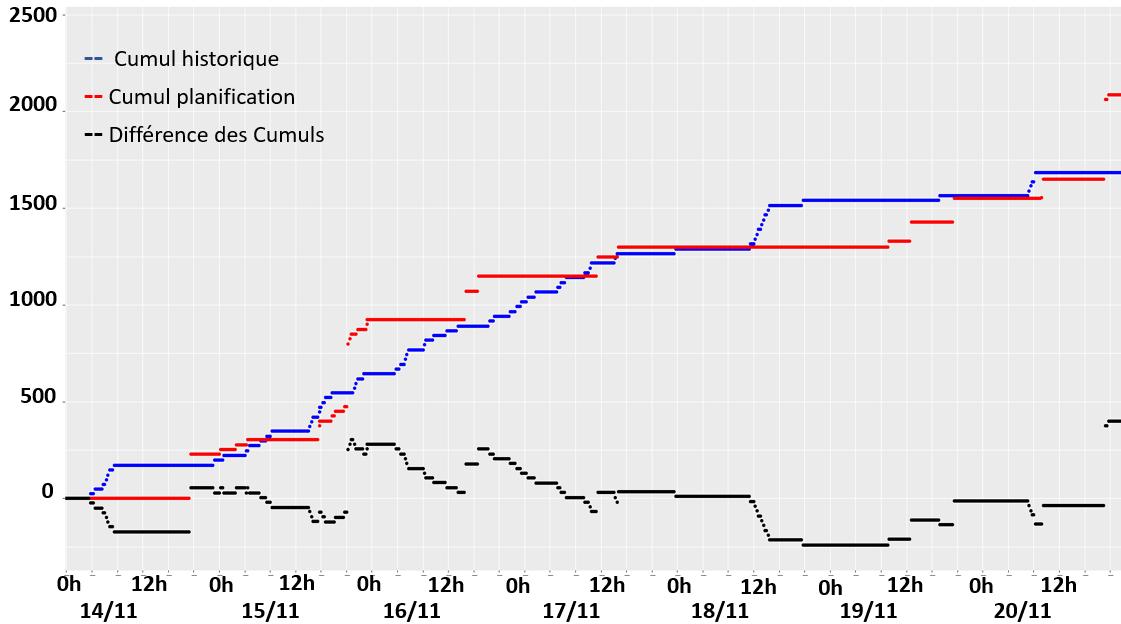


FIGURE 2.13 – Fonction cumul de la planification et de l'historique

- En rouge : fonction cumul de la planification.
- En bleu : fonction cumul de l'historique.
- En noir : la différence entre les cumuls de la planification et de l'historique.

La limite de la planification peut être observée avec des hausses sur une journée contrairement à l'historique dont la production au cours du temps est plus progressive.

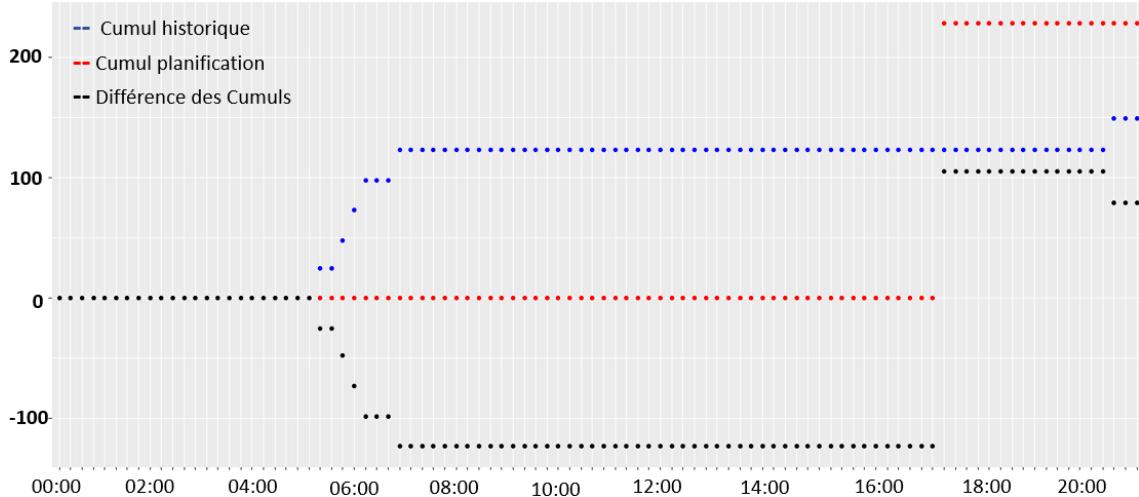


FIGURE 2.14 – Zoom sur la journée du 14/11/20 de la fonction cumul de la planification et de l'historique

La figure 2.14 est le zoom de la figure 2.13 sur la journée du 14/11/2020, on peut y constater que la courbe en bleu sur l'historique traduit une production linéaire. Alors que sur l'ensemble de la journée l'historique est en avance sur la planification en fin de journée la planification prévoit 250 plaques. Ce qui implique au final sur cette journée un retard de la production effective sur la planification.

Pour étudier le comportement à un instant donné de la planification/historique, on met également en place la fonction débit, qui permet de connaître le débit de production de la planification et de l'historique. Cette fonction est le débit de production effective sur une fenêtre glissante de $k \in T : d_e^n : T \setminus \{1, \dots, 7 \times 24 \times 4 - k\} \rightarrow \mathbb{N}$

$$d_{e,k}^n(t) = \frac{f_e^n(t+k) - f_e^n(t)}{k}$$

De la même façon on calcule le débit de production planifiée sur une fenêtre glissante de $k \in T$:
 $d_p^n : T \setminus \{1, \dots, 7 \times 24 \times 4 - k\} \rightarrow \mathbb{N}$

$$d_{p,k}^n(t) = \frac{f_p^n(t+k) - f_p^n(t)}{k}$$

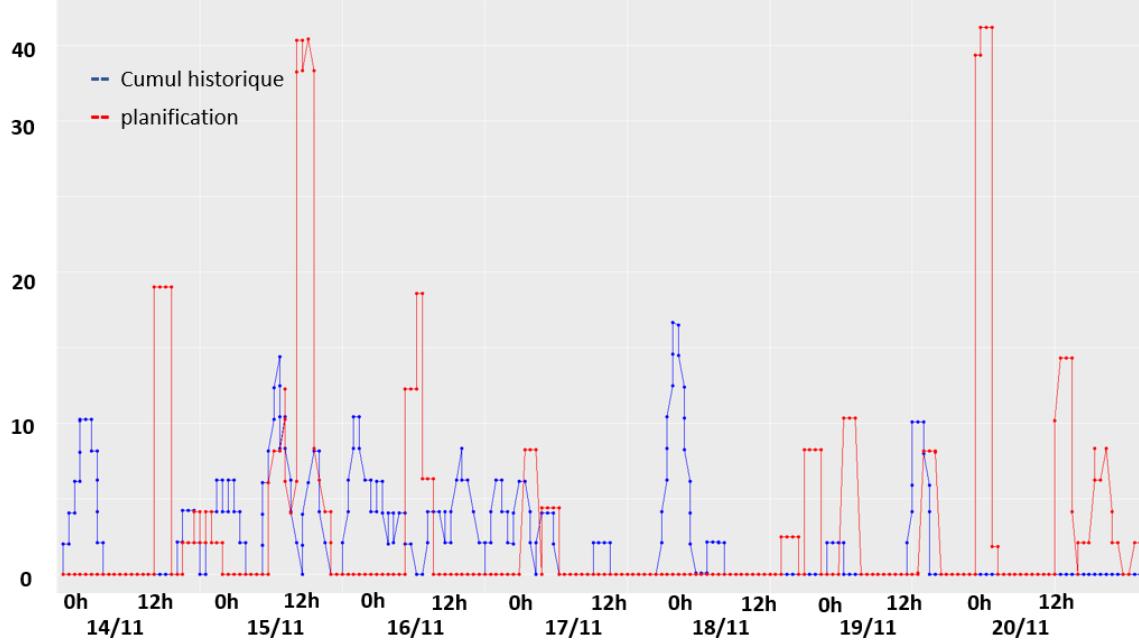


FIGURE 2.15 – Fonction débit de la planification et de l'historique pour une fenêtre glissante de 3 heures

La figure 2.15 représente les fonctions débit pour la planification et pour l'historique. La fonction débit de la planification est représentée par la courbe en rouge. La fonction débit de l'historique est représentée par la courbe en bleu. L'échantillonnage étant tous les quarts d'heure pour avoir une fenêtre glissante de 3 heures on considère 12 observations. Les courbes montrent que la production n'a pas un débit constant dans le temps que ce soit en planification ou en historique. La figure montre que le débit de la planification n'est pas linéaire, en effet on observe des pics plus importants que l'historique.

La courbe sur le cumul entre l'historique et la planification permet de savoir si notre historique est en retard par rapport à notre planification. La section suivante propose d'aller plus loin dans l'analyse de ce retard en étudiant ses conséquences sur les étapes ultérieures.

2.3.2 Indicateur du retard critique entre la planification et l'historique

Dans cette section, on présente un indicateur sur l'influence du retard accumulé entre la planification et l'historique. On veut savoir quand le retard accumulé sur l'étape n n'est structurellement plus rattrapable. Ce type de retard est dit *critique*. Pour une étape n nous allons déterminer le *travail* (Le travail est l'énergie fournie par cette étape) de production maximal de l'étape $n+1$ notée Tr_{max} . Considérons la figure 2.16, elle est constituée de deux étapes n et $n+1$. Imaginons que l'usine soit en mesure de produire un maximum de 100 plaques heure pour l'étape $n+1$. Si sur une heure donnée, on observe un retard de production de 125 plaques à l'étape n , ce retard ne pourra *jamais* être rattrapé dans l'étape $n+1$ qui ne peut que produire que 100 plaques sur l'heure et qui devient ainsi un goulot d'étranglement temporaire. Le retard de 125 plaques sur l'étape n est critique et peut se propager sur les étapes en aval de n . Par contre, si le retard en question n'est que de 25 plaques, on ne le considère pas critique car il peut être possible de le rattraper dans l'heure suivante.

L'objectif de l'analyse des retards critiques sur une étape n est d'observer au plus tôt des retards pouvant se propager sur une route avant qu'ils n'atteignent des étapes en goulot d'étranglement structurel et ainsi anticiper, voire éviter, une livraison de produits retardée.

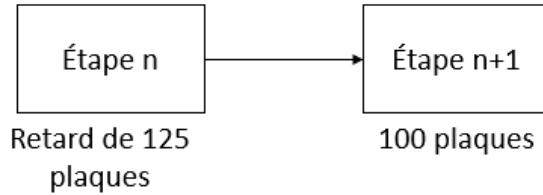


FIGURE 2.16 – Exemple de retard entre deux étapes

Pour calculer le travail de production de l'étape $n+1$, on utilise les fichiers d'entrées de OPERA sur la capacité des équipements et le fichier qui décompose les étapes/opérations/briques/routes pour les technologies. Le travail de production Tr_{max} , définit le nombre de plaques maximum produites pour l'étape $n+1$. Le fichier qui décompose les étapes/opérations/briques/routes permet d'avoir les caractéristiques de l'étape $n+1$ dont le temps de processus. La capacité des équipements permet de savoir combien d'équipements peuvent être utilisés à l'étape $n+1$ et quels types de lots passent sur ces équipements (par lot ou par *batch* c'est-à-dire plusieurs lots sur l'équipement) notée $NbrPlaque$. Le travail de production Tr_{max} est donnée par l'équation suivante :

$$Tr_{max} = CTTH \times NbrEquipement \times NbrPlaque$$

Exemple 2.7. soit l'étape $n+1$ suivant l'étape n , supposons que le temps de processus est $CTTH = 2.2h$ et que le nombre d'équipements qui permet de faire l'étape $n+1$ est de 1, $NbrEquipement = 1$. L'équipement traite lot par lot donc le nombre de plaques est de 25, $NbrPlaque = 25$. On obtient donc le travail de production de l'étape $n+1$, $Tr_{max} = 2.2 \times 1 \times 25 = 54.2$.

Calcul de la fusion planification/historique

Le calcul de l'indicateur repose sur la comparaison entre la planification et l'historique. Pour cette comparaison, on fusionne les tableaux de la planification et de l'historique. Le résultat de la fusion est un tableau constitué des colonnes suivantes :

- n : le descriptif d'une étape
- STEP_OUT_TIME : la date de sortie associée à cette étape
- TRACKOUT_QTY : le nombre de plaques associée à cette étape
- TYPE : cette étape est-elle issue du tableau de planification (OPERA) ou du tableau de l'historique (Histo)
- $DeltaW$ correspond au nombres de plaques d'une ligne du tableau qui est positif si la ligne vient de l'historique et négatif si elle vient de la planification.
- $DeltaH$ correspond au nombres de plaques d'une ligne du tableau de l'historique
- $DeltaO$ correspond au nombres de plaques d'une ligne du tableau de la planification

Dans l'algorithme 1, on va procéder en sélectionnant dans les tableaux les lignes associées à la même étape (te, b, o, n) (ligne 1). La fusion des tableaux de l'historique et de la planification est effectuée ligne 3. Les lignes ainsi insérées dans le tableau fusion associées au même type d'étape (te, b, o, n) sont ordonnées chronologiquement en s'appuyant sur la colonne STEP_OUT_TIME (ligne 4). Les colonnes $DeltaO$, $DeltaH$ et $DeltaW$ sont alors initialisées en fonction de leurs types (lignes 10-23). Le tableau 2.6 illustre le résultat d'une telle fusion pour une étape donnée (te, b, o, n) (notée n).

Algorithm 1: Algorithme de l'union des tableaux et du nombre de plaques en sortie d'une étape selon leurs types

Input : $tgvLotHistories \leftarrow$ ensemble des étapes dans l'historique
operaLotProjection \leftarrow ensemble des étapes dans la projection

Output:
 fusionOperaHisto \leftarrow ensemble des étapes de l'historique et de la projection synchronisées

```

1 for  $(te, b, o, n) \in \mathcal{S}$  do
2   //sélection d'une étape dans le tableau de planification et de l'historique
3   fusionOperaHisto  $\leftarrow tgvLotHistories + operaLotProjection$  (union des données entre la projection et l'historique)
4   fusionOperaHisto  $\leftarrow$  order(fusionOperaHisto par STEP_OUT_TIME) (fonction qui permet d'ordonner le tableau en fonction des dates de sortie de l'étape)
5   N  $\leftarrow$  Nombre de lignes du tableau fusionOperaHisto
6   fusionOperaHisto  $\leftarrow$  ajout de la colonne DeltaW
7   fusionOperaHisto  $\leftarrow$  ajout de la colonne DeltaH
8   fusionOperaHisto  $\leftarrow$  ajout de la colonne DeltaO
9   for  $i \in [0 \dots N]$  do
10    if  $Type(i) == "Histo"$  then
11      | DeltaW(i) = TRACKOUT_QTY(i)
12    else
13      | DeltaW(i) = - TRACKOUT_QTY(i)
14    end
15    if  $Type(i) == "Histo"$  then
16      | DeltaH(i) = TRACKOUT_QTY(i)
17    else
18      | DeltaH(i) = 0
19    end
20    if  $Type(i) == "Opera"$  then
21      | DeltaO(i) = TRACKOUT_QTY(i)
22    else
23      | DeltaO(i) = 0
24    end
25  end
26 end

```

Afin de disposer des données ainsi fusionnées réparties de façon homogène en fonction du temps, on va procéder à un échantillonage afin de rassembler et de cumuler dans chaque quart d'heure et sur le même couple (te, b, o, n) toutes les données issues de l'historique d'un côté et toutes les données issues de la planification de l'autre. Le résultat de cet échantillonage sur le tableau 2.6 est le tableau 2.7. Les dates sont omises pour des raisons de lisibilité ici. Par exemple, les 25 plaques sorties à 06:23:21 de l'historique sont incluses dans le créneau entre 06:15 et 06:30 (pas de cumul avec d'autres sorties puisque c'est la seule qui est produite sur le créneau).

Calcul de l'indicateur

Après avoir obtenu le tableau échantillonné, on l'enrichit avec le calcul du cumul de l'association historique/planification ($DeltaW$) puis on met en place une indication du retard critique détecté par rapport au travail de production de l'étape suivante ($n+1$). Le calcul de l'indicateur est donné par l'algorithme 2.

La ligne 6 calcule le cumul du retard entre la planification et l'historique. Avec le calcul du cumul du retard, on calcule la dernière date où le cumul a été positif (voir ligne 8-13 de l'algorithme) avec $LastPositive$ représentant cette date. Cette date est la dernière date à laquelle le cumul du retard est positif, autrement dit depuis cette date la production est en retard sur la planification. La ligne 14 correspond à la colonne "THretard" qui calcule le travail du retard accumulé, le travail du retard représente le retard en heure multiplié par le nombre de plaques en retard. Ce retard accumulé est la différence entre la date de la sortie de l'étape et $LastPositive$ la dernière date où le cumul est positif (en avance) multiplié par le cumul du retard. L'indicateur est calculé à la ligne

n	STEP_OUT	Track_Qty	TYPE	DeltaW	DeltaH	DeltaO
$T_1b_1o_2e_3$	14/10/20- 06 :23 :21	25	Histo	25	25	0
$T_1b_1o_2e_3$	14/10/20- 06 :39 :21	25	OPERA	-25	0	25
$T_1b_1o_2e_3$	14/10/20- 07 :10 :21	25	Histo	25	25	0
$T_1b_1o_2e_3$	14/10/20- 07 :46 :21	50	OPERA	-50	0	50

TABLE 2.6 – Tableau de l’union des tableaux et du nombre de plaques en sortie d’un étape pour la planification et pour l’historique

n	STEP_OUT	Track_Qty	TYPE	Echan	DeltaW	DeltaH	DeltaO
$T_1b_1o_2e_3$	05 :00 :00	0		05 :00 :00	0	0	0
$T_1b_1o_2e_3$	05 :15 :00	0		05 :15 :00	0	0	0
$T_1b_1o_2e_3$	05 :30 :00	0		05 :30 :00	0	0	0
$T_1b_1o_2e_3$	05 :45 :00	0		05 :45 :00	0	0	0
$T_1b_1o_2e_3$	05 :00 :00	0		06 :00 :00	0	0	0
$T_1b_1o_2e_3$	06 :23 :21	25	Histo	06 :15 :00	25	25	0
$T_1b_1o_2e_3$	06 :39 :21	25	OPERA	06 :30 :00	-25	0	25
$T_1b_1o_2e_3$	06 :45 :00	0		06 :45 :00	0	0	0
$T_1b_1o_2e_3$	07 :10 :21	25	Histo	07 :00 :00	25	25	0
$T_1b_1o_2e_3$	07 :15 :00	0		07 :15 :00	0	0	0
$T_1b_1o_2e_3$	07 :30 :00	0		07 :30 :00	0	0	0
$T_1b_1o_2e_3$	07 :46 :21	50	OPERA	07 :45 :00	-50	0	50

TABLE 2.7 – Tableau de l’union de la planification et de l’historique avec un échantillonnage d’un quart d’heure pour la journée du 14/10/20

15 qui représente la différence entre le travail du retard notée "THretard" de l’étape n et le travail maximal de l’étape $n + 1$ Tr_{max} . Si l’indicateur est négatif alors on considère que le retard est critique.

Le tableau 2.8 est un exemple de la sortie de l’algorithme 2, où seules les colonnes utilisées pour le calcul de l’indicateur sont présentées. Par exemple pour la dernière ligne du tableau on obtient $THretard(12) = (07 : 30 : 00 - 07 : 46 : 21) \times -25 = (16 \times 60 + 21) \times -25 = \frac{-981s}{60 \times 60} \times -25 = -0.27 \times -25 = 6.8$ et $Indicateur(12) = Tr_{max} - THretard(12) = 12 - 6.8 = 5.2$. Pour cette valeur de l’indicateur, le retard n’est pas considéré comme critique car le travail de production $n + 1$ permet d’effectuer le travail du retard.

On applique notre algorithme sur une étape donnée (te, b, o, n) , avec un $Tr_{max} = 54.3$ de l’étape $n + 1$ et on obtient la figure 2.17 sur la journée du 14/11/20.

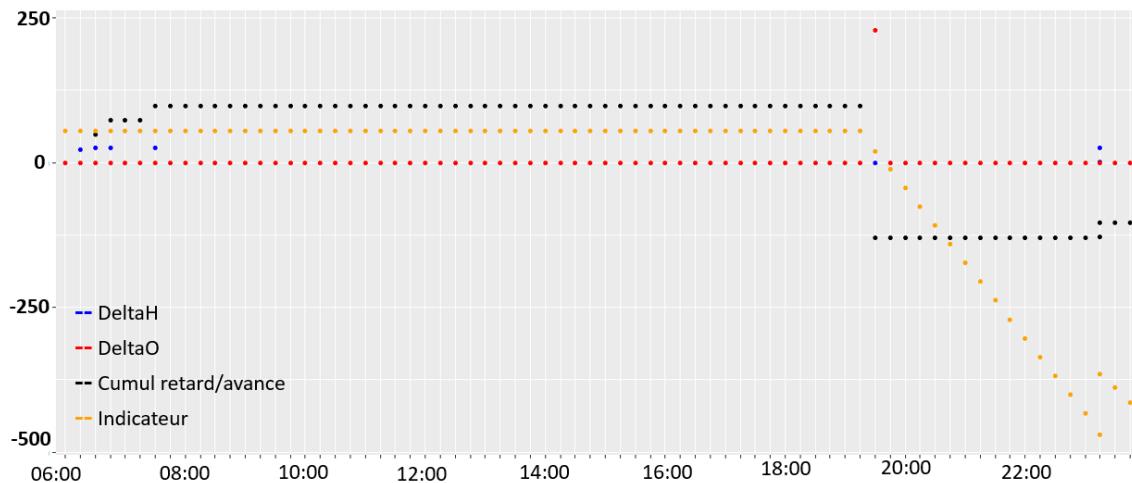


FIGURE 2.17 – Représentation graphique de notre indicateur pour la date du 14/11/20

Algorithm 2: Algorithme du calcul du l'indicateur

Input :

fusionOperaHistoEch \leftarrow ensemble échantillonné des étapes de l'historique et de la projection $Tr_{max} \leftarrow$ le travail de production de l'étape $n + 1$

Output:

fusionOperaHistoEch \leftarrow ensemble des étapes de l'historique et de la projection augmenté avec les indicateurs

```

1 N  $\leftarrow$  Nombre de lignes du tableau fusionOperaHistoEch
2 fusionOperaHistoEch  $\leftarrow$  ajout de la colonne CumulW
3 fusionOperaHistoEch  $\leftarrow$  ajout de la colonne LastPositive
4 fusionOperaHistoEch  $\leftarrow$  ajout de la colonne THretard
5 fusionOperaHistoEch  $\leftarrow$  ajout de la colonne Indicateur
6 for  $i \in /0 \dots N/$  do
7   | CumulW(i)  $\leftarrow$  CumulW(i) + DeltaW(i)
8   | if CumulW(i+1)  $> 0$  then
9     |   //Calcul de la dernière date où le cumul est positif
10    |   LastPositive(i+1)  $\leftarrow$  STEP_OUT_TIME(i+1)
11   | else
12   |   | LastPositive(i+1)  $\leftarrow$  LastPositive(i)
13   | end
14   | THretard(i)  $\leftarrow$  (LastPositive(i)-STEP_OUT_TIME(i))  $\times$  CumulW(i)
15   | Indicateur(i)  $\leftarrow$   $Tr_{max}$  - THretard(i)
16 end

```

Sur cette figure 2.17 les points en noir correspondent au cumul du retard, les points en bleu au nombre de plaques observé dans l'historique, les points en rouge au nombre de plaques prévu par la planification et les points en orange à l'indicateur calculé. La figure montre aussi le problème de la planification avec des pics de production visibles dans la section 2.3.1. Comme on peut observer pour la journée du 14/11/2020 du début de la journée jusqu'à 19h15, l'historique est en avance par rapport à la planification. A partir de 19h30, la planification prévoit 250 plaques ce qui implique que le cumul est négatif, la production est en retard. L'indicateur est toujours positif donc le retard n'est pas critique. Enfin, l'indicateur devient négatif à partir de 19h45, cela implique que le retard est critique. Le reste de la journée l'indicateur reste négatif malgré la production de 25 plaque à 23h15.

STEP_OUT	TYPE	Echan	DeltaW	CumulW	LastPositive	THretard	I
05:00:00		05:00:00	0	0	05:00:00	0	12
05:15:00		05:15:00	0	0	05:15:00	0	12
05:30:00		05:30:00	0	0	05:30:00	0	12
05:45:00		05:45:00	0	0	05:45:00	0	12
05:00:00		06:00:00	0	0	06:00:00	0	12
06:23:21	Histo	06:15:00	25	25	06:23:21	0	12
06:39:21	OPERA	06:30:00	-25	0	06:39:21	0	12
06:45:00		06:45:00	0	0	06:45:00	0	12
07:10:21	Histo	07:00:00	25	25	07:10:21	0	12
07:15:00		07:15:00	0	25	07:15:00	0	12
07:30:00		07:30:00	0	25	07:30:00	0	12
07:46:21	OPERA	07:45:00	-50	-25	07:30:00	6.8	5.2

TABLE 2.8 – Tableau correspondant aux colonnes calculées pour notre indicateur pour la journée du 14/10/20

2.4 Synthèse du chapitre

Ce chapitre a présenté le logiciel de planification l'organisation de la production, de la planification et les données utilisées.

La première section présente le fonctionnement de l'outil de planification et une comparaison des différents paramétrages possibles par l'outil de planification et comment ces paramétrages sont utilisés dans l'usine Crolles300. Cette section montre qu'il n'existe pas de paramétrage parfait pour faire de la planification. La variabilité des entrées pour la planification a une importance dans le fait que la planification est réalisable ou non réalisable.

Dans un second temps, la section présente une définition de la *variabilité* dans l'usine et comment caractériser une dérive temporelle de la production par rapport à la planification. Cette section montre que la forte *variabilité* de l'usine fait qu'il y a des variations importantes selon la configuration des encours et du temps de cycle des produits. Cela montre que la détection des dérives temporelles dans l'usine au niveau des ateliers n'est pas possible.

Dans un troisième temps, la section présente la méthode développée pour comparer la planification et l'historique par rapport à une opération. La méthode définit un indicateur d'effet du retard par rapport à l'étape suivante. Cette section montre les limites de la planification qui produit des pics ce qui complique la comparaison de la planification avec la réalité. Cette analyse montre que si l'on veut détecter des dérives temporelles, on doit se placer au niveau des opérations, pour mettre en place une détection. Pour réaliser cela différentes méthodes de diagnostic peuvent être utilisées. Un état de l'art sur le diagnostic de système est proposé dans la suite de nos travaux.

Chapitre 3

État de l'art du diagnostic dans les chaînes de production

Ce chapitre présente quelles sont les différentes méthodes de diagnostic dans la littérature en se focalisant en particulier sur leurs applications à des chaînes de production. Il se décompose de la façon suivante. La première section définit le problème général de diagnostic en définissant les différentes méthodes de diagnostic. Puis la deuxième section présente différentes méthodes de diagnostic qui sont appliquées dans des systèmes type chaînes de production.

3.1 Définition général du problème de diagnostic

Le diagnostic consiste à déterminer, pour un système donné, à partir des connaissances du système et des observations du système la localisation des défaillances et leurs causes. Selon la définition donnée dans [Zwi95] : *"Une défaillance est une altération ou une cessation de l'aptitude d'un ensemble à accomplir sa ou ses fonctions requises avec les performances définies dans les spécifications techniques"*. Une défaillance est ainsi une perte du fonctionnement du système. Les trois tâches essentielles du diagnostic sont la détection des défaillances, leur localisation et leur identification. La détection a pour principe de comparer la connaissance du système et une observation du système et d'indiquer quand l'observation et la connaissance du système ne sont plus en cohérence. La localisation a ensuite pour principe de pouvoir savoir où est la cause de la défaillance et enfin l'identification a pour principe de déterminer la nature de cette cause. Le processus de diagnostic peut être réalisé en mode en-ligne ou en mode hors-ligne. Le diagnostic en-ligne consiste en l'étude du système où les observations ne sont pas toutes connues *a priori*. Le diagnostic en-ligne sert généralement l'activité de supervision du système afin de détecter la défaillance au plus tôt pour traiter la défaillance le plus rapidement possible. Le temps de réponse pour le diagnostic en-ligne est contraint. Le diagnostic hors-ligne correspond à un diagnostic où l'on connaît l'ensemble des observations. Le diagnostic hors-ligne permet de détecter et de localiser les défaillances dans un délai fini.

Comme le montre la figure 3.1 plusieurs méthodes de diagnostic existent dans la littérature. Les méthodes de diagnostic sont séparées en trois grands ensembles, s'appuyant sur l'expérience (méthodes basées sur la connaissance), s'appuyant sur des historiques (méthodes orientées traitement de données) ou reposant sur des modèles estimés du comportement du système (méthodes à base de modèle).

3.1.1 Méthodes basées sur la connaissance

Les méthodes basées sur la connaissance sont définies sur un lien entre les effets de la défaillance et la défaillance dans un système. Ces méthodes essaient, d'après les causes de la défaillance, de trouver d'où vient la défaillance. La connaissance est acquise dans l'analyse du fonctionnement structurel et fonctionnel du système. Plusieurs méthodes existent pour connaître le fonctionnement structurel et fonctionnel comme les systèmes experts, l'Analyse des Modes de Défaillances de leurs Effets et de leurs Criticités (AMDEC) ou encore les arbres de défaillances.

Les systèmes experts définissent un ensemble de méthodes qui codifient des règles empiriques, des intuitions statistiques et de l'expérience. Un système expert est composé de trois parties : les



FIGURE 3.1 – Représentation des différentes méthodes de diagnostic

faits, les règles et un algorithme. Basés sur les faits et les règles, l'algorithme rend une expertise. Souvent les systèmes experts reposent sur une logique formelle.

Exemple 3.1. La représentation logique formelle d'une règle est notée par :

SI (A est vrai) ET (A implique B) ALORS (B est vrai)

où les propositions 'A est vrai' et 'A implique B' sont des faits et la règle permet de conclure que 'B est vrai'.

Cette approche codifie l'expérience humaine pour faire du diagnostic, cette expertise humaine va se baser sur la connaissance du système comme dans l'article [BS84] qui donne un exemple d'une approche basée sur l'expérience avec le système *MYCIN Experiments of the Stanford Heuristic Programming Project* pour des systèmes médicaux.

La méthode par AMDEC est utilisée comme outil en sûreté de fonctionnement et dans la gestion de qualité. Cette méthode est une analyse des modes de fonctionnements du système comme le montre la figure 3.2. Cette figure présente les différentes étapes de la mise en place d'un AMDEC. Pour un système donné il faut faire l'identification des modes de fonctionnements puis pour chaque mode de fonctionnement faire une identification des défaillances de ces modes et enfin faire une classification de la gravité. L'analyse des modes de fonctionnements et du niveau de gravité de la défaillance est effectuée sur la connaissance du système et sur l'expérience des défaillances déjà observées. Par exemple, l'article [SSS19] fait une analyse des modes de défaillances de l'AMDEC par rapport à des défaillances réelles pour améliorer la maintenance des transformateurs sur un réseau de distribution d'électricité en Inde.

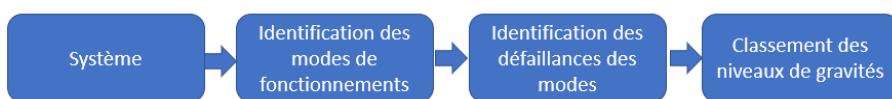


FIGURE 3.2 – Fonctionnement de l'AMDEC

La *méthode des arbres de défaillances* est une méthode d'étude de sécurité et de fiabilité des systèmes statiques. Un système statique est un système dont la défaillance ne dépend pas de l'ordre d'apparition des défaillances sur des composants du système. La méthode consiste en une représentation graphique des combinaisons possibles des événements qui peuvent traduire une défaillance, cette représentation met en évidence les relations de cause à effet dans un système. Grâce à un traitement mathématique il est possible de calculer la probabilité qu'une défaillance ait lieu. L'arbre de défaillances se base sur une analyse des modes de défaillances par AMDEC. La représentation est de haut vers le bas où le haut est la défaillance et chaque ligne correspond à un événement qui peut se réaliser. Les lignes sont reliées par les portes logiques "ou" et "et". Les principes du fonctionnement des arbres de défaillances sont expliqués dans l'article [Mor02] ainsi que la façon dont se met en place un arbre de défaillances dans un système.

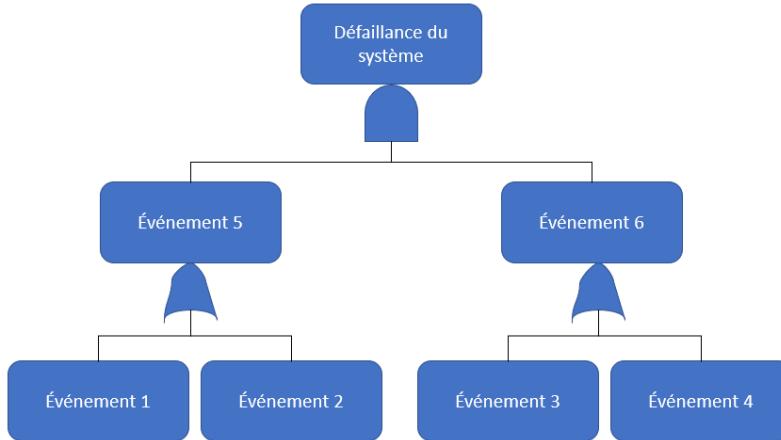


FIGURE 3.3 – Exemple d'un arbre de défaillances

Exemple 3.2. La figure 3.3 est un exemple d'arbre de défaillances. L'arbre est expliqué par le fait que si l'événement 1 ou 2 a lieu alors l'événement 5 a lieu. Si les événements 5 et 6 ont lieu alors la défaillance a lieu.

L'une des premières utilisations des arbres de défaillances a été faite au laboratoire Bell Telephone pour faciliter l'analyse des systèmes de contrôle de missile [W⁺61]. L'utilisation des arbres à défaillances va évoluer jusqu'à créer le logiciel *GALILEO* qui permet de faire des arbres de défaillances [SDC99].

3.1.2 Méthodes basées sur le traitement des données

Les méthodes basées sur le traitement des données sont des méthodes qui utilisent seulement des données issues de l'historique et qui par un traitement de données particulier vont permettre de faire une classification permettant de faire des classes spécifiques de fonctionnement du système. La figure 3.4 représente le fonctionnement d'une méthode de traitement des données. À partir des données de l'historique, ce type de méthode procède dans un premier temps à un prétraitement (par exemple, l'élimination du bruit possible dans les données à l'aide de techniques de filtrage) puis à la classification de ces données pour ainsi obtenir des classes de fonctionnement caractéristiques du système.



FIGURE 3.4 – Fonctionnement général d'une méthode basée sur le traitement des données

Plusieurs méthodes existent pour faire ce traitement des données et leur classification ou pour faire de la reconnaissance de forme comme présentées dans [Zwi95] et [Dub90]. Parmi elles, les méthodes de reconnaissance de forme sont des méthodes qui élaborent des algorithmes permettant de reconnaître des formes parmi des observations dans le but d'identifier l'état du système (fonctionnement normal, fonctionnement en défaillance, fonctionnement dégradé...). Une forme correspond à un motif qui est un ensemble de caractéristiques associées aux données étudiées, ces formes peuvent être numériques ou symboliques. Les méthodes de classification ont pour but d'identifier à partir des mesures du système les différentes classes des données. Les classes regroupent les données qui ont des caractéristiques similaires qui permettent de discriminer des comportements, des états du système (fonctionnement normal, fonctionnement en défaillance, fonctionnement dégradé...). De nombreuses méthodes permettent de faire de la reconnaissance de forme et d'associer les nouvelles formes obtenues en de nouvelles classes : citons par exemple les méthodes d'apprentissage automatique (*Machine Learning*) [FCdPFT19], modèles de Markov cachés [TS10], l'analyse statistique [NAB⁺12], ou les réseaux de neurones artificiels [Hop82].

L'apprentissage automatique est fondé sur les approches mathématiques et statistiques pour donner à un ordinateur la capacité d'analyse et d'apprentissage sur un grand nombre de données.

La méthode comporte deux phases, la première est d'élaborer une estimation d'un système à partir d'observations disponibles. L'estimation du système est élaboré par l'apprentissage du fonctionnement du système avec les données. La seconde phase est la mise en production de l'estimation du système et une amélioration de cette estimation avec de nouvelles données même si le modèle est en production. On distingue différentes classes d'algorithme d'apprentissage, en particulier, l'apprentissage supervisé (un expert va étiqueter des données brutes en rapport avec des propriétés prédefinies sur le comportement du système et l'apprentissage tente de caractériser ces propriétés à partir de ces données brutes observées) et l'apprentissage non supervisé (l'algorithme fait par lui-même un étiquetage des données brutes observées). L'analyse statistique est une méthode qui se base sur les données et sur des probabilités qu'un événement ait lieu. Les méthodes d'analyse statistique les plus utilisées reposent sur un modèle bayésien : elles calculent les probabilités qu'une cause arrive en fonction des événements observés. Ce type de méthode se base sur le théorème de Bayes (voir [Bay63]). Cette méthode donne une probabilité que la cause ait lieu entre 0 (la cause ne s'est jamais produite) et 1 (la cause s'est produite avec certitude). Cette approche est utilisée dans l'article [NAB¹²] qui utilise l'apprentissage automatique pour faire du diagnostic pour la consommation d'énergie d'une centrale de traitement de l'air. Les algorithmes proposés sont basés sur l'analyse du comportement observé du système et sa comparaison avec un ensemble de modèles de comportement générés à partir de diverses conditions de défaillances. La comparaison du modèle par connaissance automatique est un modèle bayésien. L'article [FCdPFT19] présente l'exploitation des méthodes d'apprentissage pour faire du diagnostic dans des accélérateurs de rayons lumineux.

Les modèles de Markov cachés sont des modèles statistiques qui se basent sur le processus markovien mais à paramètres inconnus. Le processus markovien est un processus stochastique (il représente l'évolution d'une variable aléatoire) possédant la propriété de Markov (l'information utile pour la prédiction du futur est entièrement contenue dans l'état présent du processus et qui n'est pas dépendant des états antérieurs). Dans le processus de Markov les transitions sont inconnues mais les états sont connus contrairement aux chaînes de Markov cachées dont les transitions sont inconnues ainsi que certains états. L'article [TS10] décrit une méthode de diagnostic sur des modèles multi-états (utilisée pour décrire des processus qui ne peuvent pas être représentés par des systèmes discrets). L'article est une revue des méthodes de diagnostic qui utilisent l'ajustement du modèle par les modèles de Markov à temps continu observés et les modèles de Markov cachés.

Les réseaux de neurones artificiels forment un système dont le fonctionnement correspond à un représentation ressemblant schématiquement à un réseau de neurones d'un cerveau humain. Les réseaux de neurones sont optimisés par des méthodes probabilistes. La figure 3.5 représente la

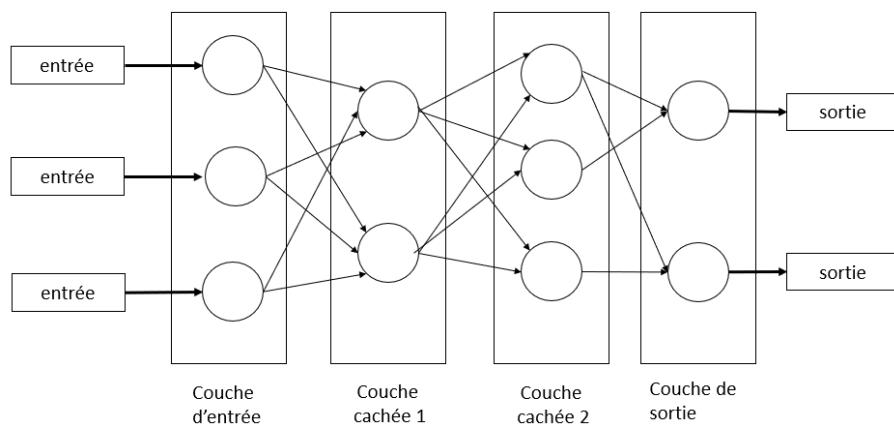


FIGURE 3.5 – Fonctionnement de réseau de neurones

structure d'un réseau de neurones. La première partie correspond aux données d'entrées qui sont disponibles pour un système. Les parties suivantes sont une succession de couches dites "cachées" qui peuvent être plusieurs à la suite et se terminent par une couche de sortie. L'article [Hop82] utilise les réseaux de neurones avec un algorithme pour l'évolution temporelle de l'état du système basé sur un traitement parallèle asynchrone. La méthode présentée dans [VMN²¹] utilise ce type de réseau de neurones pour analyser les images médicales d'un cerveau et permettre d'identifier la maladie de Parkinson.

3.1.3 Méthodes à base de modèle

Les méthodes à base de modèle sont des méthodes qui décrivent les systèmes et identifient les causes des défaillances. La description du système et les observations de ce système permettent la comparaison, ce qui permet de faire un diagnostic de détection de la défaillance et de la localisation de la défaillance. La figure 3.6 représente le fonctionnement du diagnostic à base de modèle. La connaissance du fonctionnement et les hypothèses de ce fonctionnement du système permettent de faire un modèle du système. La méthode de diagnostic utilise les observations du système et les observations attendues par le modèle pour faire une comparaison.

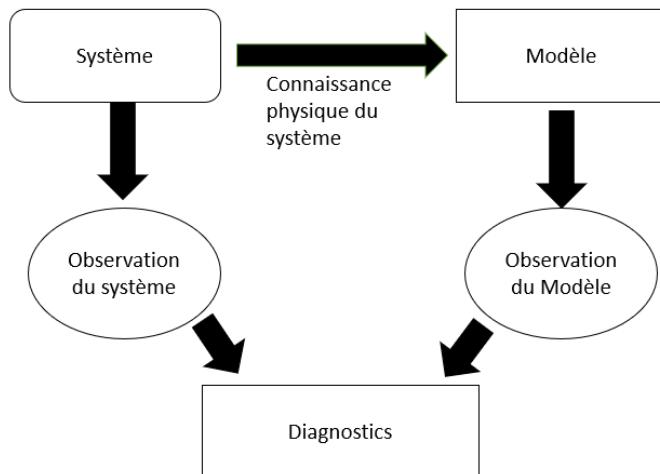


FIGURE 3.6 – Fonctionnement du diagnostic à base de modèle

Ce type de méthode s'appuie sur différents types de modèles dont la classification est donnée en figure 3.7. Les modèles de type quantitatifs sont majoritairement exploités dans la communauté scientifique de l'Automatique (méthodes de type FDI : *Failure Detection and Identification*) et les modèles qualitatifs sont initialement issus des communautés de l'Informatique, de l'Intelligence Artificielle (communauté DX) et des Systèmes à Événements Discrets (SED). Les différents modèles quantitatifs et qualitatifs sont détaillés dans l'article [Fra96].

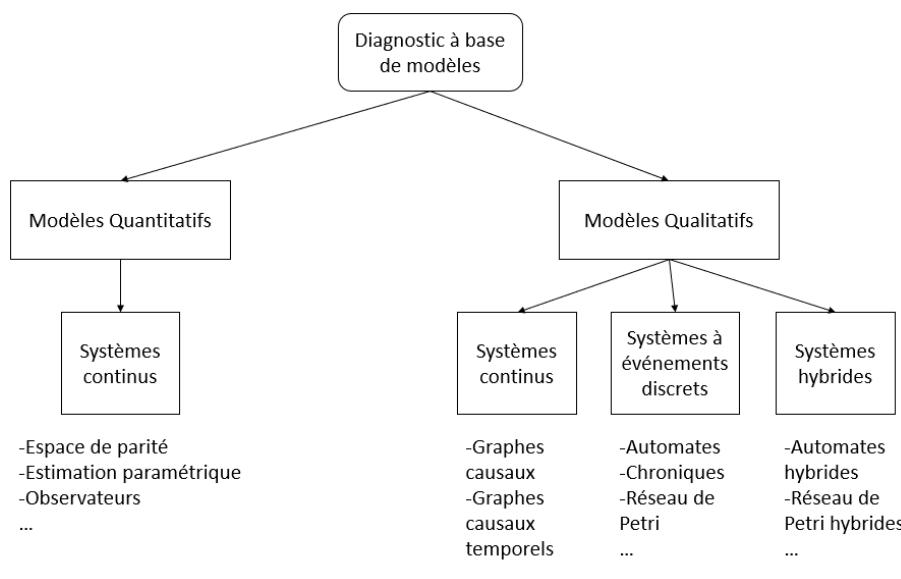


FIGURE 3.7 – Présentation des différentes méthodes à base de modèle.

Les modèles quantitatifs

Les modèles quantitatifs reposent sur la connaissance formelle du système et s'appliquent essentiellement sur des systèmes à dynamique continue. Les méthodes de diagnostic reposent sur des techniques d'estimation d'états ou d'estimation des paramètres. Elles exploitent des indications de fautes appelées *résidus*. D'après [Ger98] un résidu est associé à la comparaison des mesures issues de la simulation du modèle et des mesures réelles du système.

L'espace de parité permet de faire une modélisation du système par un modèle. L'espace de parité a pour objectif de vérifier la cohérence des modèles du système avec les entrées observées. Pour faire cela la méthode utilise les résidus. Le système sans défaillance se traduit par la valeur nulle de tous les résidus. L'espace de parité est constitué d'expressions analytiques obtenues par la redondance des fonctions analytiques entre l'entrée et la sortie du système. Par exemple, l'article [EW70] utilise les espaces de parité pour faire du diagnostic dans une expérimentation dans un ensemble de capteurs redondants. En général la méthode consiste à établir une redondance analytique (fonction exprimant la différence entre les informations obtenues par le système et celles obtenues par le modèle) entre les entrées et les sorties du système indépendamment des états du système.

L'estimation paramétrique a pour principe d'estimer les paramètres du système avec les informations des entrées et des sorties. L'estimation paramétrique utilise une approche analytique pour calculer les paramètres du modèle. Cette méthode compare les paramètres formels obtenus par la connaissance du système et les paramètres calculés par rapport aux informations des entrées et des sorties. Cette méthode est détaillée dans [Ise84] pour les systèmes linéaires et dans [Ise05] qui l'utilise sur des systèmes non-linéaires.

Les méthodes à base d'observateurs s'appuient sur l'estimation des états du système. Ce type de méthode consiste à effectuer la reconstruction des sorties non mesurables du système en utilisant la connaissance et les observations du système. Avec la reconstruction des états, le calcul de la sortie du système est possible. On compare alors la sortie observée par rapport à la sortie calculée par l'observateur pour faire du diagnostic. La figure 3.8 représente la structure d'un observateur avec le système. La représentation d'état du système est définie par les équations suivantes :

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t). \end{cases}$$

Où $x(t) \in \mathbb{R}^{n \times 1}$ représente l'état du système de dimension n , $u(t) \in \mathbb{R}^{p \times 1}$ représente l'entrée de dimension p , $y(t) \in \mathbb{R}^{q \times 1}$ représente la sortie du système de dimension q . Le système est décrit par les matrices $A \in \mathbb{R}^{n \times n}$ la matrice d'état, $B \in \mathbb{R}^{n \times p}$ la matrice de commande, $C \in \mathbb{R}^{n \times q}$ la matrice d'observation. L'observateur est défini par la représentation d'état :

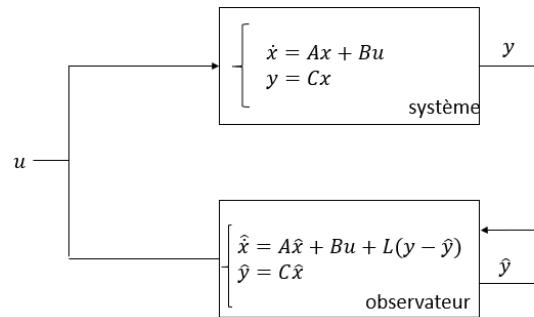


FIGURE 3.8 – Structure d'un observateur

$$\begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(y(t) - \hat{y}(t)), \\ \hat{y}(t) = C\hat{x}(t). \end{cases}$$

où $L \in \mathbb{R}^{n \times q}$ est la matrice de l'observateur. On peut trouver l'utilisation d'une telle technique pour le diagnostic de systèmes linéaires dans [MM94].

Les modèles qualitatifs

Les modèles qualitatifs permettent d'abstraire les modèles du comportement du système sans avoir à manipuler des équations mathématiques qui représentent le système mais uniquement des modèles symboliques. Selon les abstractions que vont prendre en compte le modèle sélectionné, une classification des modèles est possible : les modèles à temps continu, les modèles à temps discret et les modèles hybrides. Le diagnostic à base de modèle s'appuie alors en général sur la modélisation symbolique du système en fonctionnement nominal (c'est à dire sans défaillance dans le système) et sur une comparaison entre les observations du système et celles obtenues par ce modèle. Il existe également d'autres méthodes qui exploitent des modèles de système en défaillance et qui déterminent le mode de défaillance compatible avec les observations du système à partir de ces modèles.

Modèles pour les systèmes à temps continu Pour les systèmes à temps continu, les méthodes développées sont à base de graphes causaux [TMP97]. Les graphes causaux sont une représentation graphique des systèmes continus qui représentent les transferts dans un système. Une méthode de diagnostic utilisant les graphes causaux est décrite dans [SMB⁰⁶] pour faire de la détection et de l'isolation de défaillance dans des systèmes avec des grandes variations dans les mesures, les paramètres et les observations.

Modèles pour les systèmes à événements discrets atemporels Pour la mise en place du diagnostic dans les SEDs atemporels plusieurs méthodes existent et sont répertoriées dans cet état de l'art [ZL13]. Les différents formalismes utilisés pour le diagnostic de SED sont brièvement décrits ci-dessous.

L'un des premiers formalismes à avoir été utilisé pour le diagnostic à base de modèles dans les SEDs est celui des automates. Les automates sont des machines à états qui décrivent l'ensemble des évolutions possibles d'un système comme un ensemble de séquences d'événements discrets. Les automates sont une représentation graphique contenant un ensemble d'états et un ensemble de transitions qui modélisent le comportement du système.

Définition 3.1 (Automate à états fini). Un automate à états fini est une machine à états défini par :

$$G = \{Q, \Sigma, \delta, q_0, Q_m\}$$

où Q est l'ensemble fini des états, Σ est l'ensemble des événements, δ est la fonction de transition, $q_0 \in Q$ est l'état initial, $Q_m \subseteq Q$ est l'ensemble des états finaux.

Exemple 3.3. La figure 3.9 représente un automate avec deux états e_1 et e_2 ($Q = \{e_1, e_2\}$) avec deux événements t_1 et t_2 ($\Sigma = \{t_1, t_2\}$). Cela traduit le fait que si dans l'état e_1 l'événement t_1 a lieu alors l'état va changer pour l'état e_2 . L'état initial est e_1 et tous les états sont accepteurs pour le système ($Q_m = Q$).

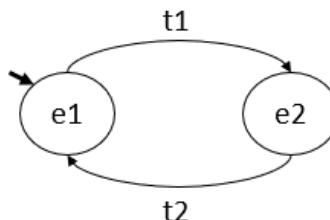


FIGURE 3.9 – Représentation d'un automate

L'une des premières méthodes de diagnostic exploitant ce formalisme est décrite dans la méthode extraite de [SSL⁹⁵]. Les comportements normaux et fautifs du système sont modélisés sous forme d'automates, ce qui permet ensuite de construire un diagnostiqueur afin de suivre et connaître l'état de ce système en fonction des observations. Avec une séquence d'observations on peut prendre une décision sur le diagnostic c'est-à-dire savoir si le système est en fonctionnement normal, si le système est en défaillance ou si le système est dans un état ambigu (l'état est ambigu si la décision n'est pas certaine, le système peut être en fonctionnement normal ou en défaillance). Cette méthode est en général utilisée pour faire du diagnostic en-ligne.

Exemple 3.4. La figure 3.10 représente un exemple de l'approche de [SSL⁺95]. La figure à gauche correspond à un automate où les transitions en traits pleins correspondent à des observations et les transitions en traits pointillés correspondent à une faute qui n'est pas observable. La figure à droite correspond au diagnostiqueur qui évolue entre l'état $1N$ qui correspond au système sans faute à l'état $4F_1$ qui correspond au système en faute. L'état $2F_1 3N$ correspond quant à lui à un état ambigu car on ne sait pas si on est dans l'état en faute $2F_1$ ou dans un état sans faute $3N$ à cet instant.

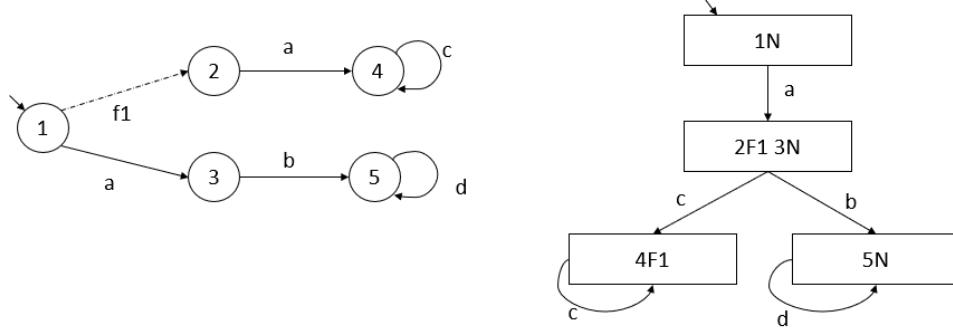


FIGURE 3.10 – Exemple de l'approche de Sampath

Les Réseaux de Petri (RdP) sont aussi utilisés pour faire une modélisation des systèmes et sont également exploitées en diagnostic. Les RdP sont une représentation graphique constituée d'un ensemble de places et d'un ensemble de transitions. Les RdP permettent de mieux représenter le parallélisme, la synchronisation et les ressources partagées que les automates. Plusieurs méthodes utilisant les RdP sont répertoriées dans l'état de l'art [Bas14].

Définition 3.2 (Réseau de Petri). Un réseau de Petri est décrit formellement par :

$$\mathcal{R} = \langle P, T, F, M_0 \rangle$$

- $P = \{p_1, p_2, \dots, p_m\}$ est l'ensemble des nœuds places
- $T = \{t_1, t_2, \dots, t_n\}$ est l'ensemble des nœuds transitions
- $P \cap T = \emptyset$
- $F \subseteq (P \times T) \cup (T \times P)$ est l'ensemble des arcs entre les nœuds places et transitions
- $M_0 : P \rightarrow \{0, 1, 2, \dots\}$ marquage initial des places

Exemple 3.5. La figure 3.11 représente un RdP avec deux places p_1, p_2 et trois transitions t_1, t_2 et t_3 . Le marquage initial M_0 est tel que $M_0(p_1) = 0, M_0(p_2) = 0$. La transition t_1 peut être tirée à tout instant. Si la transition t_1 est tirée une fois, alors un jeton va se positionner sur les places p_1 et p_2 (le nouveau marquage M_1 est $M_1(p_1) = 1, M_1(p_2) = 1$). Puis si la transition t_2 est tirée le jeton de la place p_1 disparaît (le nouveau marquage M_2 est $M_2(p_1) = 0, M_2(p_2) = 1$).

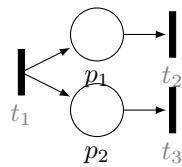


FIGURE 3.11 – Représentation d'un réseau de Petri

De nombreuses méthodes utilisent des RdP pour faire un diagnostic avec une approche similaire à celle du diagnostiqueur de Sampath [SSL⁺95], par exemple [UOO98][BCDT09] [CGS12]. Récemment, [AD21] propose une telle méthode en s'appuyant sur une technique de *Fourier–Motzkin* sur un RdP labelisé. [PS17] étend le problème de diagnostic de fautes simples à des motifs d'événements plus complexes dans les RdP labelisés et propose de résoudre le problème de diagnostic par vérification automatique de modèles.

Modèles pour les systèmes à événements discrets temporels Dans les SEDs qui ont été présentés jusqu'à présent, le temps est caractérisé par la séquence d'événements (un événement se produit après un événement). Il n'est nullement fait mention de la dimension quantitative du temps, des notions de durées et de dates. Le temps a une importance considérable car il peut être symptomatique d'un problème dans le système. Les différents formalismes utilisés pour le diagnostic de SED temporel sont brièvement décrits ci-dessous.

Les chroniques [DGG93] sont des représentations d'informations, à la fois temporelles et à la fois événementielles. Dans [DGG93] le système de reconnaissance utilise les informations d'événements temporisés pour identifier l'occurrence de chroniques dans un flux d'observations. Les chroniques sont utilisées dans le diagnostic car elles permettent de représenter des signatures temporelles causales [TCG90, TCG97] et ainsi d'identifier des fautes à partir de leurs effets observables [CD00]. Un système de diagnostic à base de chroniques est constitué de 4 modules, la modélisation (élaboration d'une base de chroniques basées sur l'expertise, la connaissance du système), la découverte de chroniques [SLCLL18] (élaboration de nouvelles chroniques avec l'utilisation des données, par apprentissage automatique), l'analyse (l'étude des différentes propriétés des chroniques) et la reconnaissance de chronique (le processus de diagnostic).

Exemple 3.6. La figure 3.12 est un exemple de la représentation d'une chronique. Un noeud correspond à un événement observé et les transitions correspondent aux contraintes temporelles. Si l'on observe les événements 1, 2 et 3 et que les contraintes temporelles entre ces événements sont vérifiées alors la chronique est reconnue.

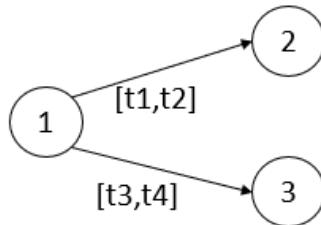


FIGURE 3.12 – Représentation graphique d'une chronique

Une application d'un système de chroniques pour le diagnostic de services web est décrite dans [PS09].

Les automates (Définition 3.1) peuvent être étendus à des automates temporisés, où les temporisations s'expriment sur les arcs entre les états à l'aide d'horloges. Les premiers travaux sur le diagnostic temporel basé sur les automates temporisés sont décrits dans [CL01]. Dans cette approche, le comportement du système est modélisé par un automate temporisé et le diagnostic est mis en œuvre par vérification automatique de modèle (*model-checking*) en exploitant l'outil Kronos [Yov97]. La méthode du diagnostiqueur de Sampath [SSL⁺95] a également été formellement étendue dans [Tri02] aux automates temporisés. Cette extension au temps affine les décisions sur le diagnostic en prenant en compte les observations datées. La méthode issue de [BHJ⁺21] propose une approche utilisant [Tri02] avec les marquages temporels comme un formalisme permettant de suivre l'évolution du système.

Les RdPs (Définition 3.2), peuvent être étendus à des RdPs temporels, où la temporisation des RdPs est généralement un intervalle situé sur les transitions du réseau. Dans [GTY09] la méthode de diagnostic utilise ce formalisme étendu. Elle repose sur la construction d'un estimateur à partir de l'analyse d'un réseau de Petri temporel, puis cette estimation est utilisée pour détecter les défaillances. [BCS14] calcule une estimation d'état pour le diagnostic de défauts d'un système modélisé en réseau de Petri temporel labélisé (RdP où chaque transition est labélisée par un événement observable ou non). L'approche [WMS15] utilise des RdPs temporels, et après le calcul de son graphe des classes, la méthode en extrait un graphe de diagnostic des défauts qui constitue le diagnostiqueur du système. La méthode dans [PSC21] utilise également les RdPs temporels mais étend le problème de diagnostic à des motifs plus complexes que de simples événements de faute. Le diagnostic est produit par vérification automatique de modèles (*model-checking*).

D'autres méthodes sur les SED sont utilisées pour faire du diagnostic comme les *Timed Failure Propagation Graphs* (TFPG) qui sont un formalisme utilisé dans l'industrie permettant de capturer la propagation temporelle des fautes dans les systèmes complexes, et pour soutenir d'importantes

activités d'exécution telles que le diagnostic et le pronostic. L'article [BCGM15] utilise cette modélisation pour faire de la détection et de l'identification et de la correction dans des systèmes critiques. Cette méthode utilise le diagnostic en-ligne pour faire de la correction des défaillances au plus vite pour éviter l'indisponibilité du système.

Modèles pour les systèmes hybrides Les systèmes hybrides intègrent les concepts du continu et du discret. Ils sont la combinaison entre des variables continues et des variables discrètes. Dans les systèmes hybrides les états discrets peuvent être vus comme continus sous certaines conditions. L'article [HAYHAL06] utilise une méthode hybride pour faire du diagnostic dans des systèmes informatiques. [CRV19] propose un logiciel HyMU (*Hybrid system Monitoring under Uncertainty*) pour simuler, diagnostiquer et pronostiquer les systèmes hybrides en utilisant des méthodes basées sur des modèles traitant plusieurs incertitudes (incertitudes sur la modélisation, sur les observations et sur les entrées...).

3.2 Diagnostic des chaînes de production

La section précédente présente les différents types de diagnostic qui peuvent exister en général. Le diagnostic fait l'objet de nombreux travaux. L'objectif dans cette section est de se concentrer sur les différentes méthodes de diagnostic utilisées dans les chaînes de production.

3.2.1 Diagnostic à base de connaissance

La section 3.1.1 présente les différentes méthodes de diagnostic qui sont utilisées pour le diagnostic à base de connaissance. Dans la littérature peu de travaux utilisent les méthodes à base de système expert pour faire du diagnostic des chaînes de production. L'article [PK93] propose une méthode de diagnostic par la conception de deux systèmes experts pour la fabrication de semiconducteurs. Le premier système expert est utilisé pour l'analyse des tests paramétriques dans une usine de plaques. Ce système expert contrôle le flux d'informations entre la base de connaissances et les données reçues des plaques. Le deuxième système expert est utilisé pour l'analyse des tests paramétriques dans une installation de plaques. Les deux systèmes permettent de faire un diagnostic sur les plaques en les comparant. La limite de cette méthode est l'utilisation des systèmes d'expertise pour toutes les spécificités des plaques dans la production. [LHK13] définit également une méthode de détection de fautes sur des équipements par système expert. La connaissance dans ce système expert est obtenue par une description en UML (*Unified Modelling Language* : l'UML est un langage de modélisation graphique à base de pictogramme pour aider à la visualisation d'un développement logique et du concept orienté objet). Les réseaux bayésiens sont ensuite utilisés comme sources de fautes pour le diagnostic, avec les différentes causes, la probabilité que ces fautes arrivent et les solutions proposées quand elles arrivent.

Les articles [YLLH06] et [BAEMB19] détaillent des méthodes de diagnostic de chaînes de production utilisant les analyses AMDEC. L'article [YLLH06] définit une AMDEC pour le diagnostic en utilisant des données d'une usine de semiconducteurs à Taiwan. L'objectif est d'augmenter le taux d'activation/utilisation des équipements, améliorer la vitesse et la valeur des flux, réduire les délais et les coûts de manière efficace en réaction au raccourcissement du cycle de vie des produits. L'objectif de cette étude est d'établir un mécanisme d'évaluation et d'amélioration systématique pour localiser le numéro de priorité du risque des éléments à partir de l'AMDEC. L'article [BAEMB19] présente une méthode de diagnostic et de prédiction utilisant la méthode AMDEC pour connaître les causes de défaillance dans des chaînes de production dans l'automobile. La méthode de détection utilise également un réseau bayésien pour modéliser des probabilités sur une défaillance définie par la méthode AMDEC.

Dans la littérature il existe également peu de travaux utilisant les méthodes à base d'arbres de défaillance pour faire du diagnostic des chaînes de production. [SCC06] présente une méthode utilisant les arbres de défaillance pour une chaîne de production de circuits électroniques. La méthode propose de comparer les techniques existantes utilisant les méthodes d'arbres de défaillance. Les résultats montrent que la méthode proposée peut estimer l'intervalle de défaillance de manière plus flexible que les méthodes précédentes et aider les décideurs à trouver les composants les plus critiques du système pour améliorer la fiabilité.

3.2.2 Diagnostic à base de données

Dans la littérature plusieurs travaux utilisent les méthodes à base d'apprentissage automatique pour faire du diagnostic des chaînes de production.

L'apprentissage avec des algorithmes mathématiques est utilisé pour faire du diagnostic dans les chaînes de production comme le montre l'article [CBYP18] qui propose une méthode de détection des causes de défaillance dans un équipement en fonction des données de l'équipement. L'article propose une approche pour détecter les causes profondes de l'inadéquation équipement/chambre en temps réel en exploitant toutes les données disponibles, telles que les mesures du produit, les lectures des capteurs de l'équipement et les données de maintenance. La méthode utilise une analyse multivariable pour faire de la détection de fautes sur des paramètres des équipements. L'analyse multivariable s'effectue par une étude de l'analyse de variance (l'analyse de variance fait partie d'un ensemble d'analyses et de modèles statistiques). Une autre méthode de diagnostic par un algorithme statistique est défini dans [OKT⁺09]. Elle utilise un algorithme de détection et d'identification des groupes de défaut sur une plaque dans les ateliers de défectivité (vérification de défaut des plaques). L'algorithme est découpé en trois étapes distinctes : détection, découpage et reconnaissance. L'article propose l'utilisation de statistiques de comptage conjoint pour effectuer la détection des groupes de défaut qui sont définis par rapport à l'expertise humaine.

Des méthodes de diagnostic basées apprentissage automatique sont également utilisées pour d'autres applications. [IRA21] présente une méthode de détection de défaillance des plaques dans la production des semiconducteurs. L'article présente une revue de la littérature des méthodes de classification et le choix de classification utilisant une machine à vecteurs de support à une classe et dont l'apprentissage est effectué à partir de mesures obtenues sur des lots. Dans [CLI21] une méthode d'apprentissage automatique est proposée en métrologie (atelier de mesure de plaques). Cette méthode s'appuie sur des polynômes de Zernike pour préparer et nettoyer les données. Puis, les données pré-traitées sont utilisées pour alimenter un algorithme d'apprentissage automatique non supervisé basé sur la densité pour faire la détection des données anormales (*Random Forest Discriminant Analysis*).

Concernant les réseaux de neurones, il existe plusieurs travaux sur le diagnostic des chaînes de production qui les utilisent. Par exemple [CWL20] propose une méthode de classification et d'inspection des défauts sur des plaques utilisant l'apprentissage automatique. Cet article présente une méthode d'apprentissage automatique basée sur la vision pour classer les défauts de surface visibles sur les plaques de semiconducteurs. La méthode proposée utilise des réseaux neurones convolutifs d'apprentissage profond pour identifier et classer quatre types de défauts de surface : centré, local, aléatoire et autour de la plaque. Des expériences ont été réalisées pour déterminer sa précision, les résultats expérimentaux ont montré que cette méthode peut atteindre une précision supérieure de l'ordre de 98% à 99%. Une autre méthode d'apprentissage avec des réseaux de neurones convolutifs est utilisée dans [WGC⁺20] qui détecte et classe les défauts sur une plaque. Cet article propose une nouvelle méthode basée sur des réseaux neurones convolutifs pour l'inspection des défauts des plaques de semiconducteurs. Dans un premier temps, il faut faire une caractérisation des plaques avec le réseau de neurones pour générer des cartes de cartographies des défauts sur une plaque, puis avec les cartes des caractéristiques des plaques on classe les défauts en trois différentes branches. Les résultats expérimentaux montrent que la méthode proposée donne de bonnes performances globales avec une précision moyenne de 93% à 97%. [YJXZ21] présente une méthode sur la détection de fautes sur un capteur utilisant un réseau de neurones avec une dynamique de *time wrapping loss*. Le modèle de réseau de neurones est à structure auto-encodeur (c'est un réseau de neurones qui utilise l'apprentissage non supervisé), il est proposé pour améliorer la précision de la détection des anomalies et enfin pour éliminer la cause des défauts et ainsi réduire la perte de rendement. Les résultats expérimentaux sur un ensemble de données de capteurs révèlent une bonne performance de l'enveloppement temporel dynamique par rapport aux modèles formés avec la fonction de perte standard (*Mean Squared Error*). Un travail similaire [YXZ20] propose une méthode de détection de fautes sur un capteur utilisant un réseau de neurones avec la structure auto-encodeur. Cette méthode est utilisée pour reconstruire les différentes capteurs d'équipements. Le réseau structuré peut être appliquée à différentes étapes de traitement, ce qui permet à cette méthode d'être acceptée comme une méthode standard de traitement des données. L'article [HL21] présente une méthode sur la détection en utilisant les séries temporelles pour faire le diagnostic. Les séries temporelles correspondent aux données des capteurs. L'article propose un modèle de réseau de neurones convolutif avec des séries temporelles multiples pour la détection et le diagnostic des défauts dans la fabrication des semiconducteurs. Les informations sur les capteurs peuvent être identifiées par la

couche du diagnostic dans le réseau de neurones convolutif.

L'article [JKHH10] décrit une méthode utilisant un modèle de Markov caché étendu pour résoudre les problèmes de détection de transition dans les processus de gravure par plasma. Le modèle de Markov caché proposé est une combinaison de plusieurs modèles : la combinaison d'un modèle semi-Markov et un modèle segmentaire. Le modèle segmentaire permet une modélisation adaptable des données par exemple par des fonctions de régressions linéaires. Le modèle semi-Markov permet de faire une modélisation avec les connaissances avec les algorithmes d'apprentissage et de déduction. Le diagnostic s'effectue par la comparaison des deux modèles pour faire de la détection des défaillances dans le système.

D'autres méthodes de diagnostic par utilisation de données existent dans la littérature comme le diagnostic par internet (le e-diagnostic) comme le montre l'article [HY05]. La méthode consiste à faire du diagnostic sur des machines de semiconducteurs puis à faire une connexion avec des experts à distance des équipements de l'usine pour récolter des informations. Avec celles-ci, un diagnostic à distance est possible.

3.2.3 Diagnostic à base de modèle

Cette section présente un panel d'exemples d'application à des chaînes de production des divers types de méthodes de diagnostic à base de modèle.

Modèles quantitatifs Les modèles quantitatifs sont exploités à de multiples reprises sur des applications liées à des chaînes de production. [MS93] met en place une méthodologie du diagnostic automatisé des équipements d'une usine de semiconducteurs. L'approche proposée combine l'aspect d'un algorithme sur un modèle quantitatif avec la connaissance des équipements de l'usine. Grâce à l'application de techniques de modèles quantitatifs il est possible de déduire les causes des défauts, et en utilisant des preuves générées avant, pendant et après le fonctionnement de l'équipement, les dysfonctionnements peuvent être identifiés avant qu'une erreur de traitement importante ne se produise. Cette méthode d'identification de défauts a été appliquée sur un atelier de 29 équipements *Etch* (type de gravure) usinant une seule plaque à la fois. Ce choix des équipements *Etch* est lié à la complexité de détecter des défaillances dans ce type d'équipement.

L'article [PC94] propose une revue des approches de diagnostic exploitées dans le diagnostic de défaillance de moteur de l'aéronautique en utilisant les espaces de parités. La problématique de la robustesse et de l'isolation dans le diagnostic est au centre de l'article. L'exemple concerne un système de moteur à réaction dans l'aéronautique.

[DMBDT07] propose un système de diagnostic pour la détection et l'identification des défaillances sur banc d'essai en électromécanique. Cette approche consiste à générer des indicateurs de défaillances et de seuils qui sont représentés sous la forme de graphes causaux. Les indicateurs sont définis par l'analyse de résidus en présence d'incertitudes sur les paramètres. Les différents paramètres sont trois imperfections mécaniques (frictions visqueuses, jeu, et liaison élastique).

[GDI02] propose une approche de diagnostic pour l'identification des défaillances dans le contexte d'un réacteur chimique (système non linéaire). Le problème est ici divisé en plusieurs modèles linéaires et la méthode s'appuie sur des estimations paramétriques.

Une méthode de diagnostic par observateurs dans des chaînes de production est proposée dans [JPP+19]. Cette technique de détection et d'isolation des défaillances se base sur un modèle pour le diagnostic des conditions de santé des équipements de production. L'approche propose une identification des défaillances en extrayant des informations du signal des défaillances, telles que l'amplitude ou la forme de la défaillance, et la comparaison de cette défaillance avec les signaux obtenus par l'utilisation d'un observateur sur le système.

Modèles qualitatifs Dans le cadre des chaînes de production, le diagnostic à base de modèles qualitatifs a également été largement exploité. [SLL93] définit une méthode de contrôle et de diagnostic dans le contexte de la production de semiconducteurs. La méthode de contrôle traite un atelier de photolithographie et exploite une simulation basée sur la physique du système pour faire du diagnostic de santé des équipements de l'atelier. La décision de diagnostic s'appuie sur un modèle statistique multivarié qui caractérise des modes de dégradation par des seuils sur des variables mesurées. Ce type de techniques peut être adaptée à des systèmes non-linéaires en exploitant plusieurs modèles statistiques à la fois (version multi-modèles).

[MCDB10] utilise une méthode de diagnostic qui combine les chroniques et une analyse par arbre de défaillances pour la supervision d'une chaîne de production dans le conditionnement du lait. Cette chaîne de production associe une chaîne automatique et une partie humaine. L'approche utilisée fournit un diagnostic en-ligne pour assurer la qualité des produits. Le modèle de surveillance est une hiérarchie de machines à états qui déterminent le comportement du système et de ses sous-systèmes, ces comportements sont définis par apprentissage de chroniques. Le diagnostic s'effectue par comparaison des observations du système et les observations attendues par le modèle. L'arbre de défaillances est construit en fonction des observations du système et permet de faire une analyse des causes des défaillances dans la chaîne de production.

[PMGR13] présente une méthode de décentralisation du diagnostic des systèmes à événements discrets qui s'appuie sur la décomposition d'un modèle d'une usine en plusieurs modèles (automates atemporels). Cette décomposition est effectuée en utilisant des techniques de vérification automatique de modèles (*Model-Checking*). Cette structure décentralisée est utilisée pour éviter le trop grand nombre d'états pour représenter le système. Avec la connaissance du système et la modélisation des différents modèles la méthode construit les comportements des modèles avec faute.

La représentation des modèles par des automates temporisés est exploitée par exemple dans [SLPQ06]. Ces travaux modélisent le comportement d'un actionneur DAMADICS (*Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems*) en automates temporisés. L'automate modélise les états de l'actionneur ainsi les durées de changement entre états. Basé sur cet automate, l'algorithme de diagnostic proposé fait une détection de délais anormaux entre états qui conduit à une identification de défauts actionneur (pression, débit, ouverture de vannes) à partir des observations de l'entrée et de la sortie du système.

Les réseaux de Petri sont aussi utilisés dans le diagnostic des chaînes de production. À titre d'exemple, [CGPS11] présente une approche du diagnostic basée sur les réseaux de Petri labélisés. L'approche est basée sur la notion de marquage en utilisant l'ensemble des marquages accessibles (graphe représentant l'ensemble des marquages que l'on peut atteindre selon le marquage initial et selon les transitions tirées) pour faire le diagnostic. Dans le cas de la modélisation du système, le modèle se fait sur le système en faute. Le principe du diagnostic est, grâce à l'ensemble des marquages accessibles, de savoir selon les observateurs si le système est en état de faute, en état incertain ou en état normal. L'application décrite dans [CGPS11] est constituée de 4 équipements et de 4 robots qui font le déplacement entre les équipements. Le problème de cette méthode est la complexité de faire du calcul de l'ensemble des marquages accessibles sur des systèmes complexes avec plusieurs équipements (explosion combinatoire). [BB19] présente une méthode de diagnostic des systèmes industriels hybrides exploitant des RdPs hybrides. La méthode utilise un algorithme qui traite les différentes étapes d'un système hybride de la partie discrète et de la partie continue. La méthode définit deux types de diagnostiqueur, l'un pour détecter les fautes discrètes et un autre diagnostiqueur pour détecter les fautes sur la dynamique continue.

[KL15] utilise l'algèbre ($\max, +$) (voir chapitres 4 et 5 pour une présentation détaillée) pour faire de la détection de retard sur les plaques lors des changements de chambres dans un équipement de production d'une usine de semiconducteurs. Le contrôle par retour d'état, servant à contrôler la variabilité du temps de traitement, est utilisé pour faire une détection du retard de l'arrivée des plaques sur les équipements. La méthode proposée utilise des graphes d'événements temporisés et modélise deux fonctionnements : un équipement à 4 chambres associé à 1 robot de transport et un équipement à 4 chambres associé 2 robots de transport. Le retour d'état est modélisé par une équation de l'algèbre ($\max, +$).

3.3 Synthèse du chapitre

Ce chapitre a présenté les différentes méthodes de diagnostic et une application de ces méthodes sur des chaînes de production. Pour répondre à notre problématique, les analyses faites dans le chapitre précédent montre qu'un diagnostic global au niveau usine ou même par ateliers ne permet pas de faire une identification des retards entre la planification et l'historique du fait de l'importante variabilité dans la production. Pour faire une identification du retard nous nous plaçons donc au niveau des opérations, en comparant le nombre de plaques produites en historique par rapport au nombre de plaques planifié. Le type de défaillance que l'on va donc chercher à détecter et à localiser dans notre méthode est un retard qui caractérise les dérives temporelles dans la production de lots d'une même technologie au niveau des opérations.

Parmi les méthodes de diagnostic qui ont été détaillées dans ce chapitre, les méthodes de diagnostic à base de modèles semblent les plus adaptés pour répondre à notre problématique. Notre objectif est de définir le comportement des opérations à l'aide de graphes d'événements temporisés. La méthode proposée dans la suite de nos travaux se base donc sur ce type de représentation et exploite l'algèbre $(\max, +)$ pour faire une comparaison le comportement attendu par notre modèle et les observations réelles issues de la production en usine.

Chapitre 4

Modélisation du problème

Le chapitre 2 montre que par la forte variabilité de l'usine, la détection de retard sur l'ensemble de l'usine ou sur des ateliers n'est pas possible. Nous proposons de définir ici le problème de détection de retards sur la production d'une technologie particulière de semiconducteurs à travers une modélisation formelle des notions de routes, opérations, étapes. Le retard pour une étape est le fait que les dates de production des plaques sont différentes des dates planifiées. Par exemple, pour une opération donnée, la planification indique que 50 plaques doivent être produites en une heure. Or dans l'historique, la production des 50 plaques est réalisée en deux heures. Il y a donc un retard de 1 heure entre la planification et la production réelle. Ce retard est une dérive temporelle qu'on cherche à détecter et à en localiser la source au niveau d'une étape. Il s'agit du type de défaillance pris en compte dans nos travaux.

Nos travaux utilisent une méthode de diagnostic à base de modèles (voir section 3.1.3). La modélisation que nous proposons s'appuie sur les informations décrivant les étapes de production qui sont détaillées dans l'outil de planification OPERA. La réalisation d'une opération (ensemble d'étapes) sur un lot donné dans l'usine va produire un ensemble d'événements datés pour chaque étape (entrées/sorties du lot dans chaque équipement, début/fin de transport du lot entre équipements, stockage....). Certains de ces événements sont observables tout au long de l'opération et sont enregistrés dans des fichiers tels que le fichier TGVHistorique (voir table 2.2, page 22 de la sous-section 2.1.2). On a notamment accès à l'événement STEP_OUT, renommé plus simplement StepOut à partir de maintenant, correspondant à la fin de l'étape. Cet événement est également présent dans les données de la planification (fichier LOTSPROJECTION, voir table 2.1, page 22 de la sous-section 2.1.2).

Nous proposons de modéliser ces opérations comme des systèmes à événements discrets temporels partiellement observés via des Graphes d'Événements Temporisés (GET), qui sont une sous-classe des réseaux de Petri (définition 3.2). Le principe général de notre approche est illustré sur la figure 4.1. L'idée est de comparer les observations réelles (StepOut historique) issues de la réalisation d'une étape avec les événements attendus par la planification (StepOut OPERA) afin de détecter les dérives temporelles, c'est-à-dire déterminer quand les dates des événements StepOut de l'historique ne correspondent plus aux dates des événements StepOut de la planification et d'en déduire la source de ces décalages (quelles étapes ? quelles opérations ?).

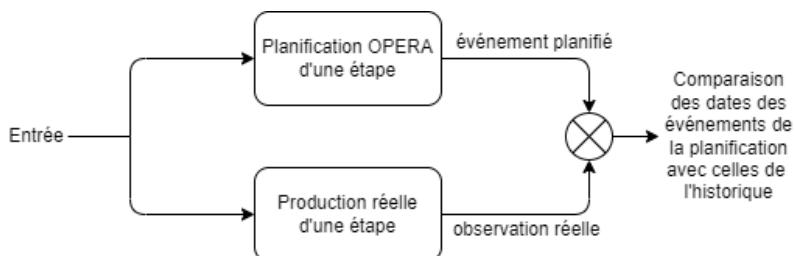


FIGURE 4.1 – Schéma général de notre problème de diagnostic de dérives temporelles

L'organisation du chapitre est la suivante. La section 4.1 définit le problème de détection de dérives temporelles dans les Graphes d'Événements Temporisés. La section 4.2 propose une modélisation formelle d'une étape de fabrication quelconque par des GET. Ces GET seront les modèles

à partir desquels nous présentons nos méthodes de diagnostic dans les chapitres 6 et 7. Enfin, le chapitre 8 exploitera ces méthodes pour le diagnostic de dérives temporelles dans une opération de fabrication de semiconducteurs.

4.1 Détection de dérives temporelles dans les Graphes d'Événements Temporisés

Pour mettre en place notre méthode de diagnostic à base de modèle, on réalise une modélisation par des Graphes d'Événements Temporisés (GET) définis dans la sous-section suivante. La sous-section 4.1.2 formalise la détection de dérives temporelles dans les GET.

4.1.1 Graphes d'Événements Temporisés (GET)

Les GET sont une sous-classe des réseaux de Petri (définition 3.2) où chaque place possède une transition en amont et une transition en aval ([Mur89], [Wan98] et [BCOQ92]). Des durées sont associées aux places et caractérisent le temps de séjour minimal des jetons.

Définition 4.1 (Graphe d'Événements Temporisés). Un Graphe d'Événements Temporisés (GET) \mathcal{R} est défini par le quintuplet :

$$\mathcal{R} = \langle P, R, F, M_0, H \rangle$$

avec

- $P = \{p_1, p_2, \dots, p_m\}$: l'ensemble des places,
- $R = \{r_1, r_2, \dots, r_n\}$: l'ensemble des transitions,
- $P \cap R = \emptyset$,
- $F \subseteq (P \times R) \cup (R \times P)$: l'ensemble des arcs entre les places et transitions,
- $M_0 : P \rightarrow \{0, 1, 2, 3, \dots\}$: le marquage initial des places,
- $H : P \rightarrow \{0, 1, 2, 3, \dots\}$: le temps de séjour des jetons par place.

L'ensemble des transitions est partitionné de la façon suivante : $R = E \cup I \cup S$, où :

- E est l'ensemble des transitions d'entrées notées u_i ,
- I est l'ensemble des transitions internes notées x_i ,
- S est l'ensemble des transitions de sorties notées y_i .

Un GET est soumis à une règle de franchissement au plus tôt des transitions ce qui signifie que toute transition franchissable est immédiatement franchie.

Par la suite, nous aurons besoin de manipuler des chemins dans un GET dont voici la définition.

Définition 4.2 (Chemin d'arcs). La séquence $(n_0, n_1), (n_1, n_2), \dots, (n_{m-1}, n_m)$ de longueur m est un chemin d'arcs dans un GET si $\forall i \in \{0, \dots, m-1\}, (n_i, n_{i+1}) \in F$. Le noeud n_i est en amont de n_j si $i < j$. Si n_i est en amont de n_j alors n_j est en aval de n_i . On note $n \rightsquigarrow n'$ le chemin entre le noeud n et le noeud n' (n est en amont de n').

Les GET sont utilisés pour faire notre modélisation car ils permettent de représenter tous les éléments qui forment une étape de fabrication d'un lot de plaques de semiconducteurs. L'élément le plus simple d'un GET, voir figure 4.2, est une place associée à une durée (temps de séjour des jetons) et entourée d'une transition aval et d'une transition amont. Il s'agit d'un délai h qui représente par exemple le temps de transport entre deux machines ou le temps de processus d'un équipement. Les autres structures élémentaires sont représentées sur la figure 4.3.

- La structure en tandem (4.3a) représente une simple succession de places et de transitions permettant de modéliser plusieurs tâches qui se suivent.

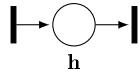


FIGURE 4.2 – Délai

- La structure parallèle (4.3b) représente l'initialisation simultanée de différentes tâches dans le système. Elles sont lancées au même instant lors du tir de la transition commune. D'un point de vue structurel, il s'agit de tandems partant d'une même transition.
- La structure de synchronisation (4.3c) permet à différentes tâches de se synchroniser, par exemple pour attendre la fin de plusieurs processus avant de démarrer l'exécution d'une ou plusieurs autres tâches. Une synchronisation est constituée de tandems se réunissant sur une transition commune.
- La structure boucle (4.3d) est par exemple utilisée pour modéliser une file d'attente d'une tâche sur un équipement. Le jeton représente le statut de l'équipement : libre ou en activité.

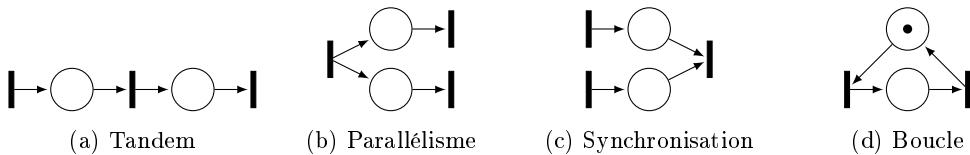


FIGURE 4.3 – Structures élémentaires d'un GET

Grâce aux structures élémentaires des GET, il est possible de modéliser différentes chaînes de production comme celle mise en exemple sur la figure 4.4.

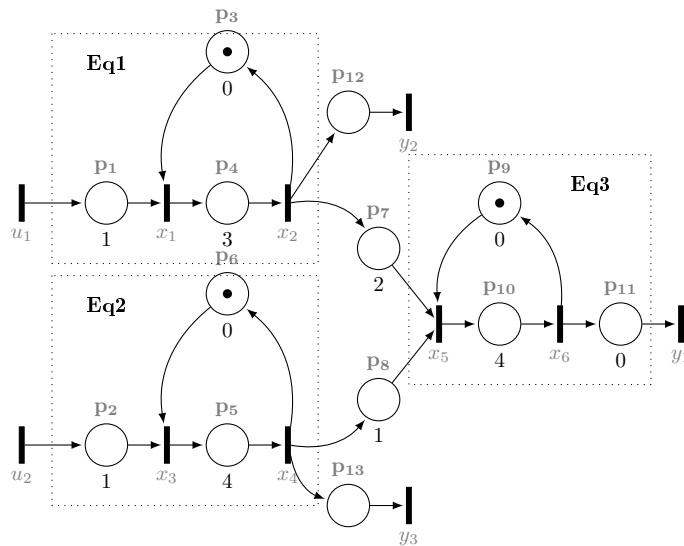


FIGURE 4.4 – Représentation d'une chaîne de production par un GET

Exemple 4.1. La figure 4.4 représente une chaîne de production modélisée par un GET et qui représente très globalement des opérations sur des plaques par différents équipements. Cette chaîne est constituée de deux équipements Eq1 et Eq2 qui traitent le même type d'opération sur les plaques mais avec des temps de processus différents, avant d'être traitées par un troisième équipement Eq3. Le processus de l'équipement 1 correspond à la place p_4 et est réalisé en 3 heures. Sa disponibilité correspond au jeton de la place p_3 . De même, les processus des équipements 2 et 3 sont respectivement modélisés par les places p_5 (durée : 4 heures) et p_{10} (durée : 4 heures) tandis que leurs disponibilités sont respectivement modélisées par les places p_6 et p_9 dans lesquelles un jeton est présent. La synchronisation entre les équipements 1 et 2 fait que l'équipement 3 ne peut commencer à travailler qu'avec des plaques disponibles venant de Eq1 et de Eq2. Les plaques prennent 2 heures

à l'équipement 1 pour arriver à l'équipement 3, 1 heure pour aller de l'équipement 2 à l'équipement 3. L'entrée de la ligne de production est un flux de plaques modélisé par les transitions u_1 , u_2 . Le déclenchement d'une transition d'entrée représente l'occurrence d'un événement provenant de capteurs sur la ligne de production qui indique l'arrivée de la plaque. L'entrée u_1 , respectivement u_2 , correspond à l'arrivée des plaques sur l'équipement 1, respectivement sur l'équipement 2. La sortie de la ligne de production est un flux de plaques modélisé par les tirs de la transition y_1 . Les sorties y_2 et y_3 fournissent via des capteurs des informations sur la fin du processus de fabrication des équipements 1 et 2.

Dans les GET qui modélisent une étape de fabrication de lots, GET qui servent de modèles au problème de diagnostic de dérives temporelles présenté en introduction de ce chapitre, les événements datés correspondent aux tirs des transitions. Les *événements observables* correspondent aux tirs de transitions dont on mesure les dates d'occurrence. Dans nos travaux, nous faisons l'hypothèse que les *transitions observables* sont les transitions d'entrée et les transitions de sortie, regroupés dans l'ensemble $R_o = E \cup S$. Autrement dit, le tir des transitions internes I est non-observable.

4.1.2 Détection de dérives temporelles dans un GET

Dans un GET, une dérive temporelle est une modification de la durée associée à une place comme indiqué dans la définition suivante.

Définition 4.3 (Dérive temporelle dans un GET). Soit h le temps de séjour associé à la place p d'un GET, une dérive temporelle sur la place p est une durée relative $\theta \in \mathbb{Z}$. Si elle se produit sur cette place, son temps de séjour devient $h + \theta$.

Si une telle dérive représente par exemple une augmentation de la durée d'opération ou du temps de transport, il s'agit alors d'un ralentissement¹. Si une dérive temporelle a lieu sur la place p , et que cette place n'est reliée en aval qu'à une seule transition, elle-même reliée en amont à cette seule place, les dates de tir de la transition seront modifiées par la valeur de θ .

Cependant, la structure du GET est souvent plus complexe et une dérive temporelle sur une place n'implique pas nécessairement que les dates de tirs des transitions situées en aval de cette place soient (toutes) affectées par la dérive. C'est notamment le cas des dérives qui se produisent sur des places situées avant une structure élémentaire de synchronisation dont une nouvelle figure plus complète est donnée en 4.5.

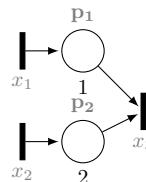


FIGURE 4.5 – Structure de synchronisation détaillée

Soient t_1 , respectivement t_2 , la date d'arrivée d'un jeton dans la place p_1 via x_1 , respectivement p_2 via x_2 . Ces temps dépendent des dates de tir des transitions d'entrée, des temps de séjour des jetons dans les places en amont de x_1 , respectivement x_2 et des dates de tir des transitions internes également en amont. La date de tir de x_3 est donc $t_3 = \max\{t_1 + 1, t_2 + 2\}$. Selon la place où la dérive temporelle se produit, selon sa valeur et selon les temps t_1 et t_2 , deux résultats sont possibles :

- Si la dérive temporelle a lieu sur la place p_1 et que $t_1 + 1 + \theta \leq t_2 + 2$, la dérive ne va pas modifier la date de tir de x_3 (en comparaison avec sa date de tir si la dérive n'avait pas lieu), car elle n'augmente pas suffisamment cette date par rapport à la date associée au chemin qui vient de x_2 ($x_2 \rightsquigarrow x_3$).
- Si la dérive temporelle a lieu sur la place p_2 et que $t_1 + 1 \geq t_2 + 2 + \theta$, la dérive ne modifie pas la date de tir de x_3 par rapport au chemin qui vient de x_1 ($x_1 \rightsquigarrow x_3$).

1. Si la dérive est une diminution du temps de séjour, on parle d'accélération.

Dans ces deux cas, l'effet de la dérive sur les dates des tirs de la transition x_3 est masqué par la synchronisation.

D'autre part, la détection d'une dérive temporelle dépend des informations observables à partir du système en fonctionnement. On parle alors de *détectabilité*. Comme précisé en fin de sous-section précédente, dans notre problème de diagnostic, les événements observables sont les tirs des transitions d'entrée et des transitions de sortie du GET regroupées dans l'ensemble R_o . Une définition de dérive temporelle détectable peut alors être établie pour caractériser les dérives qui ont un effet visible sur ces transitions observables.

Définition 4.4 (Dérive temporelle détectable dans un GET). Soit $R_o = E \cup S$ l'ensemble des transitions observables d'un GET (transitions d'entrée et transitions de sortie du GET), p une place d'un GET et $R_o^p \subseteq R_o$ l'ensemble des transitions observables situées en aval de cette place² ($\forall r_{oi}^p \in R_o^p, p \rightsquigarrow r_{oi}^p, R_o^p \subseteq S$). Une dérive temporelle $\theta \in \mathbb{Z}$ située sur la place p est *détectable* si elle modifie les occurrences datées des tirs d'au moins l'une des transitions observables de R_o^p par rapport aux occurrences datées des tirs des transitions observables lorsqu'aucune dérive n'est présente dans le système.

D'après cette définition, si les transitions de la figure 4.5 sont des transitions d'entrée et de sortie (x_1 devient u_1 , x_2 devient u_2 et x_3 devient y , donc $u_1, u_2, y \in R_o$ et $y \in R_o^p$), les deux dérives temporelles présentées ci-dessus ne sont pas détectables, la date d'occurrence de x_3 n'étant pas affectée par ces dérives.

4.2 Modélisation d'une étape de fabrication par des GET

Cette section présente la modélisation d'une étape de fabrication de lots de semiconducteurs par des GET. Le procédé de fabrication général est tout d'abord formalisé avant de se concentrer sur la formalisation d'une étape.

4.2.1 Formalisation du procédé de fabrication

Le fonctionnement de fabrication des semiconducteurs est présenté dans le chapitre 1. Cette sous-section formalise les relations entre les étapes, les opérations, les briques et les routes introduites dans la section 1.2 qui donne également les définitions des lots, produits et technologies. Pour décrire le procédé de fabrication général, nous nous plaçons dans un premier temps au niveau d'une technologie.

Une technologie représente le procédé de fabrication (processus), au sens large, liée aux grandes caractéristiques de la production de semiconducteurs, telles que la finesse de gravure, ou encore le type de matériaux. Un produit spécifie le circuit intégré fabriqué. Il requiert un enchaînement spécifique de motifs à réaliser pour chaque couche, et donc un enchaînement spécifique d'étapes. Un lot est un ensemble de plaques de semiconducteurs. Il contient généralement 25 plaques au démarrage de sa fabrication, mais peut en avoir moins, notamment si certaines plaques doivent être mises au rebut. Les plaques constituant un lot sont physiquement disposées dans un FOUP³. Le lot est l'objet de base de la logistique du système de production, c'est lui qui transite d'équipements en équipements afin que les plaques qu'il contient soient "processées". Les notations suivantes sont adoptées pour ces trois éléments.

- La notation \mathcal{T} représente l'ensemble des technologies, $te \in \mathcal{T}$ correspond à une technologie particulière de l'ensemble des technologies.
- L'ensemble \mathcal{P} représente les produits où $p \in \mathcal{P}$ correspond à un produit de l'ensemble des produits.
- \mathcal{L} est l'ensemble des lots, où $l \in \mathcal{L}$ est un lot de l'ensemble des lots.

Une route désigne la gamme de fabrication d'un lot, c'est-à-dire l'ensemble des étapes de fabrication par lesquelles le lot doit passer avant de pouvoir sortir de l'usine. Il existe une multitude de routes différentes selon le type de produit considéré. Une route se compose généralement de plusieurs centaines d'étapes et il n'est pas rare aujourd'hui de trouver des routes contenant plus

2. Ces transitions sont obligatoirement des transitions de sortie.

3. Boîte qui permet le transport des plaques entre les différents équipements, voir figure 1.4, page 10.

d'un millier d'étapes de fabrication. \mathcal{R} représente l'ensemble des routes dont chaque élément est noté $r \in \mathcal{R}$. Une route est composée d'un ensemble ordonné de briques de processus à réaliser pour une route r donnée. Cet ensemble est noté $\mathcal{B}(r)$ et $b_{1 \dots n_1} \in \mathcal{B}(r)$ représente les briques de 1 à n_1 de la route r . Chaque brique est composée d'un ensemble d'opérations noté $\mathcal{O}(r)$. $o_{1 \dots n_2} \in \mathcal{O}(r)$ représente les opérations de 1 à n_2 de la route r . Chaque opération est composée d'un ensemble d'étapes de processus noté $\mathcal{S}(r)$. $s_{1 \dots n_3} \in \mathcal{S}(r)$ représente les étapes de 1 à n_3 de la route r .

Ensemble	Indice	Description
\mathcal{T}	$te \in \mathcal{T}$	Technologies
\mathcal{P}	$p \in \mathcal{P}$	Produits
\mathcal{L}	$l \in \mathcal{L}$	Lots
\mathcal{R}	$r \in \mathcal{R}$	Routes
$\mathcal{B}(r)$	$b_{1 \dots n_1} \in \mathcal{B}(r)$	Briques d'une route r
$\mathcal{O}(r)$	$o_{1 \dots n_2} \in \mathcal{O}(r)$	Opérations d'une route r
$\mathcal{S}(r)$	$s_{1 \dots n_3} \in \mathcal{S}(r)$	Etapes d'une route r

TABLE 4.1 – Notations du procédé de fabrication général

Ces notations résumées dans la table 4.1 permettent de définir des fonctions sur les différentes technologies, produits et lots.

- La fonction $route : \mathcal{T} \rightarrow \mathcal{R}$ associe une technologie à une route. Chaque technologie $te \in \mathcal{T}$ a plusieurs routes $r \in route(te)$ possibles.
- La fonction $prod : \mathcal{R} \rightarrow \mathcal{P}$ associe une route à un produit. Chaque route a plusieurs produits $p \in prod(r)$ possibles.
- La fonction $lot : \mathcal{P} \rightarrow \mathcal{L}$ associe un produit à un lot. Chaque produit $p \in \mathcal{P}$ a plusieurs lots $l \in lot(p)$ possibles.

Enfin, les liens entre les briques, opérations et étapes sont établis.

- La fonction $bri : \mathcal{R} \rightarrow 2^{\mathcal{B}(r)}$ définit une route composée de plusieurs briques. Ces briques $bri(r) = \{b_{i_1}, \dots, b_{i_m}\}, \forall b \in \mathcal{B}(r)$ appartiennent à la route r .
- La fonction $op : \mathcal{B}(r) \rightarrow 2^{\mathcal{O}(r)}$ définit une brique composée de plusieurs opérations. Ces opérations sont notées $op(r, b) = \{o_{i_1}, \dots, o_{i_m}\}, \forall o \in \mathcal{O}(r), \exists! b \in \mathcal{B}(r), o \in op(r, b)$. Chaque opération d'une brique est unique.
- La fonction $st(r) : \mathcal{O}(r) \rightarrow 2^{\mathcal{S}(r)}$ définit une opération composée de plusieurs étapes. Ces étapes sont $st(r, o) = \{s_{i_1}, \dots, s_{i_m}\}$.

4.2.2 Focus sur une étape d'une opération

La section précédente définit formellement les liens entre les routes, briques, opérations et étapes, et les liens entre les technologies, produits et lots. Maintenant nous allons décrire le fonctionnement d'une étape de fabrication dont les différentes actions et durées sont illustrées sur la figure 4.6. Les différentes notations sont répertoriées dans la table 4.2.

Une étape fait partie d'une succession d'étapes d'une opération. Elle démarre et se termine par les actions StepIn et StepOut. De plus, une étape démarre pour un lot lorsque ce lot sort de l'équipement de l'étape précédente. Donc le StepOut de l'étape précédente est égal au StepIn de l'étape de la figure (et le StepOut de la figure correspond au StepIn de l'étape suivante). Le choix de l'équipement est acté par l'action JobPrep qui indique que le lot est assigné à un équipement. L'arrivée du lot sur l'équipement est indiqué par l'action⁴ JobIn et TrackIn. Le début du processus est représenté par l'événement StartRun, la fin par l'événement EndRun. Quand le lot sort de l'équipement, l'action effectuée est appelée JobOut, StepOut et TrackOut.

La durée entre le StepIn et le JobIn correspond à un temps d'attente noté CTWait. La durée entre JobIn et StepOut correspond au temps de processus qui est noté CTTH. Le CTTH des étapes est calculé par le service de *planification opérationnelle de la production* de la figure 1.8. La somme

4. L'arrivée du lot est unique mais deux informations sont générées au moment où l'action se produit. Ces informations ont des noms différents car sont dévolues à plusieurs destinataires (production, opérateur).

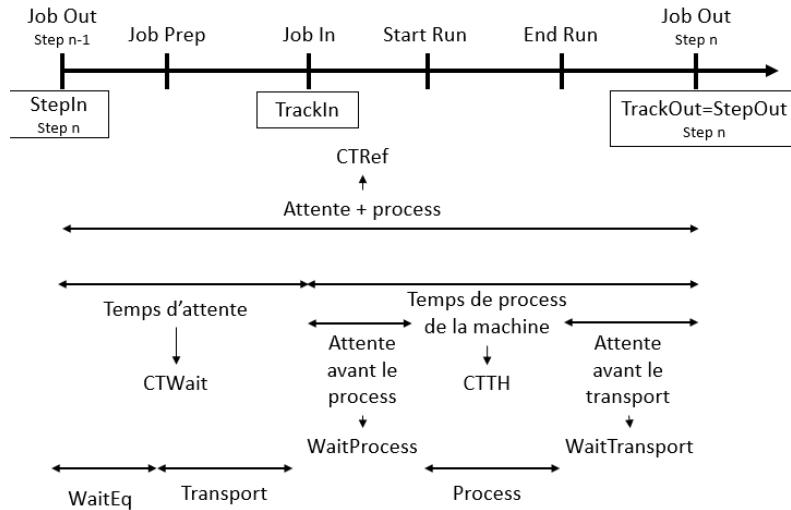


FIGURE 4.6 – Description des actions et durées d'une étape

des temps d'attente et de processus est notée **CTRef** qui est la référence sur le temps d'une étape. Ce calcul vient de la thèse [Deq17]. La durée minimale pour effectuer l'étape est notée **CTMin**. Cette durée n'apparaît pas dans la figure car elle est calculée par rapport au temps de processus et à un temps arbitraire pour chaque étape dans l'usine :

$$CTMin = \max(2.5 \times CTTH, 15min).$$

Notation	Description
StepIn	Début de l'étape
JobPrep	Assignation du lot à un équipement
JobIn	Arrivée du lot sur l'équipement
TrackIn	idem
StartRun	Début du processus
EndRun	Fin du processus
JobOut	Sortie du lot de l'équipement et fin de l'étape
StepOut	idem
TrackOut	idem
CTWait	Temps d'attente
CTTH	Temps de processus
CTMin	Temps minimal d'exécution de l'étape
CTRef	Temps de référence (temps d'attente plus le temps de processus)

TABLE 4.2 – Notations des différentes actions d'une étape

Certaines actions au cours de la vie de l'étape sont observables (StepOut, JobIn, JobPrep). Cependant, nous choisissons pour notre modélisation de ne conserver que les observations des StepOut, informations les plus sûres par rapport aux exigences de l'usine. Les occurrences datées de ces événements sont connues de l'historique et de la planification. Comme indiqué au début de cette sous-section, le StepOut d'une étape est égal au StepIn de l'étape suivante. Ces informations vont donc servir comme entrées et sorties observables des GET dans la prochaine sous-section.

4.2.3 Modélisation d'une étape par des GET

Avant de modéliser une étape par des GET, un réseau de Petri est utilisé pour représenter tous ses comportements possibles (le lot est stocké, la préparation est annulée...). On ne représente alors que les relations structurelles des différents fonctionnements de l'étape. Une fois le réseau de Petri mis en place, il est décomposé en plusieurs GET dans lesquels les informations temporelles (temps d'attente, temps de process...) de l'étape sont modélisées. Ces GET serviront de modèles lors du diagnostic de dérives temporelles pour une étape de fabrication.

La modélisation se fait pour une technologie te d'un produit p et pour une étape s d'une opération o et d'une brique b . La figure 4.7 représente la modélisation d'une étape en réseau de Petri. Elle reprend les informations des sous-sections précédentes, notamment la figure 4.6. La transition d'entrée est le StepIn de l'étape (StepOut de l'étape précédente) et sa transition de sortie est son StepOut. La place notée Transport WOJobPrep correspond au transport des lots entre l'équipement de l'étape précédente et le prochain équipement. Si les lots ne sont pas attribués à un équipement, ils partent dans une autre place notée Stocker qui correspond au stockage des lots avant leur attribution sur les équipements, sinon les lots vont dans une place notée TransportWithJobPrep qui correspond au transport des lots vers l'équipement attribué. Puis les lots vont soit sur la place notée Process pour être "processés", la date d'entrée dans la place correspond à la date du début du processus, soit sont renvoyés dans le transport pour une nouvelle attribution d'équipements via la transition JobPrepCancel. Si un problème survient dans le processus de fabrication, les lots vont dans une place notée RouteRework qui correspond à un problème et à un changement de route de fabrication.

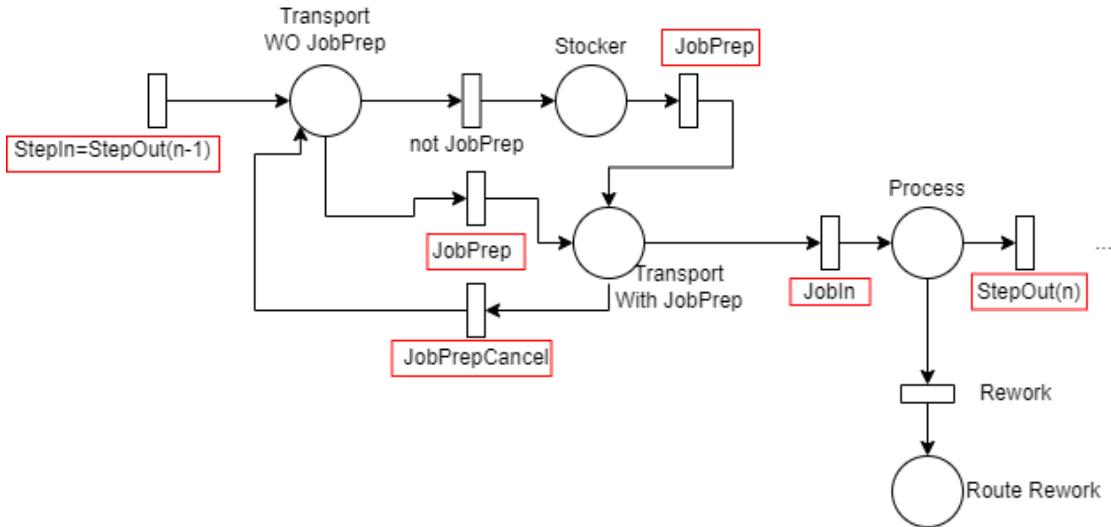


FIGURE 4.7 – Réseau de Petri d'une étape de processus de fabrication

Ce réseau de Petri modélisant toute la complexité des comportements d'une étape doit maintenant être découpé en plusieurs GET afin d'identifier plus simplement chaque stade dans lequel est le lot (en stockage, dans l'équipement). Chaque GET représente alors un fonctionnement spécifique de l'étape. Nous présentons dans un premier temps la structure des GET sans les temporisations qui seront décrites dans un second temps.

- Le premier GET modélisé est donné figure 4.8 et représente le fonctionnement de l'étape sans passer par le Stocker, donc dès que le lot sort de l'équipement précédent, il va directement sur l'équipement disponible pour exécuter l'étape.
- La deuxième modélisation est donnée figure 4.9 et représente le fonctionnement de l'étape lorsque le lot passe par le Stocker. Dès que le lot sort de l'équipement précédent, il va directement sur le Stocker car l'assignation de l'équipement n'est pas encore attribuée pour exécuter l'étape.

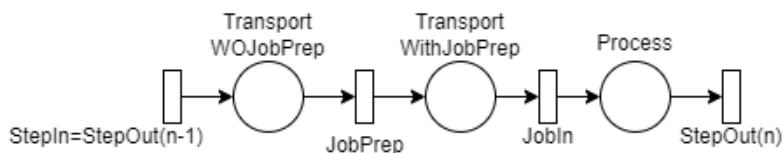


FIGURE 4.8 – GET du fonctionnement d'une étape qui assigne directement un équipement à un lot

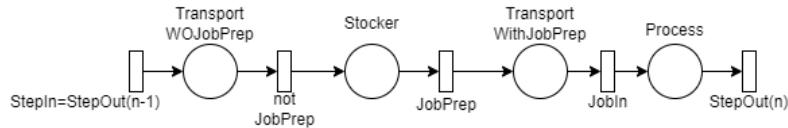


FIGURE 4.9 – GET du fonctionnement d'une étape qui utilise le Stocker pour le lot

- Le troisième GET donné figure 4.10 représente le fonctionnement de l'étape sans passer par le Stocker mais l'exécution du processus s'effectue avec un problème, le lot va donc sur une autre route représentée par la place RouteRework.
- Enfin, le fonctionnement de l'étape avec passage sur le Stocker et problème lors de l'exécution du processus est représenté par le GET de la figure 4.11.

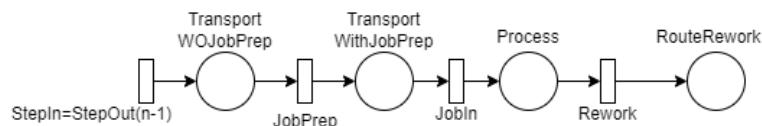


FIGURE 4.10 – GET du fonctionnement d'une étape pour laquelle le lot sort de la route

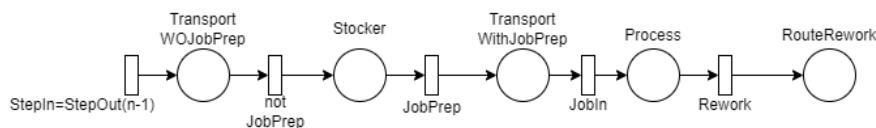


FIGURE 4.11 – GET du fonctionnement d'une étape pour laquelle le lot passe par le Stocker puis sort de la route

Les GET des figures 4.10 et 4.11 représentent des fonctionnements pour lesquels les lots sortent de la route qu'ils doivent suivre pour être produits. Ces changements de route (RouteRework) ne sont pas pris en compte dans la planification d'OPERA (et impliquent de nombreuses mesures et analyses que les opérateurs de l'usine repèrent très vite). Nous ne pouvons donc pas utiliser ces modèles dans notre problème de diagnostic de dérives temporelles. Nous nous concentrerons alors sur les fonctionnements des GET figures 4.8 et 4.9.

Pour ces modèles, les informations temporelles du tableau 4.2 (CTWait, CTTH, CTMin) peuvent être prises en compte. Le problème est qu'à partir des données détaillées dans l'outil OPERA, source d'information principale pour établir tous nos modèles, il n'est pas possible de connaître les durées associées à toutes les places des GET. Par exemple, si on connaît le temps moyen de processus (place Process), on ne connaît pas le temps de transport dans le cas où l'assignation du lot à un équipement est immédiat (place TransportWithJobPrep). Il faut donc créer de nouveaux GET à partir des modélisations précédentes et des données planifiées d'OPERA comparables aux observations réelles issues de l'historique de la production.

Commençons par le fonctionnement sans le passage par le Stocker (GET de la figure 4.8). D'après le tableau 4.2, deux configurations temporelles peuvent être modélisées par deux nouveaux GET.

- La durée moyenne minimale de l'étape CTMin est connue. Cette durée est le temps total entre le début de l'étape StepIn et la fin de l'étape StepOut. Le GET obtenu, voir figure 4.12, est le même que celui de la figure 4.8 auquel on ajoute une place entre la transition StepIn et la transition StepOut. Cette place a pour temps de séjour la durée CTMin.
- Le temps moyen d'attente CTWait est connu. Ce temps englobe l'attente et le transport (places TransportWOJobPrep et TransportWithJobPrep), et la durée moyenne de processus CTTH (place Process). Le GET obtenu pour cette configuration, voir figure 4.13, reprend le GET de la figure 4.8 auquel on ajoute une place entre la transition StepIn et la transition JobIn. Cette place a pour temps de séjour la durée CTWait.

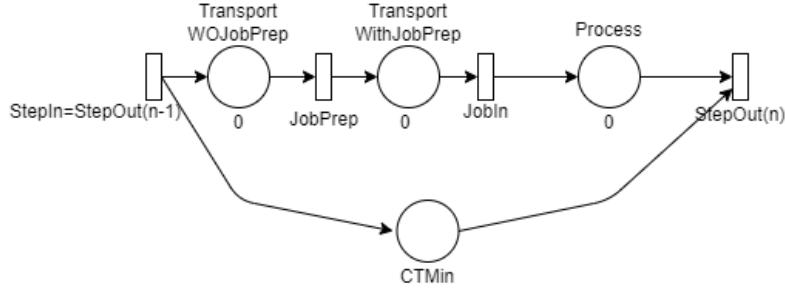


FIGURE 4.12 – GET du fonctionnement d’une étape sans Stocker dont on connaît la durée moyenne minimale

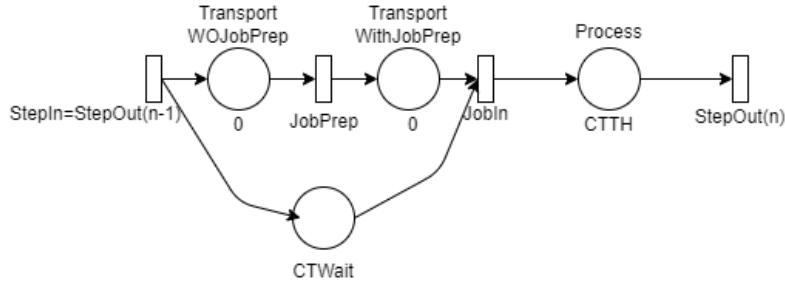


FIGURE 4.13 – GET du fonctionnement d’une étape sans Stocker dont on connaît le temps moyen d’attente et celui du processus

La même méthode de modélisation est appliquée sur le fonctionnement d’une étape pour laquelle le lot passe par le Stocker (GET de la figure 4.9). Les deux GET obtenus sont alors celui de la figure 4.14 avec le temps CTMin, et celui de la figure 4.15 avec les temps CTWait et CTTH.

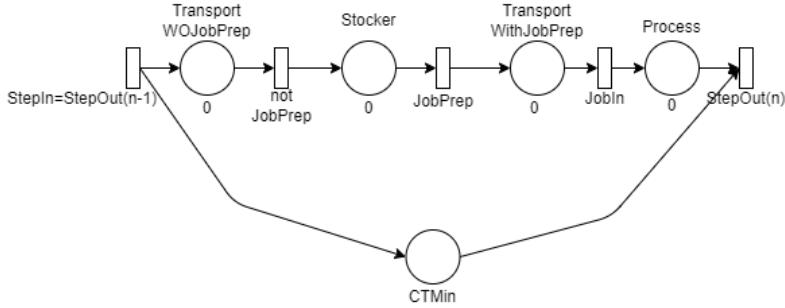


FIGURE 4.14 – GET du fonctionnement d’une étape avec Stocker dont on connaît le temps moyen minimal

Dans ces modèles, les transitions d’entrée et de sortie, StepIn et StepOut, sont observables (elles appartiennent à R_o) car il s’agit de l’information StepOut enregistrée dans l’historique et connue de la planification d’OPERA.

4.3 Synthèse du chapitre

Le chapitre présente la méthode que l’on met en place pour le diagnostic dans ces travaux. Notre objectif est d’établir une méthode à base de modèle utilisant des GET pour le diagnostic de dérives temporelles. Nous proposons une définition formelle du problème (modèles, observations, dérives temporelles). La deuxième partie de ce chapitre est consacrée à la description de notre cas d’étude à savoir l’analyse de dérives au niveau d’une étape de fabrication. Dans un premier temps, ce cas d’étude est décrit formellement par l’introduction de notations mathématiques sur sa structure et ses propriétés temporelles. Dans un second temps, nous proposons la modélisation

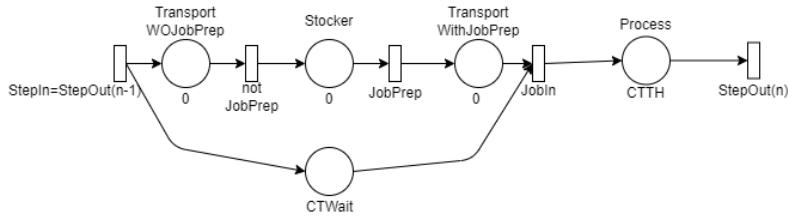


FIGURE 4.15 – GET du fonctionnement d'une étape avec *Stocker* dont on connaît le temps moyen d'attente et celui du processus

d'une étape de fabrication d'un produit dans la planification de OPERA en réseau de Petri puis de traduire cette modélisation en GET pour pouvoir mettre en place notre diagnostic. Grâce à la modélisation d'une étape de production on peut mettre en place notre méthode de diagnostic de façon algébrique en utilisant l'algèbre $(\max, +)$.

Comme présenté dans l'introduction de cette thèse, les exemples donnés dans la suite de nos travaux utilisent la bibliothèque MAXPLUSDIAG implémentée en C++, qui repose sur la bibliothèque MINMAXGD [CLHB00] qui permet de réaliser toutes les opérations formelles du dioïde $\mathcal{M}_{in}^{ax}[\gamma, \delta]$.

Chapitre 5

Outils algébriques

Ce chapitre présente les outils algébriques utilisés pour définir les dioïdes. Les dioïdes permettent de faire une définition de l'algèbre $(\max, +)$ utilisée pour faire une modélisation des systèmes $(\max, +)$ -linéaires. De nombreux documents présentent les éléments fondamentaux de la théorie des dioïdes notamment [BCOQ92], [Max91], [Coh94]. On donnera dans un premier temps, un rappel des mathématiques fondamentales définies dans le livre [ALF78] et utilisées pour la définition des dioïdes. Dans un second temps, la théorie des dioïdes et la théorie de la résiduation sont présentées. Enfin, les dioïdes, et plus particulièrement le dioïde $\mathcal{M}_{in}^{ax}[\gamma, \delta]$, sont utilisés pour modéliser mathématiquement les systèmes $(\max, +)$ -linéaires (dont la représentation graphique a été présentée dans le chapitre 4 par les GET, Graphes d'Événements Temporisés) et une comparaison temporelle entre deux séries formelles de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ est proposée.

5.1 Rappels mathématiques

Cette section fait un rappel mathématique des définitions tirées du livre [ALF78]. Il définit les notions mathématiques que l'on utilise pour présenter les dioïdes qui permettent de définir l'algèbre $(\max, +)$.

Définition 5.1 (Idempotence). Soit \mathcal{E} un ensemble muni d'une loi interne notée \diamond . Cette loi interne est idempotente si $\forall a \in \mathcal{E}$, $a \diamond a = a$.

Définition 5.2 (Injection canonique). Soit \mathcal{F} un sous-ensemble de l'ensemble \mathcal{E} . L'injection canonique de \mathcal{F} dans \mathcal{E} est l'application notée $I_{\mathcal{F}}$ définie pour tout $x \in \mathcal{F}$ telle que :

$$I_{\mathcal{F}}(x) = x.$$

Définition 5.3 (Relation d'ordre). Une relation \mathcal{R} sur un ensemble \mathcal{E} est une relation d'ordre si elle respecte les trois règles suivantes :

- la relation \mathcal{R} doit être réflexive : $x \mathcal{R} x$ avec $x \in \mathcal{E}$,
- la relation \mathcal{R} doit être transitive : si $x \mathcal{R} y$ et $y \mathcal{R} z$ alors $x \mathcal{R} z$ avec $x, y, z \in \mathcal{E}$,
- la relation \mathcal{R} doit être antisymétrique : si $x \mathcal{R} y$ et $y \mathcal{R} x$ alors $y = x$ avec $x, y \in \mathcal{E}$.

Définition 5.4 (Ensemble ordonné). Un ensemble ordonné est un ensemble muni d'une relation d'ordre \mathcal{R} . Cette relation est dite totale si deux éléments de l'ensemble sont comparables sinon la relation est partielle. Soit \mathcal{E} un ensemble ordonné par une relation d'ordre notée \preceq , on obtient les notions suivantes définies pour \mathcal{E} et $\mathcal{F} \subset \mathcal{E}$:

- majorant : $M_a \in \mathcal{E}$ est un majorant de \mathcal{F} si $\forall f \in \mathcal{F}, f \preceq M_a$,
- minorant : $M_i \in \mathcal{E}$ est un minorant de \mathcal{F} si $\forall f \in \mathcal{F}, M_i \preceq f$,
- borne supérieure : le plus petit majorant de \mathcal{F} est noté $\bigvee \mathcal{F}$ ($= \sup \mathcal{F}$),
- borne inférieure : le plus grand minorant de \mathcal{F} est noté $\bigwedge \mathcal{F}$ ($= \inf \mathcal{F}$),

- plus grand élément : $\top \in \mathcal{E}$ est l'élément maximal de \mathcal{E} si $\forall x \in \mathcal{E}, x \preceq \top$.
- plus petit élément : $\perp \in \mathcal{E}$ est l'élément minimal de \mathcal{E} si $\forall x \in \mathcal{E}, \perp \preceq x$.

Définition 5.5 (Isotonie, antitonie, monotonie). Soit $\Pi : \mathcal{E} \rightarrow \mathcal{F}$ une application définie sur des ensembles ordonnés. On dit :

- Π isotone $\equiv \forall s, s' \in \mathcal{E} s \preceq s' \Rightarrow \Pi(s) \preceq \Pi(s')$,
- Π antitone $\equiv \forall s, s' \in \mathcal{E} s \preceq s' \Rightarrow \Pi(s) \succeq \Pi(s')$,
- Π monotone $\equiv \Pi$ isotone ou Π antitone.

Définition 5.6 (Application linéaire). Une application $f : \mathcal{E} \rightarrow \mathcal{F}$ est une application linéaire si elle vérifie l'additivité :

$$\forall (x, y) \in \mathcal{E}^2, f(x + y) = f(x) + f(y),$$

l'homogénéité :

$$\forall \lambda \in \mathbb{R} \quad \forall x \in \mathcal{E}, f(\lambda \times x) = \lambda \times f(x).$$

Définition 5.7 (Application surjective). Une application $f : \mathcal{E} \rightarrow \mathcal{F}$ est dite surjective, si :

$$\forall y \in \mathcal{F} \quad \exists x \in \mathcal{E}, f(x) = y.$$

Une telle application est une application pour laquelle tout élément de l'ensemble d'arrivée a au moins un antécédent, c'est-à-dire que l'application est image d'au moins un élément de l'ensemble de départ comme le montre la figure 5.1.

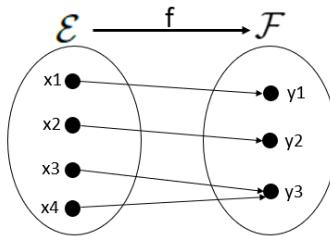


FIGURE 5.1 – Schéma d'une application surjective

Définition 5.8 (Application injective). Une application $f : \mathcal{E} \rightarrow \mathcal{F}$ est dite injective, si :

$$\forall x, x' \in \mathcal{E}, f(x) = f(x') \Rightarrow x = x'.$$

Une application est injective si tout élément de son ensemble d'arrivée a au plus un antécédent, c'est-à-dire que deux éléments distincts de son ensemble de départ ont deux images distinctes comme le montre la figure 5.2.

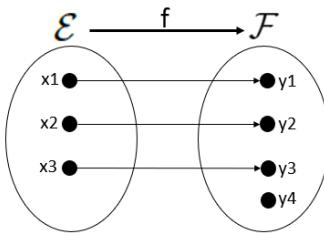


FIGURE 5.2 – Schéma d'une application injective

5.2 Théorie des dioïdes

Avec les définitions données dans la section 5.1, cette section présente la théorie des dioïdes qui permet définir le dioïde $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ qui est utilisé pour décrire les systèmes $(\max, +)$ -linéaires. Les définitions sont tirées des documents [BCOQ92], [Coh94], [Max91], [Cot99] et [Lho03].

5.2.1 Dioïdes

Définition 5.9 (Monoïde). Un ensemble \mathcal{D} est appelé monoïde s'il est muni d'une loi de composition interne notée \oplus qui est associative ($\forall a, b, c \in \mathcal{D}$, $(a \oplus b) \oplus c = a \oplus (b \oplus c) = a \oplus b \oplus c$) avec un élément neutre noté ε ($\forall a \in \mathcal{D}$, $a \oplus \varepsilon = a$). Si la loi est commutative ($\forall a, b \in \mathcal{D}$, $a \oplus b = b \oplus a$) alors le monoïde est commutatif. Si la loi est idempotente ($\forall a \in \mathcal{D}$, $a \oplus a = a$) alors le monoïde est idempotent.

Définition 5.10 (Demi-anneau). On appelle demi-anneau un ensemble \mathcal{D} muni de deux lois internes telles que :

- (\mathcal{D}, \oplus) est un monoïde commutatif dont l'élément neutre est ε .
- (\mathcal{D}, \otimes) est un monoïde, son élément neutre est noté e .
- La loi multiplicative \otimes est distributive à droite et à gauche par rapport à la loi additive \oplus ($\forall a, b, c \in \mathcal{D}$, $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$ et $(b \oplus c) \otimes a = (b \otimes a) \oplus (c \otimes a)$).
- L'élément neutre ε de la loi \oplus est absorbant pour la loi \otimes ($\forall a \in \mathcal{D}$, $a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$).

Définition 5.11 (Dioïde). Un demi-anneau \mathcal{D} est un dioïde lorsque la loi \oplus est idempotente.

Remarque. Quand il n'y a pas d'ambiguïté dans les opérations, le signe \otimes peut être omis. Ainsi, $\forall a, b \in \mathcal{D}$, $a \otimes b = ab$.

Exemple 5.1. L'ensemble $(\mathbb{R} \cup (-\infty))$ muni de l'opération max pour la loi \oplus et de l'addition classique pour la loi \otimes , correspond au dioïde \mathbb{R}_{max} pour lequel $\varepsilon = -\infty$ et $e = 0$.

Définition 5.12 (Sous-dioïde). Soit \mathcal{D} un dioïde, le sous-ensemble $\mathcal{C} \subset \mathcal{D}$ est un sous-dioïde si :

- les éléments neutres de \mathcal{D} appartiennent à \mathcal{C} : $\varepsilon \in \mathcal{C}$ et $e \in \mathcal{C}$,
- \mathcal{C} est fermé pour les lois \oplus et \otimes : $\forall a, b \in \mathcal{C}$, $a \oplus b \in \mathcal{C}$ et $a \otimes b \in \mathcal{C}$.

Exemple 5.2. L'ensemble $\mathbb{Z}_{max} = (\mathbb{Z} \cup (-\infty))$ avec l'opération max pour \oplus , et l'addition pour \otimes est un sous-dioïde de \mathbb{R}_{max} car les éléments neutres de \mathbb{R}_{max} sont les mêmes que ceux de \mathbb{Z}_{max} . De plus, pour $a, b \in \mathbb{Z}_{max}$, $a \oplus b \in \mathbb{Z}_{max}$ et $a \otimes b \in \mathbb{Z}_{max}$ donc \mathbb{Z}_{max} est fermé pour \oplus et \otimes .

Définition 5.13 (Relation d'ordre dans un dioïde). Dans un dioïde \mathcal{D} , la loi \oplus introduit naturellement une relation d'ordre notée \preceq et telle que :

$$\forall a, b \in \mathcal{D}, a \preceq b \Leftrightarrow a \oplus b = b.$$

Démonstration.

- L'opération \oplus est idempotente : $a \oplus a = a$ donc $a \preceq a$, la relation est réflexive.
- $a \preceq b$ et $b \preceq c$ si et seulement si $a \oplus b = b$ et $b \oplus c = c$. De plus, comme \oplus est une opération associative $c = (a \oplus b) \oplus c = a \oplus (b \oplus c) = a \oplus c$ d'où $a \preceq c$ donc la relation est transitive.
- L'opération \oplus est commutative donc $a \oplus b = b \oplus a = a = b$. Donc $a \preceq b$ et $b \preceq a$ impliquent $a = b$, la relation est antisymétrique.

La relation \preceq est donc une relation d'ordre. □

Cette relation d'ordre est associée aux opérateurs \oplus et \otimes comme suit :

$$\forall a, b, c \in \mathcal{D}, a \preceq b \Rightarrow \begin{cases} a \oplus c \preceq b \oplus c, \\ a \otimes c \preceq b \otimes c, \\ c \otimes a \preceq c \otimes b. \end{cases}$$

Définition 5.14 (Dioïde complet). Un dioïde est dit complet s'il est fermé pour les sommes infinies et si la loi \otimes est distributive sur les sommes infinies, c'est-à-dire si pour tout $a \in \mathcal{D}$ et pour tout sous-ensemble $\mathcal{C} \subset \mathcal{D}$,

$$\begin{aligned} a \otimes \bigoplus_{x \in \mathcal{C}} x &= \bigoplus_{x \in \mathcal{C}} a \otimes x, \\ \bigoplus_{x \in \mathcal{C}} x \otimes a &= \bigoplus_{x \in \mathcal{C}} x \otimes a. \end{aligned}$$

Un dioïde \mathcal{D} complet, admet un plus grand élément noté \top et vérifiant $\forall a \in \mathcal{D}$ et $a \neq \varepsilon$, $\top \oplus a = \top$ donc par extension :

$$\top = \bigvee_{x \in \mathcal{D}} x = \bigoplus_{x \in \mathcal{D}} x.$$

Un dioïde complet admet également une borne inférieure représentée par la loi \wedge associative, commutative, idempotente et isotone qui vérifie :

$$\forall a, b \in \mathcal{D}, a \wedge b = a \Leftrightarrow a \preceq b \Leftrightarrow a \oplus b = b.$$

Exemple 5.3. En reprenant l'exemple 5.1 du dioïde \mathbb{R}_{max} , il n'est pas complet car $+\infty$ n'appartient pas à l'ensemble \mathbb{R}_{max} . Dans l'ensemble \mathbb{R}_{max} la somme infinie n'est pas définie sur $+\infty$. En ajoutant $\top = +\infty$ au dioïde \mathbb{R}_{max} , on obtient le dioïde complet $\overline{\mathbb{R}}_{max}$. La borne inférieure \wedge de $\overline{\mathbb{R}}_{max}$ correspond alors à l'opérateur $\min (a \wedge b = \min(a, b))$.

Définition 5.15 (Dioïde distributif). Soit \mathcal{D} un dioïde complet. Il est distributif si $\forall b \in \mathcal{D}$ et pour tout sous-ensemble $\mathcal{C} \subset \mathcal{D}$:

$$\begin{aligned} \left(\bigoplus_{x \in \mathcal{C}} x \right) \wedge b &= \bigoplus_{x \in \mathcal{C}} (x \wedge b), \\ \left(\bigwedge_{x \in \mathcal{C}} x \right) \oplus b &= \bigwedge_{x \in \mathcal{C}} (x \oplus b). \end{aligned}$$

Définition 5.16 (Dioïde matriciel). Soit \mathcal{D} un dioïde complet, l'ensemble des matrices carrées de dimension n à coefficients dans \mathcal{D} est notée $\mathcal{D}^{n \times n}$. La somme et le produit de matrices sont définis pour $A, B \in \mathcal{D}^{n \times n}$ et $\forall i, j \in \{1 \dots n\}$:

$$\begin{aligned} A \oplus B : \quad (A \oplus B)_{ij} &= A_{ij} \oplus B_{ij}, \\ A \otimes B : \quad (A \otimes B)_{ij} &= \bigoplus_{k=1}^n (A_{ik} \otimes B_{kj}). \end{aligned}$$

L'élément neutre pour l'opération \oplus est noté ε et correspond à une matrice composée entièrement de ε . L'élément neutre pour l'opération \otimes est noté e et correspond à une matrice composée de e sur la diagonale et de ε sur le reste.

La somme et le produit de matrices compatibles mais pas nécessairement carrées sont possibles en définissant les opérations comme suit pour $A, B, C \in \mathcal{D}^{n \times p}$ et $D \in \mathcal{D}^{p \times q}$:

$$\begin{aligned} \forall i = [1, n], \forall j = [1, p], \quad A \oplus B : \quad (A \oplus B)_{ij} &= A_{ij} \oplus B_{ij}, \\ \forall i = [1, n], \forall j = [1, q], \quad C \otimes D : \quad (C \otimes D)_{ij} &= \bigoplus_{k=1}^n (C_{ik} \otimes D_{kj}). \end{aligned}$$

Résolution d'équation implicite et opérateur étoile de Kleene

Un résultat important de la théorie des dioïdes pour la résolution d'équation implicite du type $f(x) = x$ est le théorème de l'étoile de Kleene. Dans un premier temps, cet opérateur et ses propriétés sont définis.

Définition 5.17 (L'opérateur étoile de Kleene). L'opérateur étoile de Kleene est noté $*$ et est défini comme suit sur un dioïde complet \mathcal{D} avec $a \in \mathcal{D}$:

$$a^* = \bigoplus_{i \geq 0} a^i \text{ avec } a^{i+1} = a \otimes a^i \text{ et } a^0 = e.$$

L'opérateur "plus" construit à partir de l'étoile de Kleene est noté $+$ et est défini comme suit sur un dioïde complet \mathcal{D} avec $a \in \mathcal{D}$:

$$a^+ = \bigoplus_{i \geq 1} a^i.$$

Ainsi, $a^+ = aa^* = a^*a$.

Remarque. Les opérateurs "plus" et étoile de Kleene sont des applications isotones (voir définition 5.5). C'est-à-dire $\forall a, b \in \mathcal{D}$:

$$a \preceq b \Rightarrow a^* \preceq b^*, \tag{5.1}$$

$$a \preceq b \Rightarrow a^+ \preceq b^+. \tag{5.2}$$

L'étoile de Kleene et l'opérateur "plus" ont de nombreuses propriétés prouvées dans les documents suivants [BCOQ92], [Lho03] $\forall a, b \in \mathcal{D}$:

$$a \preceq a^*, \quad (5.3)$$

$$(a^*)^* = a^*, \quad (5.4)$$

$$a(ba)^* = (ab)^*a, \quad (5.5)$$

$$a^*a^* = a^*, \quad (5.6)$$

$$(ab^*)^* = e \oplus a(a \oplus b)^*, \quad (5.7)$$

$$a^+ = (a^+)^+, \quad (5.8)$$

$$a^* = e \oplus a^+, \quad (5.9)$$

$$(a^*)^+ = (a^+)^* = a^*, \quad (5.10)$$

$$a^+ \preceq a^*, \quad (5.11)$$

$$(a^+)^+ = a^+, \quad (5.12)$$

Lorsque \mathcal{D} est commutatif :

$$(a \oplus b)^* = a^*b^*. \quad (5.13)$$

En outre, en lien avec la relation d'ordre du dioïde, il existe ces propriétés particulières $\forall a, x \in \mathcal{D}$:

$$x^* \preceq a^* \Leftrightarrow x \preceq a^*, \quad (5.14)$$

$$x^+ \preceq a^+ \Leftrightarrow x \preceq a^+. \quad (5.15)$$

Seul le détail de la démonstration des solutions de l'inéquation (5.14) est donné ci-dessous (la démonstration de (5.15) en découle directement).

Démonstration. (\Rightarrow)

$$\Rightarrow x \preceq x^* \preceq a^* \text{ car } x \preceq x^* \text{ (inéquation (5.3))}.$$

(\Leftarrow)

$$\begin{aligned} & x \preceq a^* \\ \Rightarrow & x^* \preceq (a^*)^* \text{ car l'étoile de Kleene est isotone (équation (5.1))}, \\ \Rightarrow & x^* \preceq a^* \text{ car } (a^*)^* = a^* \text{ (équation (5.4))}. \end{aligned}$$

□

Le calcul de l'étoile de Kleene a été étendu dans le cas matriciel par le théorème suivant.

Théorème 5.1 ([Cot99]). *Soit $A \in \mathcal{D}^{n \times n}$ une matrice partitionnée en quatre blocs :*

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}.$$

La matrice A^ s'écrit alors :*

$$A^* = \begin{pmatrix} a_{11}^* \oplus a_{11}^*a_{12}(a_{21}a_{11}^*a_{12} \oplus a_{22})^*a_{21}a_{11}^* & a_{11}^*a_{12}(a_{21}a_{11}^*a_{12} \oplus a_{22})^* \\ (a_{21}a_{11}^*a_{12} \oplus a_{22})^*a_{21}a_{11}^* & (a_{21}a_{11}^*a_{12} \oplus a_{22})^* \end{pmatrix}.$$

À présent, l'opérateur étoile de Kleene est utilisé pour résoudre l'équation implicite $ax \oplus b = x$ dans le théorème ci-dessous.

Théorème 5.2 (Théorème de l'étoile [BCOQ92]). *Soit un dioïde complet \mathcal{D} , l'équation implicite $ax \oplus b = x$ a comme plus petite solution $x = a^*b$.*

Démonstration. On prouve que a^*b est solution de $ax \oplus b = x$ en prenant $x = a^*b$:

$$\begin{aligned} a \otimes (a^*b) \oplus b &= aa^*b \oplus b \\ &= a^+b \oplus b \\ &= (a^+ \oplus e) \otimes b \\ &= a^*b \text{ car } a^* = a^+ \oplus e \text{ (équation (5.9))}. \end{aligned}$$

Donc a^*b est solution de $ax \oplus b = x$. Maintenant il faut prouver que a^*b est le minorant de l'équation $ax \oplus b = x$:

$$\begin{aligned} x &= ax \oplus b \\ &= a(ax \oplus b) \oplus b \\ &= a^2x \oplus ab \oplus b \\ &= a^2(ax \oplus b) \oplus ab \oplus b \\ &= a^3x \oplus a^2b \oplus ab \oplus b \\ &= a^n x \oplus (a^{n-1}b \oplus \dots \oplus ab \oplus b) \quad \succeq \quad a^{n-1}b \oplus \dots \oplus ab \oplus b \quad \text{avec } n \geq 1 \end{aligned}$$

Toute autre solution que a^*b est donc nécessairement supérieure ou égale à a^*b . \square

5.2.2 Dioïde de séries formelles et dioïde quotient

Dioïde de séries formelles

Définition 5.18 (Série formelle). Une série formelle en l'indéterminée notée z et à coefficients dans un dioïde \mathcal{D} est l'application s définie de \mathbb{Z} dans \mathcal{D} pour laquelle $\forall k \in \mathbb{Z}$, $s(k)$ est le coefficient de z^k . La série s peut également être représentée par l'expression suivante :

$$s = \bigoplus_{k \in \mathbb{Z}} s(k)z^k = s(0) \oplus s(1)z \oplus \dots \oplus s(k)z^k \oplus \dots$$

Le support d'une série formelle s est défini par :

$$\text{Supp}(s) = \{k \in \mathbb{Z} \mid s(k) \neq \varepsilon\}.$$

Remarque. Les séries qui ont un support fini sont appelées polynômes. Les séries singulaires sont appelées monômes.

Définition 5.19 (Dioïde de séries formelles). L'ensemble des séries formelles à une indéterminée z , à coefficients dans \mathcal{D} et munies des opérations suivantes :

$$s_a \oplus s_b : (s_a \oplus s_b)(k) = s_a(k) \oplus s_b(k), \quad (5.16)$$

$$s_a \otimes s_b : (s_a \otimes s_b)(k) = \bigoplus_{k=k_a+k_b} s_a(k_a) \otimes s_b(k_b), \quad (5.17)$$

forme le dioïde $\mathcal{D}[[z]]$. L'ensemble des séries formelles à p indéterminées commutatives, de z_1 à z_p , à coefficients dans \mathcal{D} et munies des opérations (5.16) et (5.17), forme le dioïde de séries formelles noté $\mathcal{D}[[z_1 \dots z_p]]$.

Exemple 5.4 (Dioïde $\mathbb{B}[[\gamma, \delta]]$). $\mathbb{B}[[\gamma, \delta]]$ est un dioïde complet, commutatif et distributif de séries formelles à deux indéterminées commutatives γ et δ , à coefficients booléens $\{\varepsilon, e\}$ et à exposant dans \mathbb{Z} . Chaque élément de $\mathbb{B}[[\gamma, \delta]]$ est défini comme suit :

$$s = \bigoplus_{n, t \in \mathbb{Z}} s(n, t) \gamma^n \delta^t \text{ avec } s(n, t) = e \text{ ou } \varepsilon.$$

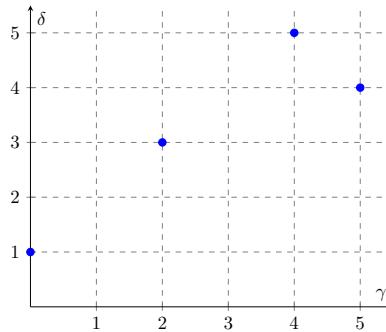
Dans ce dioïde :

- la loi additive \oplus a comme élément neutre $\varepsilon = \varepsilon(\gamma, \delta) = \bigoplus_{k, t \in \mathbb{Z}} \varepsilon \gamma^k \delta^t$,
- la loi multiplicative \otimes a comme élément neutre $e = e(\gamma, \delta) = \gamma^0 \delta^0$.

Les règles de calcul de la loi additive et multiplicative sont dans la définition 5.19.

La représentation des séries du dioïde $\mathbb{B}[[\gamma, \delta]]$ se fait par une représentation graphique dans le plan \mathbb{Z}^2 avec pour axe horizontal γ et pour axe vertical δ . Une série de $\mathbb{B}[[\gamma, \delta]]$ est décrite par les coordonnées (n, t) de chaque monôme de la série, où n correspond à l'exposant de γ et t correspond à l'exposant de δ . Les coefficients booléens $s(n, t)$ traduisent la présence du point $(s(n, t) = e)$ ou l'absence de point $(s(n, t) = \varepsilon)$.

Exemple 5.5. Soit une série $p = \gamma^0 \delta^1 \oplus \gamma^2 \delta^3 \oplus \gamma^4 \delta^5 \oplus \gamma^5 \delta^4 \in \mathbb{B}[[\gamma, \delta]]$, sa représentation graphique est donnée par la figure 5.3. Sur la figure, le point de coordonnées $(0, 1)$ correspond au monôme $\gamma^0 \delta^1$, celui de coordonnées $(2, 3)$ au monôme $\gamma^2 \delta^3 \dots$

FIGURE 5.3 – Représentation graphique de $p = \gamma^0 \delta^1 \oplus \gamma^2 \delta^3 \oplus \gamma^4 \delta^5 \oplus \gamma^5 \delta^4 \in \mathbb{B}[[\gamma, \delta]]$

Dioïde quotient

Définition 5.20 (Homomorphisme). Une application $\Pi : \mathcal{D} \rightarrow \mathcal{C}$ (avec \mathcal{D} et \mathcal{C} des dioïdes) est un homomorphisme si :

$$\forall a, c \in \mathcal{D}, \begin{cases} \Pi(a \oplus c) = \Pi(a) \oplus \Pi(c) & \text{et} & \Pi(\varepsilon) = \varepsilon, \\ \Pi(a \otimes c) = \Pi(a) \otimes \Pi(c) & \text{et} & \Pi(e) = e. \end{cases}$$

Remarque. Une application qui respecte $\Pi(a \oplus c) = \Pi(a) \oplus \Pi(c)$ est dite un \oplus -morphisme et une application qui respecte $\Pi(a \otimes c) = \Pi(a) \otimes \Pi(c)$ est dite un \otimes -morphisme.

Définition 5.21 (Relation d'équivalence). Une relation d'équivalence est une relation notée \equiv qui est réflexive ($x \equiv x$), transitive (si $x \equiv y$ et $y \equiv z$ alors $x \equiv z$) et symétrique (si $x \equiv y$ alors $y \equiv x$).

Définition 5.22 (Congruence). Dans un dioïde \mathcal{D} , une congruence est une relation d'équivalence \equiv compatible avec les opérations \oplus, \otimes soit $\forall a, b, c \in \mathcal{D}$

$$a \equiv b \Rightarrow \begin{cases} a \oplus c \equiv b \oplus c, \\ a \otimes c \equiv b \otimes c, \\ c \otimes a \equiv c \otimes b. \end{cases}$$

Définition 5.23 (Classe d'équivalence). Soit un dioïde \mathcal{D} muni d'une congruence \equiv , la classe d'équivalence d'un élément $a \in \mathcal{D}$ est notée $[a]_\equiv$ et correspond à l'ensemble des éléments équivalents à a selon \equiv :

$$[a]_\equiv = \{x \in \mathcal{D} | x \equiv a\}.$$

Définition 5.24 (Dioïde quotient). Soit \mathcal{D} un dioïde et \equiv une congruence, le quotient de \mathcal{D} par \equiv est le dioïde quotient noté \mathcal{D}_\equiv constitué des lois internes \oplus et \otimes définies par :

$$\begin{aligned} [a]_\equiv \oplus [b]_\equiv &= [a \oplus b]_\equiv, \\ [a]_\equiv \otimes [b]_\equiv &= [a \otimes b]_\equiv. \end{aligned}$$

Nous allons maintenant définir le dioïde $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ qui est au cœur de notre étude. $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ est un dioïde quotient de $\mathbb{B}[[\gamma, \delta]]$.

5.2.3 Définition et propriétés de $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$

Définition 5.25 (Dioïde $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$). $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ est le quotient du dioïde $\mathbb{B}[[\gamma, \delta]]$ par la congruence $\gamma^*(\delta^{-1})^*$ (l'opérateur $*$ est l'étoile de Kleene définie en 5.17) tel que $\forall a, b \in \mathbb{B}[[\gamma, \delta]]$:

$$a \equiv b \Leftrightarrow \gamma^*(\delta^{-1})^* a = \gamma^*(\delta^{-1})^* b.$$

Le dioïde $\mathcal{M}_{in}^{ax}[[\gamma, \delta]] = \mathbb{B}[[\gamma, \delta]] / \gamma^*(\delta^{-1})^*$ est un dioïde complet, commutatif, distributif et ses lois internes sont les mêmes que celles de $\mathbb{B}[[\gamma, \delta]]$. De plus :

- la loi additive \oplus a comme élément neutre $\varepsilon = \varepsilon(\gamma, \delta)$ (élément neutre de $\mathbb{B}[[\gamma, \delta]]$),
- la loi multiplicative \otimes a comme élément neutre¹ $e = \gamma^*(\delta^{-1})^* = \gamma^* = (\delta^{-1})^* = \gamma^0 = \delta^0 = \gamma^0 \delta^0$,

1. Les représentations de e sont équivalentes modulo $\gamma^*(\delta^{-1})^*$ mais ne sont pas identiques en tant qu'éléments de $\mathbb{B}[[\gamma, \delta]]$.

- le plus grand élément est $\top = (\gamma^{-1}\delta)^*$.

Un élément du dioïde $\mathcal{M}_{in}^{ax}[\gamma, \delta] = \mathbb{B}[\gamma, \delta]_{/\gamma^*(\delta^{-1})^*}$ est donc une classe d'équivalence notée $[a]_{\gamma^*(\delta^{-1})^*}$ qui regroupe tous les éléments de $\mathbb{B}[\gamma, \delta]$ équivalents entre eux modulo $\gamma^*(\delta^{-1})^*$.

Remarque. Les notations suivantes pour le plus petit élément et le plus grand élément de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ peuvent être utilisées : $\varepsilon = \gamma^{+\infty}\delta^{-\infty}$ et $\top = \gamma^{-\infty}\delta^{+\infty}$. Afin de simplifier la manipulation de ce dioïde, la désignation d'un élément de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ se fait fréquemment sans les crochets $[.]_{\gamma^*(\delta^{-1})^*}$.

La manipulation des éléments de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ se fait avec les règles de calcul sur l'addition et la multiplication du dioïde $\mathbb{B}[\gamma, \delta]$ à partir desquelles les règles de calcul suivantes sont établies :

$$\begin{aligned}\gamma^n\delta^t \oplus \gamma^{n'}\delta^{t'} &= \gamma^{\min(n, n')} \delta^t, \\ \gamma^n\delta^t \oplus \gamma^n\delta^{t'} &= \gamma^n \delta^{\max(t, t')}, \\ \gamma^n\delta^t \otimes \gamma^{n'}\delta^{t'} &= \gamma^{n'+n}\delta^{t'+t}.\end{aligned}\tag{5.18}$$

De plus, la relation d'ordre \preceq sur $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ conduit à l'équivalence suivante :

$$\gamma^n\delta^t \preceq \gamma^{n'}\delta^{t'} \Leftrightarrow n \geq n' \text{ et } t \leq t'.$$

Le dioïde $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ étant complet, tout couple $a, b \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$ admet comme plus grand minorant $a \wedge b$. Lorsque $a = \gamma^n\delta^t$ et $b = \gamma^{n'}\delta^{t'}$ (ce sont des monômes) on obtient la relation :

$$\gamma^n\delta^t \wedge \gamma^{n'}\delta^{t'} = \gamma^{\max(n, n')} \delta^{\min(t, t')}$$

qui permet, puisque le dioïde est distributif, d'établir la borne inférieure de deux éléments quelconques de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$.

La représentation graphique d'une série de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ se base sur la représentation graphique de $\mathbb{B}[\gamma, \delta]$. Graphiquement le produit des monômes par $\gamma^*(\delta^{-1})^*$ correspond à un cône de sommet de coordonnées (n, t) (alors qu'un élément de $\mathbb{B}[\gamma, \delta]$ correspond à un point de coordonnées (n, t)). L'élément γ^* permet au monôme $\gamma^n\delta^t$ d'être représenté de l'origine n jusqu'à $\gamma^{+\infty}$ sur l'axe horizontal. L'élément $(\delta^{-1})^*$ permet au monôme $\gamma^n\delta^t$ d'être représenté de l'origine t jusqu'à $\delta^{-\infty}$ sur l'axe vertical.

Exemple 5.6. Soit la série $p = \gamma^0\delta^1 \oplus \gamma^2\delta^3 \oplus \gamma^4\delta^5 \oplus \gamma^5\delta^4 \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$ représentée par la figure 5.4. La congruence $\gamma^*(\delta^{-1})^*$ permet au point de coordonnées $(0, 1)$ d'être prolongé sur les axes horizontal et vertical pour représenter le cône de la classe d'équivalence du monôme $\gamma^0\delta^1$. Les autres cônes sont issus des autres monômes de la série. Le cône du monôme $\gamma^5\delta^4$ est contenu par le cône du monôme $\gamma^4\delta^5$.

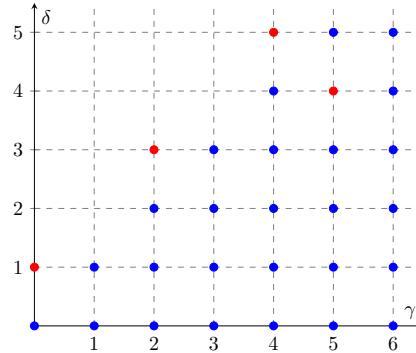


FIGURE 5.4 – Représentation graphique de $p = \gamma^0\delta^1 \oplus \gamma^2\delta^3 \oplus \gamma^4\delta^5 \oplus \gamma^5\delta^4 \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$

Définition 5.26 (Représentant maximal). Tout élément $a \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$ vérifie l'égalité suivante $a = \gamma^*(\delta^{-1})^*a$ où $\gamma^*(\delta^{-1})^*a$ correspond au représentant maximal de a . Graphiquement, cette représentation maximale correspond à tous les points du plan situés dans les cônes des monômes $\gamma^{n_i}\delta^{t_i}$.

Définition 5.27 (Représentant minimal). Le représentant minimal d'un élément $a \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$ correspond graphiquement aux sommets des cônes du représentant maximal de a .

Exemple 5.7. Soit la série $p = \gamma^0\delta^1 \oplus \gamma^2\delta^3 \oplus \gamma^4\delta^5 \oplus \gamma^5\delta^4 \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$. Le représentant maximal de cette série est $p_{max} = (\gamma^0\delta^1 \oplus \gamma^2\delta^3 \oplus \gamma^4\delta^5 \oplus \gamma^5\delta^4)(\gamma^*(\delta^{-1})^*)$ illustré graphiquement sur la figure 5.5 par les points rouges et bleus. Sur cette même figure, les points rouges (sommets des cônes) de coordonnées $(0, 1), (2, 3), (4, 5)$ sont les éléments du représentant minimal $p_{min} = \gamma^0\delta^1 \oplus \gamma^2\delta^3 \oplus \gamma^4\delta^5$.

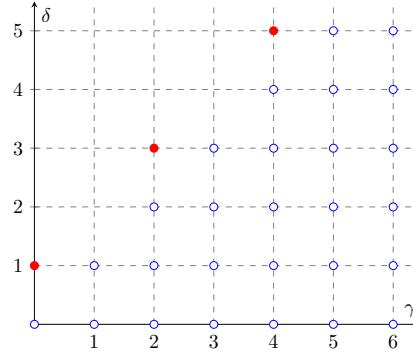


FIGURE 5.5 – Représentants maximal p_{max} et minimal p_{min} de la série $p \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$

Remarque. Par la suite, la représentation minimale des séries de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ est choisie pour présenter les résultats dans le manuscrit.

Les opérations \oplus , \otimes et \wedge de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ sont illustrées graphiquement par les règles suivantes :

- la somme $\gamma^n\delta^t \oplus \gamma^{n'}\delta^{t'}$ correspond à l'union des cônes de sommets (n, t) et (n', t') (voir figure 5.6a),
- la multiplication $\gamma^n\delta^t \otimes \gamma^{n'}\delta^{t'} = \gamma^{n+n'}\delta^{t+t'}$ correspond à la somme des coordonnées des sommets (n, t) et (n', t') (voir figure 5.6b),
- la borne inférieure $\gamma^n\delta^t \wedge \gamma^{n'}\delta^{t'} = \gamma^{\max(n, n')}\delta^{\min(t, t')}$ correspond à l'intersection des coordonnées des cônes de sommets (n, t) et (n', t') (voir figure 5.6c).

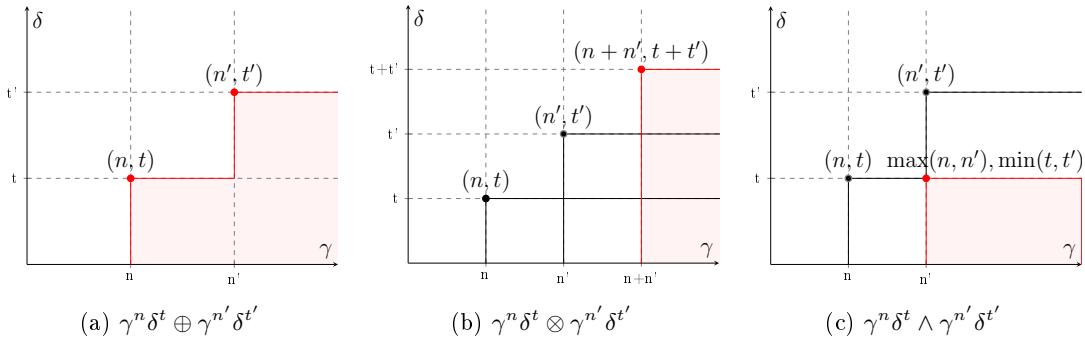


FIGURE 5.6 – Représentation graphique des opérations sur $\mathcal{M}_{in}^{ax}[\gamma, \delta]$

Deux autres représentations des séries de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ sont utilisées dans les calculs sur les séries : la fonction dateur et la fonction compteur.

Définition 5.28 (Fonction dateur). Soit s une série de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$, la *fonction dateur* de s est la fonction non-décroissante $\mathcal{D}_s(n) : \mathbb{Z} \rightarrow \mathbb{Z}$ telle que :

$$s = \bigoplus_{n \in \mathbb{Z}} \gamma^n \delta^{\mathcal{D}_s(n)}.$$

Cette fonction est obtenue à partir de la représentation maximale de s et n'en conserve, pour chaque γ^n de la série, que le plus grand δ^t correspondant ($t = \mathcal{D}_s(n)$).

Exemple 5.8. La série $p = \gamma^0 \delta^1 \oplus \gamma^2 \delta^3 \oplus \gamma^4 \delta^5 \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$ a pour fonction dateur $D_p(n) = -\infty$ pour $n \leq -1$, $D_p(0) = 1$, $D_p(1) = 1$, $D_p(2) = 3$, $D_p(3) = 3$ et $D_p(n) = 5$ pour $n \geq 4$. Cette fonction dateur énumère toutes les informations liées au temps de la série et est illustrée par la figure 5.7.

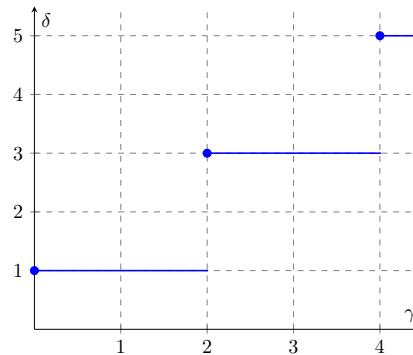


FIGURE 5.7 – Fonction dateur $D_p(n)$ de la série $p \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$

Exemple 5.9. Une série particulière sera utile pour le reste du manuscrit, il s'agit des séries avec $\gamma^n \delta^{+\infty}$ comme dernier monôme de la représentation minimale². Dans ce cas, la série $a = \gamma^0 \delta^1 \oplus \gamma^2 \delta^3 \oplus \gamma^4 \delta^5 \oplus \gamma^5 \delta^{+\infty} \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$ a pour fonction dateur $D_a(n) = -\infty$ pour $n \leq -1$, $D_a(0) = 1$, $D_a(1) = 1$, $D_a(2) = 3$, $D_a(3) = 3$, $D_a(4) = 5$ et $D_a(n) = +\infty$ pour $n \geq 5$.

Définition 5.29 (Fonction compteur). Soit s une série de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$, la *fonction compteur* de s est la fonction non-décroissante $\mathcal{C}_s(t) : \mathbb{Z} \rightarrow \mathbb{Z}$ telle que :

$$s = \bigoplus_{t \in \mathbb{Z}} \gamma^{\mathcal{C}_s(t)} \delta^t.$$

Cette fonction est obtenue à partir de la représentation maximale de s et n'en conserve, pour chaque δ^t de la série, que le plus petit γ^n correspondant ($n = \mathcal{C}_s(t)$).

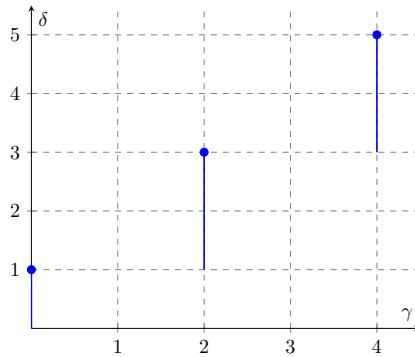
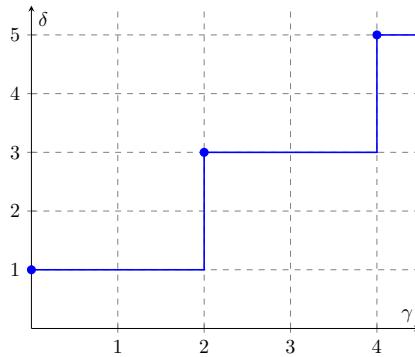
Exemple 5.10. La série $p = \gamma^0 \delta^1 \oplus \gamma^2 \delta^3 \oplus \gamma^4 \delta^5 \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$ a pour fonction compteur $\mathcal{C}_p(t) = 0$ pour $t \leq 1$, $\mathcal{C}_p(2) = 2$, $\mathcal{C}_p(3) = 2$, $\mathcal{C}_p(4) = 4$, $\mathcal{C}_p(5) = 4$ et $\mathcal{C}_p(t) = +\infty$ pour $t \geq 6$. Cette fonction compteur énumère toutes les informations liées aux événements de la série et est illustrée par la figure 5.8.

Exemple 5.11. La série $a = \gamma^0 \delta^1 \oplus \gamma^2 \delta^3 \oplus \gamma^4 \delta^5 \oplus \gamma^5 \delta^{+\infty} \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$ a pour fonction compteur $\mathcal{C}_a(t) = 0$ pour $t \leq 1$, $\mathcal{C}_a(2) = 2$, $\mathcal{C}_a(3) = 2$, $\mathcal{C}_a(4) = 4$, $\mathcal{C}_a(5) = 4$ et $\mathcal{C}_a(t) = 5$ pour $t \geq 6$.

Par la suite, pour représenter graphiquement les séries de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$, l'association des fonctions dateur et compteur de la série est utilisée comme le montre la figure 5.9 pour la série $p = \gamma^0 \delta^1 \oplus \gamma^2 \delta^3 \oplus \gamma^4 \delta^5$.

Par ailleurs, nous aurons besoin régulièrement de représenter les séries de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ selon une forme canonique qui énumère les monômes de la représentation minimale par ordre croissant des exposants de γ et de δ .

2. Dans un contexte de modélisation de systèmes à événements discrets, ces monômes représentent l'information que les événements à partir de n n'ont jamais lieu. Ces séries sont typiquement utilisées pour représenter des entrées et/ou des sorties finies de produits.

FIGURE 5.8 – Fonction compteur $C_p(t)$ de la série $p \in \mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ FIGURE 5.9 – Représentation de $p \in \mathcal{M}_{in}^{ax}[[\gamma, \delta]]$

Définition 5.30 (Représentation canonique). La représentation canonique de $s \in \mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ est :

$$s = \bigoplus_{k=0}^K \gamma^{n_k} \delta^{t_k} \text{ avec } K \in \mathbb{N} \cup \{+\infty\} \text{ et } \begin{cases} n_0 < n_1 < \dots \\ t_0 < t_1 < \dots \end{cases}.$$

Exemple 5.12. Soit la série $p = \gamma^3 \delta^5 \oplus \gamma^6 \delta^8 \oplus \gamma^1 \delta^2 \in \mathcal{M}_{in}^{ax}[[\gamma, \delta]]$, sa représentation canonique est $p = \gamma^1 \delta^2 \oplus \gamma^3 \delta^5 \oplus \gamma^6 \delta^8 \in \mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ avec $n_0 = 1, t_0 = 2, n_1 = 3, t_1 = 5$ et $n_2 = 6, t_2 = 8$.

Enfin, la définition suivante caractérise la causalité des séries de $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$.

Définition 5.31 (Causalité). Une série $p \in \mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ est dite causale si $p = \varepsilon$ (la série nulle) ou si chacun de ses monômes est à exposants dans \mathbb{N} . Une matrice de $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ est dite causale si toutes ses composantes sont causales.

Exemple 5.13. La série $p = \gamma^0 \delta^{-2} \in \mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ n'est pas causale car l'exposant de δ de p n'est pas dans \mathbb{N} tandis que la série $p = \gamma^0 \delta^1 \in \mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ est causale (l'exposant de δ de p est dans \mathbb{N}).

5.3 Théorie de la résiduation appliquée aux dioïdes

La plupart des applications définies sur les dioïdes ne sont pas inversibles. La théorie de la résiduation permet de considérer des "pseudo-inverses" d'applications isotones afin d'obtenir des solutions optimales pour des équations du type $\Pi(x) = b$. En particulier, nous nous intéressons aux plus grandes solutions des inéquations $\Pi(x) \preceq b$.

5.3.1 Théorie de la résiduation

La notion de continuité caractérise les applications où l'ordre dans l'image est effectif et préservée.

Définition 5.32 (Continuité). Soit une application $\Pi : \mathcal{D} \rightarrow \mathcal{C}$ où \mathcal{D} et \mathcal{C} sont deux dioïdes complets, l'application Π est dite semi-continue inférieurement si pour tout sous-ensemble $\mathcal{B} \subset \mathcal{D}$,

$$\Pi\left(\bigoplus_{x \in \mathcal{B}} x\right) = \bigoplus_{x \in \mathcal{B}} (\Pi(x)).$$

L'application Π est dite semi-continue supérieurement si pour tout sous-ensemble $\mathcal{B} \subset \mathcal{D}$,

$$\Pi\left(\bigwedge_{x \in \mathcal{B}} x\right) = \bigwedge_{x \in \mathcal{B}} (\Pi(x)).$$

Définition 5.33 (Application résiduable). Une application isotone $\Pi : \mathcal{D} \rightarrow \mathcal{C}$ définie sur des dioïdes complets est dite résiduable pour tout $b \in \mathcal{C}$, si l'inéquation $\Pi(x) \preceq b$ admet une plus grande solution dans \mathcal{D} . La plus grande solution de $\Pi(x) \preceq b$ est notée Π^\sharp et est appelée la résiduée de l'application Π :

$$\Pi^\sharp(b) = \bigoplus\{x \in \mathcal{D} \mid \Pi(x) \preceq b\}.$$

La caractérisation de ces applications résiduables est donnée par le théorème suivant.

Théorème 5.3 ([BCOQ92]). *Soit une application isotone $\Pi : \mathcal{D} \rightarrow \mathcal{C}$ définie sur des dioïdes complets, les équivalences suivantes sont obtenues :*

- Π est résiduable,
- Π est semi-continue inférieurement et $\Pi(\varepsilon) = \varepsilon$,
- il existe une application isotone et semi-continue supérieurement $\Pi^\sharp : \mathcal{C} \rightarrow \mathcal{D}$ telle que :
 - $\Pi \circ \Pi^\sharp \preceq I_{dC}$ où I_{dC} correspond à l'application identité dans \mathcal{C} ,
 - $\Pi^\sharp \circ \Pi \succeq I_{dD}$ où I_{dD} correspond à l'application identité dans \mathcal{D} .

On en déduit que Π^\sharp est unique.

Théorème 5.4 ([Cot99]). *Soit deux applications résiduables définies sur des dioïdes complets : $\Pi : \mathcal{D} \rightarrow \mathcal{C}$ et $f : \mathcal{C} \rightarrow \mathcal{G}$:*

- $\Pi \circ \Pi^\sharp \circ \Pi = \Pi$ et $\Pi^\sharp \circ \Pi \circ \Pi^\sharp = \Pi^\sharp$,
- $(f \circ \Pi)^\sharp = \Pi^\sharp \circ f^\sharp$,
- Π est injective $\Leftrightarrow \Pi^\sharp \circ \Pi = I_{dD} \Leftrightarrow \Pi$ est surjective,
- Π^\sharp est surjective $\Leftrightarrow \Pi \circ \Pi^\sharp = I_{dC} \Leftrightarrow \Pi^\sharp$ est injective.

L'application de l'injection canonique sur les dioïdes est définie ci-dessous.

Définition 5.34 (Injection canonique dioïde). Soit \mathcal{C} un sous-dioïde du dioïde \mathcal{D} , avec \mathcal{C} et \mathcal{D} complets, l'injection canonique de \mathcal{C} dans \mathcal{D} est l'application notée $I_{\mathcal{C}}$ et définie comme suit pour tout $x \in \mathcal{C}$:

$$I_{\mathcal{C}}(x) = x.$$

Théorème 5.5 (Lemme de projection [Gau92]). *Soit $\mathcal{C} \subset \mathcal{D}$, avec \mathcal{C} et \mathcal{D} des dioïdes complets, l'injection canonique $I_{\mathcal{C}} : \mathcal{C} \rightarrow \mathcal{D}$ est résiduable. Sa résiduée $I_{\mathcal{C}}^\sharp$ est notée $Pr_{\mathcal{C}}$ et vérifie :*

- $Pr_{\mathcal{C}} \circ Pr_{\mathcal{C}} = Pr_{\mathcal{C}}$,
- $Pr_{\mathcal{C}} \preceq I_{dD}$,
- $a \in \mathcal{C} \Leftrightarrow Pr_{\mathcal{C}}(a) = a$,

Dans la suite du manuscrit, la projection causale d'une série de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ est utilisée (voir définition 5.31 de la causalité), elle est définie ci-dessous.

Définition 5.35 (Projection causale d'une série de $\mathcal{M}_{in}^{ax}[\gamma, \delta]^+$). Soit $\mathcal{M}_{in}^{ax}[\gamma, \delta]^+$ le sous-dioïde des séries causales de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$. L'injection canonique I^+ définie de $\mathcal{M}_{in}^{ax}[\gamma, \delta]^+$ dans $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ est résiduable et sa résiduée est notée $Pr^+ : \mathcal{M}_{in}^{ax}[\gamma, \delta] \mapsto \mathcal{M}_{in}^{ax}[\gamma, \delta]^+$. Formellement, la série $Pr^+(s)$ est la plus grande série causale plus petite ou égale à s :

$$Pr^+(s) = Pr^+(\bigoplus_{i \in \mathbb{N}} f(n_i, t_i) \gamma^{n_i} \delta^{t_i}) = \bigoplus_{i \in \mathbb{N}} g(n_i, t_i) \gamma^{n_i} \delta^{t_i},$$

pour laquelle :

$$g(n_i, t_i) = \begin{cases} f(n_i, t_i) & \text{si } (n_i, t_i) \geq (0, 0), \\ \varepsilon & \text{sinon.} \end{cases}$$

Remarque. Graphiquement, la projection causale consiste à ne conserver que les monômes situés dans la partie positive du plan \mathbb{Z}^2 . Si une série x est causale, alors elle est invariante par Pr^+ , soit $Pr^+(x) = x$.

5.3.2 Résolution d'application par la théorie de la résiduation

Soient L_a le produit à gauche et R_a le produit à droite définis sur le dioïde complet \mathcal{D} tels que :

$$\begin{aligned} L_a : x &\mapsto a \otimes x, \\ R_a : x &\mapsto x \otimes a. \end{aligned}$$

Comme les applications sont définies sur un dioïde complet, le produit est distributif à gauche comme à droite sur les sommes infinies (voir définition 5.14), donc les applications sont semi-continues inférieurement. De plus, ε est absorbant pour le produit, ainsi $L_a(\varepsilon) = a \otimes \varepsilon = \varepsilon$ et $R_a(\varepsilon) = \varepsilon \otimes a = \varepsilon$. Donc les applications L_a et R_a sont résiduables (d'après le théorème 5.3) et les applications résiduées sont notées :

$$\begin{aligned} L_a^\sharp(x) &= a \backslash x, \\ R_a^\sharp(x) &= x \not a. \end{aligned}$$

$L_a^\sharp(x)$ est la plus grande solution de l'inéquation $ax \preceq b$ et $R_a^\sharp(x)$ est la plus grande solution de l'inéquation $xa \preceq b$:

$$\begin{aligned} a \backslash x &= \bigoplus \{x \mid ax \preceq b\}, \\ x \not a &= \bigoplus \{x \mid xa \preceq b\}. \end{aligned}$$

On peut alors directement écrire :

$$\begin{aligned} ax \preceq b &\Leftrightarrow x \preceq a \backslash b, \\ xa \preceq b &\Leftrightarrow x \preceq b \not a. \end{aligned}$$

Remarque. Si le dioïde \mathcal{D} complet est commutatif, alors $L_a = R_a$ ainsi que $L_a^\sharp = R_a^\sharp$.

De nombreuses propriétés peuvent être associées aux applications résiduées L_a^\sharp, R_a^\sharp .

Théorème 5.6 ([BCOQ92] et [Cot99]). *Les applications $L_a^\sharp(x)$ et $R_a^\sharp(x)$ vérifient les propriétés suivantes $\forall a, b, x, y \in \mathcal{D}$, avec \mathcal{D} complet :*

$$a \backslash (x \wedge y) = a \backslash x \wedge a \backslash y, \quad (x \wedge y) \not a = x \not a \wedge y \not a, \quad (5.19)$$

$$(a \oplus b) \backslash x = a \backslash x \wedge b \backslash x, \quad x \not (a \oplus b) = x \not a \wedge x \not b, \quad (5.20)$$

$$(ab) \backslash x = b \backslash (a \backslash x), \quad x \not (ab) = (x \not a) \not b. \quad (5.21)$$

S'il existe $b^{-1} \in \mathcal{D}$ tel que $bb^{-1} = b^{-1}b = e$ alors

$$b \backslash a = b^{-1}a, \quad a \not b = ab^{-1}. \quad (5.22)$$

Les applications L_a et R_a peuvent être définies pour le produit de matrices telles que :

$$\begin{aligned} L_A : \mathcal{D}^{p \times m} &\rightarrow \mathcal{D}^{n \times m} \\ X &\mapsto A \otimes X, \quad \text{avec } A \in \mathcal{D}^{n \times p}, \\ R_A : \mathcal{D}^{q \times p} &\rightarrow \mathcal{D}^{q \times n} \\ X &\mapsto X \otimes A, \quad \text{avec } A \in \mathcal{D}^{p \times n}. \end{aligned}$$

Les applications résiduées de L_A et R_A sont alors :

$$L_A^\sharp(B) = A \setminus B : (A \setminus B)_{ij} = \bigwedge_{k=1}^n A_{ki} \setminus B_{kj} \quad \text{avec } A \in \mathcal{D}^{n \times p} \text{ et } B \in \mathcal{D}^{n \times m},$$

$$R_A^\sharp(B) = B \not\setminus A : (B \not\setminus A)_{ij} = \bigwedge_{k=1}^n B_{ik} \not\setminus A_{jk} \quad \text{avec } A \in \mathcal{D}^{p \times n} \text{ et } B \in \mathcal{D}^{q \times n}.$$

Théorème 5.7 ([Max91]). *Soient $B, X \in \mathcal{D}^{n \times n}$ avec \mathcal{D} complet, les deux conditions suivantes sont équivalentes :*

- $X = B^*$,
- $X = B \not\setminus B$.

Ce résultat peut être lu comme "B $\not\setminus$ B est une étoile" ($B \not\setminus B = (B \not\setminus B)^*$).

5.4 Modélisation des systèmes $(\max, +)$ linéaires

Les systèmes $(\max, +)$ -linéaires permettent de modéliser les informations événementielles et temporelles des SED temporisés. Grâce à la théorie des dioïdes, ces systèmes peuvent être modélisés mathématiquement dans l'algèbre $(\max, +)$ et plus particulièrement dans le dioïde $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ ([BCOQ92], [Max91] et [CMQV89]). Graphiquement, ils sont représentés par des GET (Graphes d'Événements Temporisés, voir définition 4.1 de la section 4.1 du chapitre 4).

Un système $(\max, +)$ -linéaire est un système pouvant être représenté dans le dioïde $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ par la représentation d'état linéaire suivante :

$$\begin{cases} x = Ax \oplus Bu, \\ y = Cx, \end{cases} \quad (5.23)$$

où x est le vecteur d'état du système de dimension n , u le vecteur d'entrée de dimension p et y le vecteur de sortie de dimension q . La matrice $A \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{n \times n}$ représente les connexions entre les états internes, la matrice $B \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{n \times p}$ les connexions entre les états et les entrées et la matrice $C \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{q \times n}$ celles entre les états et les sorties.

Ces équations modélisent la structure du GET :

- Chaque entrée des matrices est un monôme $\gamma^n \delta^t$ qui correspond à une connexion entre deux transitions, c'est-à-dire la présence d'une place.
- Les transitions sont les vecteurs x, u, y et le tir d'une transition est représenté par un monôme $\gamma^n \delta^t$ indiquant que l'occurrence numéro n du tir la transition a lieu au plus tôt à la date t . L'historique de ces occurrences d'événements est appelé trajectoire et est contenu dans la série associée à cette transition.

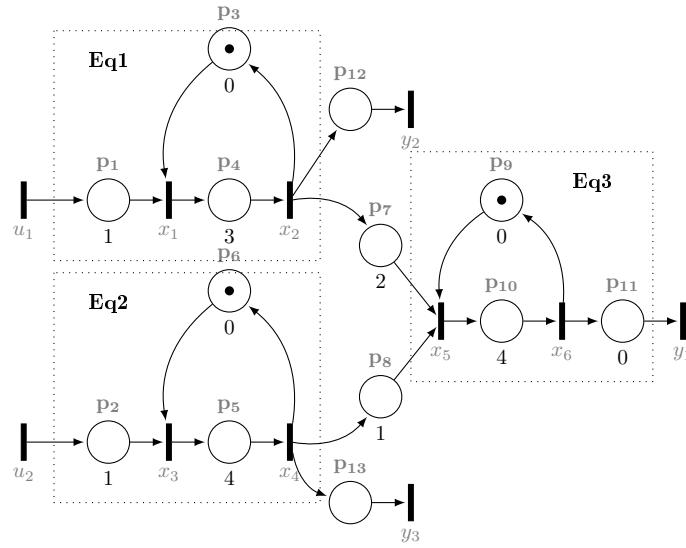
Les deux indéterminées du dioïde $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ représentent donc les deux dimensions du GET : γ pour le domaine événementiel (nombre de jetons présents dans le GET, numéros de tirs des transitions) et δ pour le domaine temporel (temps de séjour des jetons associés aux places, dates de tir des transitions).

L'équation $x = Ax \oplus B$ peut se transformer en $x = A^*Bu$ grâce au théorème 5.2 ce qui permet d'obtenir cette relation entrée/sortie :

$$y = CA^*Bu.$$

La matrice $H = CA^*B \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{q \times p}$ représente alors la fonction de transfert du GET, soit l'influence de ses entrées sur ses sorties. Il s'agit de la dynamique du système.

Exemple 5.14. Reprenons le GET de l'exemple 4.4 présenté dans la section 4.1 du chapitre 4. Ce GET est donné à nouveau dans la figure 5.10.

FIGURE 5.10 – Un GET avec 2 entrées u_1, u_2 et 3 sorties y, y_2, y_3

La représentation d'état de ce GET est :

$$\begin{cases} x = \begin{pmatrix} \cdot & \gamma^1\delta^0 & \cdot & \cdot & \cdot & \cdot \\ \gamma^0\delta^3 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \gamma^1\delta^0 & \cdot & \cdot \\ \cdot & \cdot & \gamma^0\delta^4 & \cdot & \cdot & \cdot \\ \cdot & \gamma^0\delta^2 & \cdot & \gamma^0\delta^1 & \cdot & \gamma^1\delta^0 \\ \cdot & \cdot & \cdot & \cdot & \gamma^0\delta^4 & \cdot \end{pmatrix} x \oplus \begin{pmatrix} \gamma^0\delta^1 & \cdot \\ \cdot & \gamma^0\delta^1 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{pmatrix} u, \\ y = \begin{pmatrix} \cdot & \gamma^0\delta^0 & \cdot & \cdot & \gamma^0\delta^0 \\ \cdot & \gamma^0\delta^0 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \gamma^0\delta^0 & \cdot & \cdot \end{pmatrix} x. \end{cases} \quad (5.24)$$

Dans la matrice A , $A(1, 2) = \gamma^1\delta^0$ signifie qu'une place, la place p_3 , est située entre la transition x_2 et x_1 . La temporisation de cette place est de 0 donc l'exposant de δ est 0 (δ^0) et la place contient un jeton donc l'exposant de γ est de 1 (γ^1). Dans la matrice C , $C(1, 6) = \gamma^0\delta^0$ signifie que le n^{eme} tir de y dépend du n^{eme} tir de x_6 . Vu que la durée associée à la place située entre y et x_6 (place p_{11}) est égale à 0, les dates d'occurrence de ces deux transitions sont exactement les mêmes.

La fonction de transfert H du système est :

$$H = \begin{pmatrix} \gamma^0\delta^{10}(\gamma^1\delta^4)^* & \gamma^0\delta^{10}(\gamma^1\delta^4)^* \\ \gamma^0\delta^4(\gamma^1\delta^3)^* & \gamma^0\delta^5(\gamma^1\delta^4)^* \end{pmatrix}$$

Si les flux d'événements associés aux transitions d'entrée u_1 et u_2 sont identiques et donnés par ces séries :

$$u_1 = u_2 = \gamma^0\delta^1 \oplus \gamma^1\delta^2 \oplus \gamma^2\delta^3 \oplus \gamma^3\delta^4 \oplus \gamma^4\delta^5 \oplus \gamma^5\delta^6 \oplus \gamma^6\delta^7 \oplus \gamma^7\delta^{+\infty},$$

les flux d'événements associés aux transitions de sortie y_1, y_2 et y_3 sont alors calculés grâce à ces entrées u et la fonction de transfert H :

$$\begin{aligned} y_1 &= \gamma^0\delta^{11} \oplus \gamma^1\delta^{15} \oplus \gamma^2\delta^{19} \oplus \gamma^3\delta^{23} \oplus \gamma^4\delta^{27} \oplus \gamma^5\delta^{31} \oplus \gamma^6\delta^{35} \oplus \gamma^7\delta^{+\infty}, \\ y_2 &= \gamma^0\delta^5 \oplus \gamma^1\delta^8 \oplus \gamma^2\delta^{11} \oplus \gamma^3\delta^{14} \oplus \gamma^4\delta^{17} \oplus \gamma^5\delta^{20} \oplus \gamma^6\delta^{23} \oplus \gamma^7\delta^{+\infty}, \\ y_3 &= \gamma^0\delta^6 \oplus \gamma^1\delta^{10} \oplus \gamma^2\delta^{14} \oplus \gamma^3\delta^{18} \oplus \gamma^4\delta^{22} \oplus \gamma^5\delta^{26} \oplus \gamma^6\delta^{30} \oplus \gamma^7\delta^{+\infty}. \end{aligned}$$

Dans ces séries, le monôme $\gamma^0\delta^1$ de la transition u_1 signifie que son premier tir numéroté 0 a lieu à $t = 1$. De plus, dans cet exemple, il n'y a que 7 occurrences d'événements pour ces transitions³. L'absence d'un huitième événement est indiquée par l'exposant $+\infty$ de l'indéterminée δ dans les monômes $\gamma^7\delta^{+\infty}$ de ces séries.

3. C'est également vrai pour les transitions internes dont nous ne donnons pas le détail.

Remarque. A $t = 0$, les jetons présents dans le GET sont considérés présents depuis "l'origine des temps", c'est-à-dire depuis la date $-\infty$. Donc les temps associés à ces places se sont déjà écoulés à $t = 0$.

5.5 Mesure des décalages temporels entre les séries $\mathcal{M}_{in}^{ax}[\gamma, \delta]$

Des comparaisons entre des séries de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ peuvent être faites grâce à la théorie de la résiduation définie dans la section 5.3. Ces comparaisons mesurent les décalages temporels entre deux séries.

Définition 5.36 (Fonction de décalage temporel). Soient $a, b \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$ et leurs fonctions dateurs respectives $\mathcal{D}_a(n)$ et $\mathcal{D}_b(n)$, la *fonction de décalage temporel* représentant tous les écarts de temps entre a et b pour chaque $n \in \mathbb{Z}$ est définie par :

$$\mathcal{T}_{a,b}(n) = \mathcal{D}_b(n) - \mathcal{D}_a(n).$$

Cette fonction compare les différences temporelles entre le n^{eme} événement de la série a et le n^{eme} événement de la série b . Si $\mathcal{T}_{a,b}(n) \geq 0$ alors le n^{eme} événement de b est en retard par rapport à celui de a . Si $\mathcal{T}_{a,b}(n) \leq 0$ alors le n^{eme} événement de b est en avance par rapport à celui de a . Quand $\mathcal{D}_b(n) = \mathcal{D}_a(n)$, il n'y a pas de décalage temporel pour ce n . Enfin, lorsque $\mathcal{T}_{a,b}(n)$ est égal à une valeur infinie on obtient ces résultats :

$$\begin{aligned} (-\infty) - \mathcal{D}(n) &= -\infty & (+\infty) - \mathcal{D}(n) &= +\infty, \\ \mathcal{D}(n) - (-\infty) &= +\infty & \mathcal{D}(n) - (+\infty) &= -\infty, \\ (-\infty) - (-\infty) &= +\infty & (+\infty) - (+\infty) &= +\infty. \end{aligned}$$

Exemple 5.15. Soient deux séries $a, b \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$ telles que $a = \gamma^0\delta^1 \oplus \gamma^1\delta^{+\infty}$ avec comme éléments de sa fonction dateur $\mathcal{D}_a(0) = 1$ et $\mathcal{D}_a(1) = +\infty$ et $b = \gamma^0\delta^2 \oplus \gamma^2\delta^{+\infty}$ avec $\mathcal{D}_b(0) = 2$, $\mathcal{D}_b(1) = 2$ et $\mathcal{D}_b(2) = +\infty$. Rappelons que ces séries représentent des trajectoires finies, c'est-à-dire que l'événement numéroté 1 de la série a ne se produit jamais, tout comme l'événement numéroté 2 de la série b . Pour $n = 1$ on a :

$$\mathcal{T}_{a,b}(1) = \mathcal{D}_b(1) - \mathcal{D}_a(1) = 2 - (+\infty) = -\infty.$$

Ce résultat représente le fait que la série a s'arrête avant la série b (b a une occurrence de plus que a).

Exemple 5.16. Soient deux séries $a, b \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$ telles que $a = \gamma^0\delta^3 \oplus \gamma^1\delta^{+\infty}$ avec $\mathcal{D}_a(0) = 3$ et $\mathcal{D}_a(1) = +\infty$ et $b = \gamma^1\delta^4 \oplus \gamma^2\delta^{+\infty}$ avec $\mathcal{D}_b(0) = -\infty$, $\mathcal{D}_b(1) = 4$ et $\mathcal{D}_b(2) = +\infty$. Pour $n = 0$ on a :

$$\mathcal{T}_{a,b}(0) = \mathcal{D}_b(0) - \mathcal{D}_a(0) = -\infty - 3 = -\infty.$$

Ce résultat représente pour ces séries le fait que la série b est décalée vers la droite sur l'axe horizontal d'un événement par rapport à a .

Théorème 5.8 ([Max91]). Soient $a, b \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$, la fonction de décalage temporel peut être bornée comme suit :

$$\forall n \in \mathbb{Z}, \quad \mathcal{D}_{b \neq a}(0) \leq \mathcal{T}_{a,b}(n) \leq -\mathcal{D}_{a \neq b}(0).$$

Ce théorème permet de déterminer les bornes minimales et maximales de la fonction de décalage temporel. L'extraction des bornes se fait sur les séries $a \neq b$ et $b \neq a$. La borne $\mathcal{D}_{b \neq a}(0)$ est obtenue avec la série $b \neq a$ où $\mathcal{D}_{b \neq a}(0)$ est contenue dans le monôme $\gamma^0\delta^{\mathcal{D}_{b \neq a}(0)} \in b \neq a$. Graphiquement, celle-ci est déterminée par l'intersection de $b \neq a$ avec l'axe du domaine temporel δ et l'observation de l'exposant de δ du monôme $\gamma^0\delta^{\mathcal{D}_{b \neq a}(0)}$. La borne $\mathcal{D}_{a \neq b}(0)$ est obtenue avec la série $a \neq b$ où $\mathcal{D}_{a \neq b}(0)$ est contenue dans le monôme $\gamma^0\delta^{\mathcal{D}_{a \neq b}(0)} \in a \neq b$. Graphiquement, celle-ci est déterminée par l'intersection de $a \neq b$ avec l'axe du domaine temporel δ et l'observation de l'exposant de δ du monôme $\gamma^0\delta^{\mathcal{D}_{a \neq b}(0)}$.

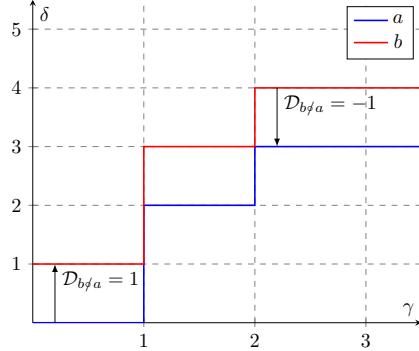
Exemple 5.17. Soient deux séries $a = \gamma^0\delta^0 \oplus \gamma^1\delta^2 \oplus \gamma^2\delta^3 \oplus \gamma^4\delta^{+\infty}$ et $b = \gamma^0\delta^1 \oplus \gamma^1\delta^3 \oplus \gamma^2\delta^4 \oplus \gamma^4\delta^{+\infty}$. Les calculs de $b \neq a$ et $a \neq b$ donnent les résultats suivants :

$$\begin{aligned} b \neq a &= \gamma^0\delta^1 \oplus \gamma^1\delta^2 \oplus \gamma^2\delta^4 \oplus \gamma^4\delta^{+\infty}, \\ a \neq b &= \gamma^0\delta^{-1} \oplus \gamma^1\delta^0 \oplus \gamma^2\delta^1 \oplus \gamma^4\delta^{+\infty}, \end{aligned}$$

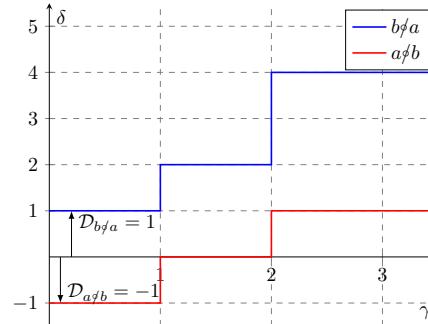
et donc :

$$\mathcal{D}_{b \not\sim a}(0) = 1 \quad -\mathcal{D}_{a \not\sim b}(0) = 1.$$

De la série $b \not\sim a$, on retire l'information $\mathcal{D}_{b \not\sim a}(0) = 1$ sur le monôme $\gamma^0 \delta^1$. De la série $a \not\sim b$ on retire l'information $-\mathcal{D}_{a \not\sim b}(0) = 1$ sur le monôme $\gamma^0 \delta^{-1}$. Les figures 5.11a et 5.11b représentent respectivement les séries a (en bleu) et b (en rouge) ainsi que les résiduations $b \not\sim a$ (en bleu) et $a \not\sim b$ (en rouge).



(a) Représentation des séries a et b



(b) Représentation des séries $b \not\sim a$ et $a \not\sim b$

FIGURE 5.11 – Représentation graphique des bornes du décalage temporel entre les séries a et b

Les bornes minimales et maximales entre deux séries du théorème 5.8 permettent de caractériser un intervalle de décalages temporels.

Définition 5.37 (Intervalle de décalages temporels). Soient $a, b \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$, les décalages temporels entre a et b sont inclus dans l'intervalle :

$$\Delta(a, b) = [\mathcal{D}_{b \not\sim a}(0); -\mathcal{D}_{a \not\sim b}(0)], \quad (5.25)$$

avec $\gamma^0 \delta^{\mathcal{D}_{b \not\sim a}(0)} \in b \not\sim a$ et $\gamma^0 \delta^{\mathcal{D}_{a \not\sim b}(0)} \in a \not\sim b$. Dans cet intervalle, la série de référence, c'est-à-dire la série à partir de laquelle le décalage temporel est mesuré, est la série a .

Propriété 5.1. *Lorsque $\mathcal{D}_{b \not\sim a}(0)$ et $-\mathcal{D}_{a \not\sim b}(0)$ de la définition 5.37 sont des valeurs finies, on a les propriétés graphiques suivantes :*

- Si les bornes sont positives

$$\mathcal{D}_{b \not\sim a}(0) \geq 0 \text{ et } -\mathcal{D}_{a \not\sim b}(0) \geq 0,$$

la série b est graphiquement au-dessus de la série a . Ainsi $a \preceq b$.

- Si les bornes sont négatives

$$\mathcal{D}_{b \not\sim a}(0) \leq 0 \text{ et } -\mathcal{D}_{a \not\sim b}(0) \leq 0,$$

la série b est graphiquement en-dessous de la série a . Ainsi $a \succeq b$.

- Si l'une des bornes est négative et l'autre positive

$$\mathcal{D}_{b \not\sim a}(0) \leq 0 \text{ et } -\mathcal{D}_{a \not\sim b}(0) \geq 0,$$

la série b se croise avec la série a . Ainsi les deux séries ne sont pas comparables.

Si l'une des bornes est infinie alors les propriétés changent.

Propriété 5.2. *Quand l'une des bornes de la définition 5.37 a une valeur infinie ($\mathcal{D}_{b \not\sim a}(0) = -\infty$ ou $-\mathcal{D}_{a \not\sim b}(0) = +\infty$) cela implique que l'un des exposants de γ d'une des séries n'existe pas dans l'autre. Un décalage d'événements s'est produit sur les occurrences d'événements entre a et b . On a alors :*

- Si $\mathcal{D}_{b \not\sim a}(0) \geq 0$ et $-\mathcal{D}_{a \not\sim b}(0) = +\infty$, alors $a \preceq b$.

- Si $\mathcal{D}_{b \not\sim a}(0) = -\infty$ et $\mathcal{D}_{a \not\sim b}(0) \leq 0$, alors $a \succeq b$.
- Si $\mathcal{D}_{b \not\sim a}(0) = -\infty$ et $\mathcal{D}_{a \not\sim b}(0) \geq 0$, ou si $\mathcal{D}_{b \not\sim a}(0) \leq 0$ et $-\mathcal{D}_{a \not\sim b}(0) = +\infty$ alors a et b ne sont pas comparables.
- Si $\mathcal{D}_{b \not\sim a}(0) = -\infty$ et $-\mathcal{D}_{a \not\sim b}(0) = +\infty$ alors les événements de l'une des deux séries commencent juste après le dernier événement de l'autre série.

Exemple 5.18. À partir de l'exemple 5.17, puisque $\Delta(a, b) = [\mathcal{D}_{b \not\sim a}(0); -\mathcal{D}_{a \not\sim b}(0)] = [1; 1]$, les bornes sont positives donc la série b est graphiquement au-dessus de la série a , ce qui donne bien $a \preceq b$.

5.6 Synthèse du chapitre

Ce chapitre présente un rappel mathématique utilisé notamment pour les définitions formelles des dioïdes et en particulier celle du dioïde $\mathcal{M}_{in}^{ax}[\gamma, \delta]$. Ce dioïde permet de modéliser les informations événementielles et temporelles des GET (Graphes d'Événements Temporisés) dans une représentation d'état. De la définition des séries de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$, une comparaison temporelle entre deux séries est établie grâce à la théorie de la résiduation.

Cette comparaison va permettre de déterminer les décalages temporels entre deux séries utilisées dans la suite du manuscrit pour faire de la détection et de la localisation des dérives temporelles dans un système via des observateurs ou de la détection de dérives temporelles dans des systèmes à temps intervalles.

Chapitre 6

Détection et localisation de dérives temporelles par des observateurs

Ce chapitre présente une méthode de détection et une analyse de la localisation de dérives temporelles (voir section 4.1 du chapitre 4) dans des systèmes (max+)-linéaires représentés par des Graphes d'Événements Temporisés (GET). Ce diagnostic est réalisé à l'aide d'observateurs (max,+) ([HMCSM10], [HMCL10], [Har04]) qui permettent d'estimer les états d'un système, c'est-à-dire les trajectoires des tirs des transitions internes d'un GET représentées par des séries de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$, soumis à des perturbations qui ralentissent le système, nouvelles entrées du GET. Les états estimés proposés dans ce manuscrit sont épurés des éléments ne prenant potentiellement pas en compte l'effet de la dérive, caractérisée par les entrées de perturbations. Ces états estimés sont appelés SAP pour *sensibles-aux-perturbations* et sont comparés aux états du modèle sans faute du système. Le résultat de cette comparaison fournit un indicateur qui renvoie vrai si la dérive est détectée. La localisation utilise les valeurs de l'intervalle obtenu au moment de la comparaison pour définir un ensemble de places candidates d'où peut venir la dérive. Cette localisation se fait par structure élémentaire de GET (tandem, parallélisme, synchronisation, boucle). Des exemples de localisation dans des GET complets sont également proposés.

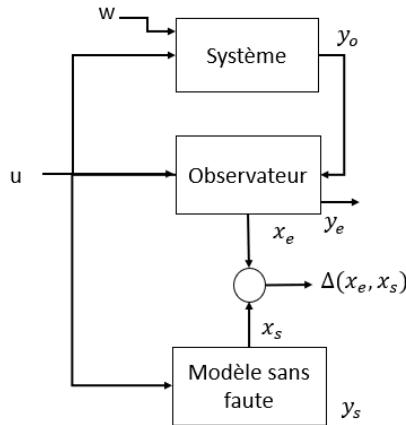


FIGURE 6.1 – Structure de la méthode de détection

La figure 6.1 présente plus en détail la méthode de détection utilisée dans ce chapitre, elle est définie par trois blocs : le *système perturbé*, c.-à-d. le système dans lequel la dérive temporelle se produit, l'*observateur* et le modèle du système sans la dérive, également appelé *modèle nominal* ou *modèle sans faute*. Sur la figure, u est l'entrée du système, w est l'entrée de perturbations, y_o est la sortie du système perturbé, y_e est la sortie estimée grâce à l'observateur et y_s est la sortie du modèle sans faute. Les états estimés sont notés x_e , ceux du modèle sans faute x_s . L'entrée u et la sortie y_o sont connues car mesurées lors du fonctionnement du système, le modèle sans faute est la modélisation GET du fonctionnement du système. L'étape de détection compare dans un intervalle noté $\Delta(x_e, x_s)$ les états x_e avec les états x_s grâce à la résiduation de séries de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$. Les informations de cet intervalle sont celles utilisées pour la localisation de la dérive. Tout au long

du chapitre, le terme dérive temporelle désigne une *dérive temporelle détectable* selon la définition 4.4, page 57. De plus, les perturbations qui vont caractériser nos dérives temporelles étant des entrées qui ne peuvent que ralentir le système, nous ne pourrons détecter que des dérives de type retard.

6.1 Synthèse d'un observateur de systèmes $(\max, +)$ -linéaires avec perturbations

Cette section présente la synthèse d'un observateur de systèmes $(\max, +)$ -linéaires soumis à des perturbations telle que représentée sur la figure 6.2. Cette synthèse est issue des documents suivants [HMCSM10], [Har04] et est appliquée aux GET. Un observateur d'un système $(\max, +)$ -linéaire estime l'état interne x_e du système réel en *mesurant* son entrée u et sa sortie y_o . Le système réel est soumis à des perturbations *inconnues* représentées par des entrées spécifiques w et l'estimation x_e est la plus proche possible de l'état x du système réel. Deux nouvelles matrices sont à prendre en compte dans les représentations d'état du système et de l'observateur : la matrice de perturbation R et la matrice d'observation L .

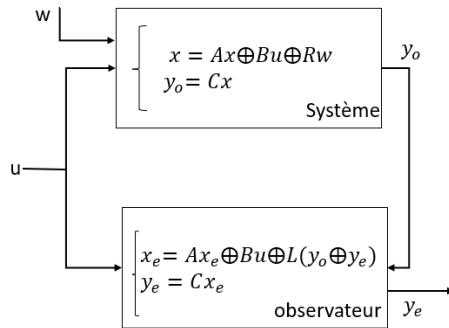


FIGURE 6.2 – Structure d'un observateur d'un système $(\max, +)$ -linéaire soumis à des perturbations

6.1.1 Système avec perturbations

La représentation d'état du système soumis à des perturbations est la suivante :

$$\begin{cases} x &= Ax \oplus Bu \oplus Rw = A^*Bu \oplus A^*Rw, \\ y_o &= Cx = CA^*Bu \oplus CA^*Rw, \end{cases} \quad (6.1)$$

avec $A \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{n \times n}$, $B \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{n \times p}$, $C \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{q \times n}$ (ces matrices viennent du modèle sans faute du système), $x \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{n \times 1}$, $u \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{p \times 1}$, $y \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{q \times 1}$, $R \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{n \times l}$ et $w \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{l \times 1}$. Les entrées de perturbation sont reliées aux états internes du système par la matrice de perturbation R telle que :

$$R_{ij} = \begin{cases} \gamma^0 \delta^0 & \text{si l'entrée } x_i \text{ est perturbée par l'entrée } w_j \\ \varepsilon & \text{sinon.} \end{cases}$$

La dimension l représente alors le nombre d'états perturbés. Un état perturbé est une transition interne x_i dont les dates de tirs sont retardées dans le temps (elle ne peuvent jamais être avancées).

Remarque. Nous choisissons dans ces travaux de toujours associer la même numérotation entre les états perturbés et les entrées de perturbation. Cela signifie que la matrice R ne contient des monômes $\gamma^0 \delta^0$ que sur sa diagonale R_{ii} .

Remarque. La dimension l de w peut être identique à la dimension n de x , tous les états sont alors perturbés. Il est également possible que seuls certains états soient perturbés et pas les autres, on a alors $l < n$. La dimension de R est dans ce dernier cas est également réduite.

Exemple 6.1. Reprenons le GET de l'exemple 5.14 du chapitre précédent. La figure 6.3 ci-dessous représente le même GET de l'exemple précédent avec seulement 1 sortie et dont chaque état est soumis à une entrée de perturbation ($l = n$ et chaque x_i est connecté à un w_i).

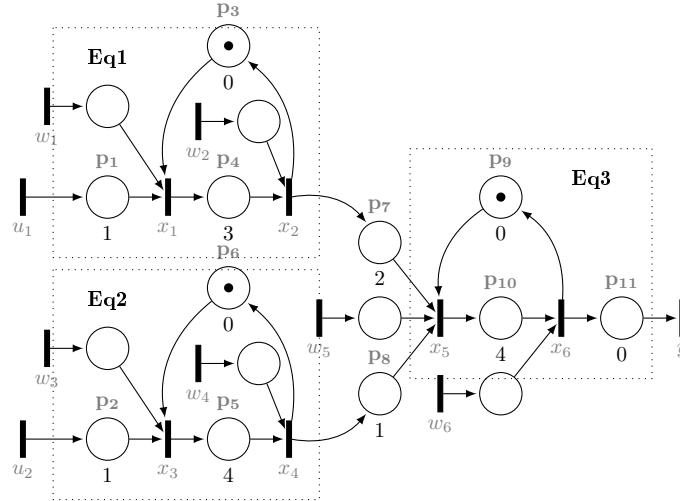


FIGURE 6.3 – GET dont chaque état est soumis à une entrée de perturbation

La représentation d'état de ce GET perturbé est :

$$\left\{ \begin{array}{l} x = \begin{pmatrix} \cdot & \gamma^1 \delta^0 & \cdot & \cdot & \cdot & \cdot \\ \gamma^0 \delta^3 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \gamma^1 \delta^0 & \cdot & \cdot \\ \cdot & \cdot & \gamma^0 \delta^4 & \cdot & \cdot & \cdot \\ \cdot & \gamma^0 \delta^2 & \cdot & \gamma^0 \delta^1 & \cdot & \gamma^1 \delta^0 \\ \cdot & \cdot & \cdot & \cdot & \gamma^0 \delta^4 & \end{pmatrix} x \oplus \begin{pmatrix} \gamma^0 \delta^1 & \cdot & \cdot \\ \cdot & \cdot & \gamma^0 \delta^1 \\ \cdot & \cdot & \cdot \end{pmatrix} u \\ \oplus \begin{pmatrix} \gamma^0 \delta^0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \gamma^0 \delta^0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \gamma^0 \delta^0 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \gamma^0 \delta^0 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \gamma^0 \delta^0 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \gamma^0 \delta^0 \end{pmatrix} w, \\ y = (\cdot \ \cdot \ \cdot \ \cdot \ \cdot \ \gamma^0 \delta^0) x. \end{array} \right.$$

Exemple 6.2. Si dans le GET de la figure 6.3, seuls les états x_1, x_3, x_5 sont perturbés ($l = 3 < n = 6$), la matrice R est alors :

$$R = \begin{pmatrix} \gamma^0 \delta^0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \gamma^0 \delta^0 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \gamma^0 \delta^0 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}.$$

6.1.2 Observateur

Comme indiqué dans la figure 6.2, la représentation d'état de l'observateur est la suivante :

$$\begin{cases} x_e = Ax_e \oplus Bu \oplus L(y_e \oplus y_o), \\ y_e = Cx_e. \end{cases} \quad (6.2)$$

Les états estimés sont notés x_e et la sortie estimée calculée à partir de ces états est notée y_e . La matrice d'observation $L \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{n \times q}$ représente les connexions entre la sortie observée y_o et les transitions estimées x_e de l'observateur.

L'équation d'état de (6.2) peut être ré-écrite par $x_e = (A \oplus LC)^*Bu \oplus (A \oplus LC)^*LCA^*Rw$ comme le montre la démonstration suivante.

Démonstration.

$$\begin{aligned}
 x_e &= Ax_e \oplus Bu \oplus L(y_e \oplus y_o) \\
 &= Ax_e \oplus Bu \oplus Ly_e \oplus Ly_o \\
 &= Ax_e \oplus Bu \oplus LCx_e \oplus LCx \\
 &= Ax_e \oplus Bu \oplus LCx_e \oplus LC(A^*Bu \oplus A^*Rw) && \text{car } y_o = Cx \text{ et } y_e = Cx_e \\
 &= Ax_e \oplus Bu \oplus LCx_e \oplus LCA^*Bu \oplus LCA^*Rw && \text{car } x = A^*Bu \oplus A^*Rw \\
 &= (A \oplus LC)^*(Bu \oplus LCA^*Bu \oplus LCA^*Rw) && \text{car théorème 5.2} \\
 &= (A \oplus LC)^*Bu \oplus (A \oplus LC)^*LCA^*Bu \oplus (A \oplus LC)^*LCA^*Rw \\
 &= A^*(LCA^*)^*Bu \oplus A^*(LCA^*)^*LCA^*Bu \oplus A^*(LCA^*)^*LCA^*Rw && \text{car } (a \oplus b)^* = a^*(ba)^* \\
 &= A^*(LCA^*)^*Bu \oplus A^*(LCA^*)^+Bu \oplus A^*(LCA^*)^*LCA^*Rw && \text{car } (a)^*a = a^+ \\
 &= A^*(LCA^*)^*Bu \oplus A^*(LCA^*)^*LCA^*Rw && \text{car } a^+ \preceq a^* \\
 &= (A \oplus LC)^*Bu \oplus (A \oplus LC)^*LCA^*Rw && \text{car } a^*(ba)^* = (a \oplus b)^*
 \end{aligned}$$

□

La représentation d'état de l'observateur devient alors :

$$\begin{cases} x_e = (A \oplus LC)^*Bu \oplus (A \oplus LC)^*LCA^*Rw, \\ y_e = Cx_e. \end{cases} \quad (6.3)$$

À partir de ces équations, l'objectif de l'observateur est d'estimer l'état x_e de manière aussi proche que possible de l'état réel x sous la condition $x_e \preceq x$. La matrice L qui satisfait cette condition est donnée par $L = L_1 \wedge L_2$ où $L_1 = A^*B \not\propto CA^*B$ et $L_2 = A^*R \not\propto CA^*R$ comme présentée ci-dessous.

Démonstration. Soit,

$$(A \oplus LC)^*Bu \oplus (A \oplus LC)^*LCA^*Rw \preceq x \quad A^*Bu \oplus A^*Rw$$

Les équivalences suivantes sont obtenues par rapport aux entrées u et w :

$$\begin{aligned}
 (A \oplus LC)^*B &\preceq A^*B, \\
 (A \oplus LC)^*LCA^*R &\preceq A^*R
 \end{aligned}$$

Dans un premier temps, la première équivalence permet de calculer la valeur de L_1 .

$$\begin{aligned}
 (A \oplus LC)^*B &\preceq A^*B \\
 \Leftrightarrow (A^*LC)^*A^*B &\preceq A^*B && \text{car } (a \oplus b)^* = (a^*b)^*a^* \\
 \Leftrightarrow (A^*LC)^* &\preceq A^*B \not\propto A^*B && \text{car } xa \preceq b \Leftrightarrow x \preceq (b \not\propto a) \\
 \Leftrightarrow (A^*LC)^* &\preceq (A^*B \not\propto A^*B)^* && \text{car } a \not\propto a = (a \not\propto a)^* \\
 \Leftrightarrow A^*LC &\preceq A^*B \not\propto A^*B && \text{voir la proposition 5.14} \\
 \Leftrightarrow A^*L &\preceq (A^*B \not\propto A^*B) \not\propto C \\
 \Leftrightarrow L &\preceq A^* \not\propto (A^*B \not\propto A^*B \not\propto C) && \text{car } ax \preceq b \Leftrightarrow x \preceq (a \not\propto b) \\
 \Leftrightarrow L &\preceq A^* \not\propto (A^*B \not\propto (CA^*B)) && \text{car } x \not\propto a \not\propto b = x \not\propto (ba) \\
 \Leftrightarrow L &\preceq A^*B \not\propto CA^*B && \text{car } a^* \not\propto (a^*x) = a^*x
 \end{aligned}$$

L'objectif est de calculer L_1 le plus proche possible de L , ainsi :

$$(A \oplus LC)^*B \preceq A^*B \Leftrightarrow L \preceq L_1 = A^*B \not\propto CA^*B$$

La deuxième équivalence permet de calculer la valeur de L_2 .

$$\begin{aligned}
 (A \oplus LC)^*LCA^*R &\preceq A^*R \\
 \Leftrightarrow A^*(LCA^*)LCA^*R &\preceq A^*R \\
 \Leftrightarrow (LCA^*)LCA^*R &\preceq A^* \not\propto A^*R && \text{car } ax \preceq b \Leftrightarrow x \preceq (a \not\propto b) \\
 \Leftrightarrow (LCA^*)LCA^*R &\preceq A^*R && \text{car } a^* \not\propto (a^*b) = (a^*b) \\
 \Leftrightarrow (LCA^*)LCA^*A^*R &\preceq A^*R && \text{car } a^*a^* = a^* \\
 \Leftrightarrow (LCA^*)^+A^*R &\preceq A^*R && \text{car } a^+ = a^*a \\
 \Leftrightarrow (LCA^*)^+ &\preceq A^*R \not\propto A^*R && \text{car } xa \preceq b \Leftrightarrow x \preceq (b \not\propto a) \\
 \Leftrightarrow (LCA^*)^+ &\preceq (A^*R \not\propto A^*R)^* && \text{car } a \not\propto a = (a \not\propto a)^* \\
 \Leftrightarrow LCA^* &\preceq A^*R \not\propto A^*R && \text{voir la proposition 5.14 et 5.15} \\
 \Leftrightarrow L &\preceq A^*R \not\propto A^*R \not\propto CA^* && \text{car } x \oplus a \preceq b \Leftrightarrow (b \not\propto a) \\
 \Leftrightarrow L &\preceq A^*R \not\propto CA^*R && \text{car } x \not\propto a \not\propto b = x \not\propto (ba)
 \end{aligned}$$

L'objectif est de calculer L_2 le plus proche possible de L , ainsi :

$$(A \oplus LC)^* LCA^* R \preceq A^* R \Leftrightarrow L \preceq L_2 = A^* R \not\preceq CA^* R$$

Pour le calcul de L_1 on prend l'inégalité $L \preceq L_1$ et pour calculer L_2 on prend l'inégalité suivante $L \preceq L_2$, alors $L \preceq L_1 \wedge L_2$ cela donne la solution suivante :

$$L = L_1 \wedge L_2.$$

□

D'après la représentation d'état (6.2), chaque ligne de la matrice d'observation L représente un état estimé x_{ei} . Si cette ligne est remplie par un élément de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ différent de ε , l'estimation de l'état x_{ei} prend en compte les perturbations.

Remarque. Dans le reste du manuscrit, pour une matrice donnée M , M_{ij} désigne l'élément de M à la ligne i , et à la colonne j ; $M_{i\bullet}$ est la $i^{ième}$ ligne de M et $M_{\bullet i}$ sa $i^{ième}$ colonne.

Proposition 6.1. *Supposons que $L_{i\bullet} = (\varepsilon \dots \varepsilon)$, alors $x_{ei} = A_{i\bullet}^* B_{\bullet i} u$, c'est-à-dire que l'état estimé x_{ei} ne dépend pas des perturbations.*

Démonstration. Rappelons que ε est neutre pour l'addition et absorbant pour la multiplication. Si $L_{i\bullet} = (\varepsilon \dots \varepsilon)$, alors :

$$\begin{aligned} x_{ei} &= (A_{i\bullet} \oplus L_{i\bullet} C_{\bullet i})^* B_{\bullet i} u \oplus (A_{i\bullet} \oplus L_{i\bullet} C_{\bullet i})^* L_{i\bullet} C_{\bullet i} A_{i\bullet}^* R_{\bullet i} w_i \\ &= A_{i\bullet}^* B_{\bullet i} u \quad \text{car } L_{i\bullet} C_{\bullet i} = (\varepsilon \dots \varepsilon) \\ &\quad \text{et } L_{i\bullet} C_{\bullet i} A_{i\bullet}^* R_{\bullet i} w_i = (\varepsilon \dots \varepsilon) \end{aligned}$$

L'état estimé ne dépend que de A , B et u mais ne dépend pas des perturbations w . □

Du point de vue GET, les états concernés par cette proposition sont les transitions qui ne sont pas connectées à la matrice L .

Exemple 6.3. D'après l'exemple 6.1 avec le GET de la figure 6.3, la matrice L est calculée comme suit :

$$\begin{aligned} L_1 &= A^* B \not\preceq CA^* B = \begin{pmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \gamma^0 \delta^{-4} (\gamma^1 \delta^4)^* \\ \gamma^0 \delta^0 (\gamma^1 \delta^4)^* \end{pmatrix} \\ L_2 &= A^* R \not\preceq CA^* R = \begin{pmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \gamma^1 \delta^0 (\gamma^1 \delta^4)^* \\ \gamma^0 \delta^0 (\gamma^1 \delta^4)^* \end{pmatrix} \\ L &= L_1 \wedge L_2 = \begin{pmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \gamma^1 \delta^0 (\gamma^1 \delta^4)^* \\ \gamma^0 \delta^0 (\gamma^1 \delta^4)^* \end{pmatrix} \end{aligned}$$

Dans cette matrice L , les lignes 1, 2, 3 et 4 sont remplies de ε . D'après la proposition 6.1, cela signifie que les états estimés x_{e1}, x_{e2}, x_{e3} et x_{e4} ne dépendent pas des perturbations. En effet, comme le montre la figure 6.4 qui représente la combinaison de l'observateur (en rouge), composée du GET du modèle sans faute (en jaune) et du GET de la matrice d'observation (en vert), avec le GET du système avec perturbations (en bleu), on peut constater que la sortie y_o n'est connectée qu'aux états x_{e5} et x_{e6} via la matrice L , les seuls états dépendant des perturbations.

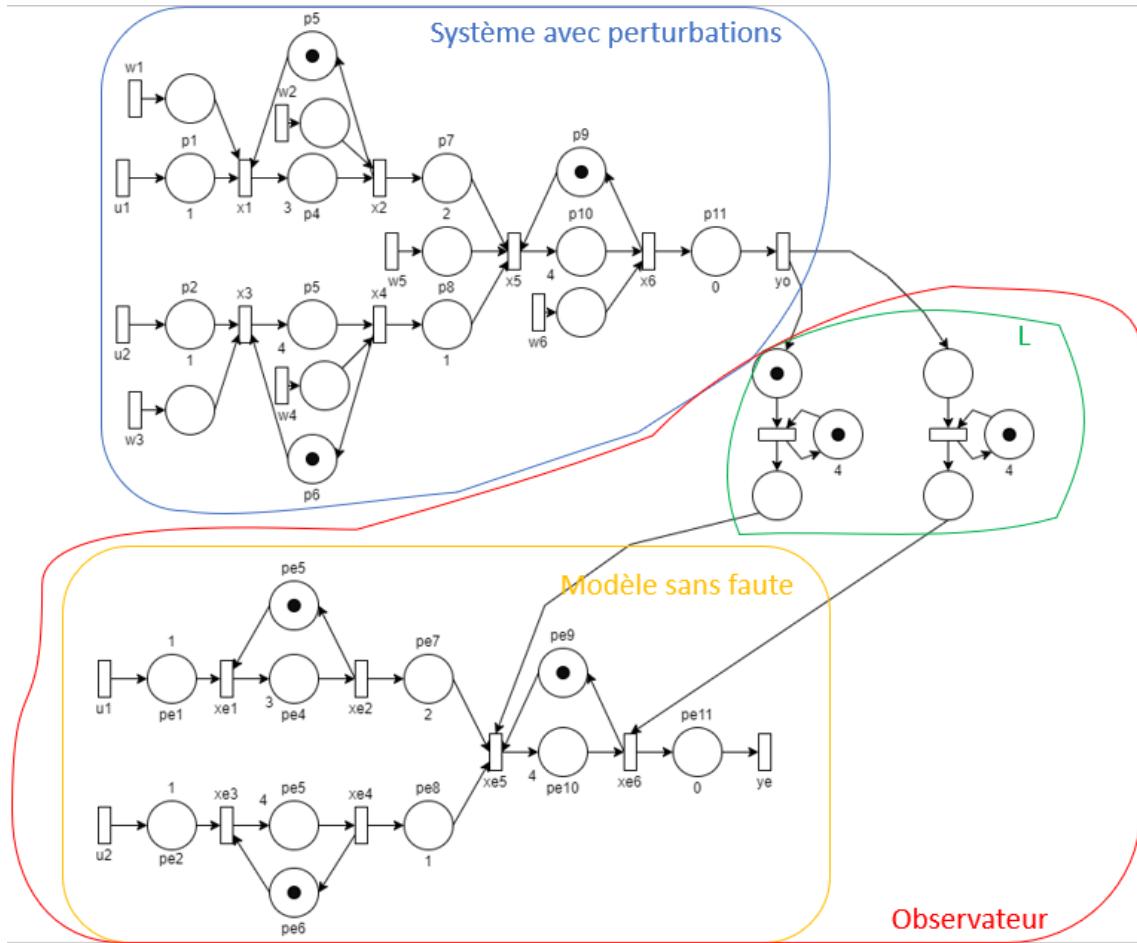


FIGURE 6.4 – Lien entre le système perturbé et l'observateur

Exemple 6.4. Pour le GET de la figure 6.5 avec trois sorties, y_1 , y_2 et y_3 et dont tous les états sont soumis à une entrée de perturbation, la représentation d'état du système avec perturbation est :

$$\left\{ \begin{array}{l}
 x = \begin{pmatrix} \cdot & \gamma^1 \delta^0 & \cdot & \cdot & \cdot & \cdot \\ \gamma^0 \delta^3 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \gamma^1 \delta^0 & \cdot & \cdot \\ \cdot & \gamma^0 \delta^4 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \gamma^0 \delta^2 & \cdot & \gamma^0 \delta^1 & \cdot & \gamma^1 \delta^0 \\ \cdot & \cdot & \cdot & \cdot & \gamma^0 \delta^4 & \cdot \end{pmatrix} x \oplus \begin{pmatrix} \gamma^0 \delta^1 & \cdot & \cdot \\ \cdot & \gamma^0 \delta^1 & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix} u \\
 \oplus \begin{pmatrix} \gamma^0 \delta^0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \gamma^0 \delta^0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \gamma^0 \delta^0 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \gamma^0 \delta^0 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \gamma^0 \delta^0 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \gamma^0 \delta^0 \end{pmatrix} w, \\
 y = \begin{pmatrix} \cdot & \gamma^0 \delta^0 & \cdot & \cdot & \cdot & \gamma^0 \delta^0 \\ \cdot & \gamma^0 \delta^0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \gamma^0 \delta^0 & \cdot & \cdot & \cdot \end{pmatrix} x,
 \end{array} \right.$$

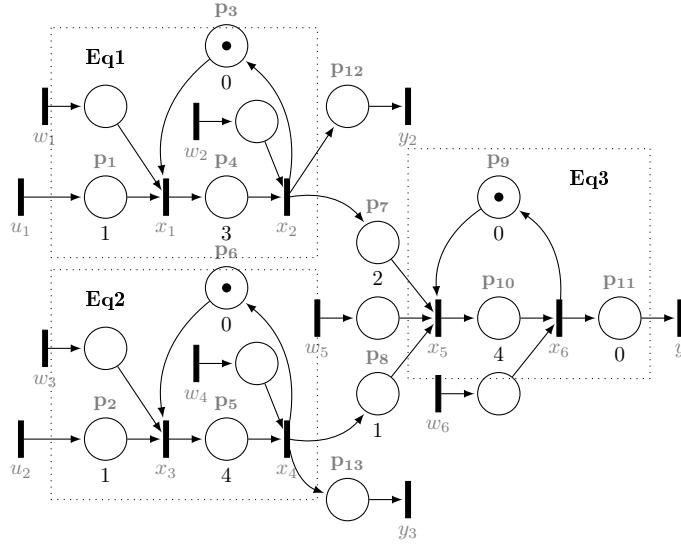


FIGURE 6.5 – GET à 3 sorties dont chaque état est soumis à une entrée de perturbation

Avec cette représentation d'état, on obtient les matrices L_1 , L_2 et L suivantes :

$$\begin{aligned}
 L_1 &= A^* B \not\propto C A^* B = \begin{pmatrix} \cdot & \gamma^0 \delta^{-3} (\gamma^1 \delta^3)^* & \cdot \\ \cdot & \gamma^0 \delta^0 (\gamma^1 \delta^3)^* & \cdot \\ \cdot & \cdot & \gamma^0 \delta^{-4} (\gamma^1 \delta^4)^* \\ \gamma^0 \delta^{-4} (\gamma^1 \delta^4)^* & \gamma^0 \delta^2 (\gamma^1 \delta^4)^* & \gamma^0 \delta^1 (\gamma^1 \delta^4)^* \\ \gamma^0 \delta^0 (\gamma^1 \delta^4)^* & \gamma^0 \delta^6 (\gamma^1 \delta^4)^* & \gamma^0 \delta^5 (\gamma^1 \delta^4)^* \end{pmatrix}, \\
 L_2 &= A^* R \not\propto C A^* R = \begin{pmatrix} \cdot & \gamma^1 \delta^0 (\gamma^1 \delta^3)^* & \cdot \\ \cdot & \gamma^0 \delta^0 (\gamma^1 \delta^3)^* & \cdot \\ \cdot & \cdot & \gamma^1 \delta^0 (\gamma^1 \delta^4)^* \\ \gamma^1 \delta^0 (\gamma^1 \delta^4)^* & \gamma^0 \delta^2 (\gamma^1 \delta^4)^* & \gamma^0 \delta^1 (\gamma^1 \delta^4)^* \\ \gamma^0 \delta^0 (\gamma^1 \delta^4)^* & \gamma^0 \delta^6 (\gamma^1 \delta^4)^* & \gamma^0 \delta^5 (\gamma^1 \delta^4)^* \end{pmatrix}, \\
 L &= L_1 \wedge L_2 = \begin{pmatrix} \cdot & \gamma^1 \delta^0 (\gamma^1 \delta^3)^* & \cdot \\ \cdot & \gamma^0 \delta^0 (\gamma^1 \delta^3)^* & \cdot \\ \cdot & \cdot & \gamma^1 \delta^0 (\gamma^1 \delta^4)^* \\ \gamma^1 \delta^0 (\gamma^1 \delta^4)^* & \gamma^0 \delta^2 (\gamma^1 \delta^4)^* & \gamma^0 \delta^1 (\gamma^1 \delta^4)^* \\ \gamma^0 \delta^0 (\gamma^1 \delta^4)^* & \gamma^0 \delta^6 (\gamma^1 \delta^4)^* & \gamma^0 \delta^5 (\gamma^1 \delta^4)^* \end{pmatrix}.
 \end{aligned}$$

Dans cet exemple, chaque ligne de la matrice L contient au moins un élément différent de ε donc tous les états estimés dépendent des perturbations.

Dans certains cas, les séries contenues dans les matrices L peuvent ne pas être causales (voir définition 5.31). En effet, le calcul de L utilise de nombreuses résiduations qui peuvent avoir pour effet de décaler les séries vers la gauche et vers le bas dans le plan \mathbb{Z}^2 , faisant ainsi apparaître des exposants négatifs pour les γ et les δ . La projection causale définie en 5.35 peut si nécessaire être appliquée pour obtenir une matrice L causale notée L^+ . Nous en aurons notamment besoin lors de la mise en place des états estimés SAP (sensibles-aux-perturbations) utilisés pour la détection des dérives temporelles.

Exemple 6.5. Dans le cas de la matrice L suivante :

$$L = \begin{pmatrix} \gamma^0 \delta^{-2} (\gamma^1 \delta^3)^* \\ \gamma^0 \delta^1 (\gamma^1 \delta^3)^* \end{pmatrix},$$

la série de la première ligne n'est pas causale car le monôme $\gamma^0 \delta^{-2}$ a un exposant qui n'appartient

pas à l'ensemble \mathbb{N} . Si la projection causale est appliquée sur cette série, on obtient L^+ :

$$L^+ = Pr^+(L) = \begin{pmatrix} \gamma^1 \delta^1 (\gamma^1 \delta^3)^* \\ \gamma^0 \delta^1 (\gamma^1 \delta^3)^* \end{pmatrix}.$$

6.2 Détection de dérives temporelles

6.2.1 Caractérisation des dérives temporelles par des perturbations

L'utilisation des observateurs pour détecter et localiser les dérives temporelles nécessite de les caractériser par des entrées de perturbation. La définition 4.3 d'une dérive temporelle dans un GET indique que si une telle dérive survient sur une place, la durée associée à cette place, et donc le temps de séjour des jetons dans la place, est modifiée. Dans ce chapitre, en caractérisant les dérives par des entrées de perturbations qui ne peuvent que ralentir le système, les modifications considérées ne sont que des allongements des temps de séjour, donc des dérives de type retard.

Ainsi, la dérive temporelle est considérée comme un phénomène permanent et est définie par un délai inconnu $\theta > 0$ de type retard sur une place p d'un GET. Chaque place p a une transition en amont x_{i-1} , une transition en aval x_i , un nombre o de jeton(s) et une durée t comme on peut le voir sur la figure 6.6. La transition en amont de la place p est décrite par une série de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$

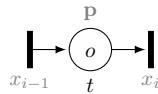


FIGURE 6.6 – Représentation d'une place dans un GET

sous forme de représentation canonique (voir définition 5.30) :

$$x_{i-1} = \bigoplus_{n=0}^K \gamma^{s_n} \delta^{h_n},$$

où s_n est le numéro de l'événement, h_n est la date de l'événement et K le nombre d'événements de la transition. La transition en aval est décrite par :

$$x_i = \bigoplus_{n=0}^K \gamma^{s_n+o} \delta^{h_n+t}.$$

Dans le cas où une dérive temporelle a lieu sur la place p , si la dérive temporelle est $\theta > 0$, alors la transition en aval devient :

$$x_i = \bigoplus_{n=0}^K \gamma^{s_n+o} \delta^{h_n+t+\theta}.$$

Les dates de tir de la transition x_i sont ralenties par la dérive temporelle θ par rapport à la transition x_i du comportement sans faute. La même dérive temporelle sur une place p peut être caractérisée par une entrée de perturbation w_i comme illustré dans la figure 6.7. L'expression de cette perturbation est :

$$w_i = \bigoplus_{n=0}^K \gamma^{s_n+o} \delta^{h_n+t+\theta},$$

avec $s_n + o$ et $h_n + t$ définis par la série x_i (s_n et h_n viennent quant à eux de la série x_{i-1}). La synchronisation sur la transition x_i entre w_i et x_{i-1} impose un décalage temporel de valeur θ sur les dates de tir de x_i . Cette entrée w_i ralentit donc la transition x_i par la dérive θ (puisque θ est strictement positif). En d'autres termes, avoir une dérive temporelle inconnue $\theta > 0$ sur une place p d'un GET est équivalent à dire qu'il existe une perturbation temporelle inconnue w_i reliée à la transition en aval x_i de la place. Dans l'expression de w_i , seules les dates de tir de transition sont donc inconnues (les exposants de δ) tandis que le nombre de tirs de transition de w_i (les exposants de γ) est identique au nombre de tirs de x_i , qu'il y ait une dérive temporelle sur la place p ou non.

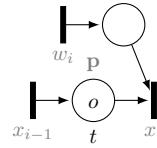


FIGURE 6.7 – Représentation d'une place avec perturbation dans un GET

Exemple 6.6. Reprenons le GET de la figure 6.7 avec $x_{i-1} = x_1, x_i = x_2, o = 0$ (pas de jeton dans la place p) et $t = 3$. L'entrée de perturbation w_2 est connectée à la transition x_2 . Si la trajectoire de la transition x_1 est

$$x_1 = \gamma^0 \delta^0 \oplus \gamma^1 \delta^2 \oplus \gamma^2 \delta^3 \oplus \gamma^4 \delta^{+\infty},$$

la trajectoire de la transition x_2 est alors

$$x_2 = \gamma^0 \delta^{0+3} \oplus \gamma^1 \delta^{2+3} \oplus \gamma^2 \delta^{3+3} \oplus \gamma^4 \delta^{+\infty}.$$

Pour caractériser une dérive temporelle $\theta = 1$ sur la place p , l'entrée de perturbation doit être :

$$\begin{aligned} w_2 &= \gamma^0 \delta^{0+3+1} \oplus \gamma^1 \delta^{2+3+1} \oplus \gamma^2 \delta^{3+3+1} \oplus \gamma^4 \delta^{+\infty} \\ &= \gamma^0 \delta^4 \oplus \gamma^1 \delta^6 \oplus \gamma^2 \delta^7 \oplus \gamma^4 \delta^{+\infty}. \end{aligned}$$

Selon cette méthode, les dérives temporelles ne peuvent se produire que sur les places dont les transitions en aval sont perturbées. Les dimensions du vecteur w et de la matrice R sont modifiées en fonction de ce choix (voir exemples 6.2 et 6.1).

6.2.2 Indicateur

Notre méthode de détection de dérives temporelles, présentée dans la figure 6.1, cherche à comparer les états estimés par l'observateur x_e avec les états du modèle sans faute x_s , connaissant l'entrée u appliquée au système et la mesure de la sortie y_o . Un indicateur booléen $I_{x_i}(u, y_o)$ par état est proposé pour réaliser cette comparaison et utilise les définitions de la section 5.5 sur les décalages temporels entre deux séries de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$, notamment l'intervalle $\Delta(x_e, x_s)$ de la définition 5.37. Les hypothèses de cette méthode sont qu'une ou plusieurs dérives temporelles permanentes peuvent avoir lieu dans le système et que tous les états du système sont connectés à des entrées de perturbations, donc les dérives peuvent se produire n'importe où dans le système. Rappelons qu'un état connecté à une entrée de perturbation n'implique pas forcément que cet état est perturbé. C'est-à-dire qu'un état est dit perturbé si la place située en amont de la transition reliée à une entrée de perturbation est soumise à une dérive temporelle.

Définition 6.1. Soient x_e l'état estimé par l'observateur et x_s l'état du modèle sans faute, l'indicateur de dérive temporelle associé à l'état x_i est noté $I_{x_i}(u, y_o)$ et est défini par la fonction booléenne suivante :

$$I_{x_i}(u, y_o) = \begin{cases} \text{faux} & \text{si } \Delta(x_{ei}, x_{si}) = [0; 0], \\ \text{vrai} & \text{sinon,} \end{cases}$$

avec

$$\begin{aligned} x_s &= [x_{s1} \dots x_{sn}]^T = A^* B u \\ x_e &= [x_{e1} \dots x_{en}]^T = A x_e \oplus B u \oplus L C x_e \oplus L y_o \\ \Delta(x_{ei}, x_{si}) &= [\mathcal{D}_{x_{ei} \neq x_{si}}(0); -\mathcal{D}_{x_{si} \neq x_{ei}}(0)]. \end{aligned}$$

Proposition 6.2. L'indicateur $I_{x_i}(u, y_o)$ retourne vrai, donc $\Delta(x_{ei}, x_{si}) \neq [0, 0]$, seulement si une dérive temporelle impacte la trajectoire de l'état x_i par rapport à son comportement sans faute.

Démonstration. Si le système ne présente aucune dérive temporelle, par définition de l'observateur, les états estimés x_e sont les mêmes que les états du modèle sans faute x_s (aucune transition interne n'est perturbée). En effet, pour un état donné, si $x_{si} = x_{ei}$, alors $x_{si} \neq x_{ei} = x_{ei} \neq x_{si} = x_{ei} \neq x_{ei}$. Or $x_{ei} \neq x_{ei} = (x_{ei} \neq x_{ei})^* = e \oplus \dots = \gamma^0 \delta^0 \oplus \dots$ selon le théorème 5.7 et la définition 5.2 de l'étoile de Kleene. Donc si $x_{si} = x_{ei}$, l'intervalle de décalage temporel de la définition 5.37 est égal à $\Delta(x_{ei}, x_{si}) = [\mathcal{D}_{x_{ei} \neq x_{si}}(0); -\mathcal{D}_{x_{si} \neq x_{ei}}(0)] = [0, 0]$. \square

Exemple 6.7. Reprenons l'exemple 6.4 avec le GET de la figure 6.5 (3 sorties) pour lequel l'entrée suivante est appliquée :

$$u_1 = u_2 = \gamma^0\delta^1 \oplus \gamma^1\delta^2 \oplus \gamma^2\delta^3 \oplus \gamma^3\delta^4 \oplus \gamma^4\delta^5 \oplus \gamma^5\delta^6 \oplus \gamma^6\delta^7 \oplus \gamma^7\delta^{+\infty}.$$

Une dérive temporelle $\theta = 1$ est appliquée sur la place p_4 , dérive réalisée par l'entrée de perturbation w_2 . L'état du modèle sans faute est :

$$x_s = \begin{pmatrix} \gamma^0\delta^2 \oplus \gamma^1\delta^5 \oplus \gamma^2\delta^8 \oplus \gamma^3\delta^{11} \oplus \gamma^4\delta^{14} \oplus \gamma^5\delta^{17} \oplus \gamma^6\delta^{20} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^5 \oplus \gamma^1\delta^8 \oplus \gamma^2\delta^{11} \oplus \gamma^3\delta^{14} \oplus \gamma^4\delta^{17} \oplus \gamma^5\delta^{20} \oplus \gamma^6\delta^{23} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^2 \oplus \gamma^1\delta^6 \oplus \gamma^2\delta^{10} \oplus \gamma^3\delta^{14} \oplus \gamma^4\delta^{18} \oplus \gamma^5\delta^{22} \oplus \gamma^6\delta^{26} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^6 \oplus \gamma^1\delta^{10} \oplus \gamma^2\delta^{14} \oplus \gamma^3\delta^{18} \oplus \gamma^4\delta^{22} \oplus \gamma^5\delta^{26} \oplus \gamma^6\delta^{30} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^7 \oplus \gamma^1\delta^{11} \oplus \gamma^2\delta^{15} \oplus \gamma^3\delta^{19} \oplus \gamma^4\delta^{23} \oplus \gamma^5\delta^{27} \oplus \gamma^6\delta^{31} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^{11} \oplus \gamma^1\delta^{15} \oplus \gamma^2\delta^{19} \oplus \gamma^3\delta^{23} \oplus \gamma^4\delta^{27} \oplus \gamma^5\delta^{31} \oplus \gamma^6\delta^{35} \oplus \gamma^7\delta^{+\infty} \end{pmatrix}.$$

L'état estimé par l'observateur est :

$$x_e = \begin{pmatrix} \gamma^0\delta^2 \oplus \gamma^1\delta^6 \oplus \gamma^2\delta^{10} \oplus \gamma^3\delta^{14} \oplus \gamma^4\delta^{18} \oplus \gamma^5\delta^{22} \oplus \gamma^6\delta^{26} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^6 \oplus \gamma^1\delta^{10} \oplus \gamma^2\delta^{14} \oplus \gamma^3\delta^{18} \oplus \gamma^4\delta^{22} \oplus \gamma^5\delta^{26} \oplus \gamma^6\delta^{30} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^2 \oplus \gamma^1\delta^6 \oplus \gamma^2\delta^{10} \oplus \gamma^3\delta^{14} \oplus \gamma^4\delta^{18} \oplus \gamma^5\delta^{22} \oplus \gamma^6\delta^{26} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^6 \oplus \gamma^1\delta^{10} \oplus \gamma^2\delta^{14} \oplus \gamma^3\delta^{18} \oplus \gamma^4\delta^{22} \oplus \gamma^5\delta^{26} \oplus \gamma^6\delta^{30} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^8 \oplus \gamma^1\delta^{12} \oplus \gamma^2\delta^{16} \oplus \gamma^3\delta^{20} \oplus \gamma^4\delta^{24} \oplus \gamma^5\delta^{28} \oplus \gamma^6\delta^{32} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^{12} \oplus \gamma^1\delta^{16} \oplus \gamma^2\delta^{20} \oplus \gamma^3\delta^{24} \oplus \gamma^4\delta^{28} \oplus \gamma^5\delta^{32} \oplus \gamma^6\delta^{36} \oplus \gamma^7\delta^{+\infty} \end{pmatrix}.$$

Les indicateurs obtenus pour cet exemple sont alors :

$$\begin{aligned} I_{x_1}(u, y_o) &= \text{vrai} & \text{car } \Delta(x_{e1}, x_{s1}) &= [0, 6], \\ I_{x_2}(u, y_o) &= \text{vrai} & \text{car } \Delta(x_{e2}, x_{s2}) &= [1, 7], \\ I_{x_3}(u, y_o) &= \text{faux} & \text{car } \Delta(x_{e3}, x_{s3}) &= [0, 0], \\ I_{x_4}(u, y_o) &= \text{faux} & \text{car } \Delta(x_{e4}, x_{s4}) &= [0, 0], \\ I_{x_5}(u, y_o) &= \text{vrai} & \text{car } \Delta(x_{e5}, x_{s5}) &= [1, 1], \\ I_{x_6}(u, y_o) &= \text{vrai} & \text{car } \Delta(x_{e6}, x_{s6}) &= [1, 1]. \end{aligned}$$

Puisque certains indicateurs sont égaux à vrai (les intervalles sont différents de $[0, 0]$), la dérive temporelle est détectée.

Exemple 6.8. Reprenons à présent le GET de la figure 6.3 (GET identique au précédent avec 1 seule sortie y) pour lequel l'entrée de l'exemple précédent est appliquée et rappelé ici :

$$u_1 = u_2 = \gamma^0\delta^1 \oplus \gamma^1\delta^2 \oplus \gamma^2\delta^3 \oplus \gamma^3\delta^4 \oplus \gamma^4\delta^5 \oplus \gamma^5\delta^6 \oplus \gamma^6\delta^7 \oplus \gamma^7\delta^{+\infty}.$$

La dérive temporelle $\theta = 1$ est à nouveau appliquée sur la place p_4 . L'état du modèle sans faute est :

$$x_s = \begin{pmatrix} \gamma^0\delta^2 \oplus \gamma^1\delta^5 \oplus \gamma^2\delta^8 \oplus \gamma^3\delta^{11} \oplus \gamma^4\delta^{14} \oplus \gamma^5\delta^{17} \oplus \gamma^6\delta^{20} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^5 \oplus \gamma^1\delta^8 \oplus \gamma^2\delta^{11} \oplus \gamma^3\delta^{14} \oplus \gamma^4\delta^{17} \oplus \gamma^5\delta^{20} \oplus \gamma^6\delta^{23} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^2 \oplus \gamma^1\delta^6 \oplus \gamma^2\delta^{10} \oplus \gamma^3\delta^{14} \oplus \gamma^4\delta^{18} \oplus \gamma^5\delta^{22} \oplus \gamma^6\delta^{26} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^6 \oplus \gamma^1\delta^{10} \oplus \gamma^2\delta^{14} \oplus \gamma^3\delta^{18} \oplus \gamma^4\delta^{22} \oplus \gamma^5\delta^{26} \oplus \gamma^6\delta^{30} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^7 \oplus \gamma^1\delta^{11} \oplus \gamma^2\delta^{15} \oplus \gamma^3\delta^{19} \oplus \gamma^4\delta^{23} \oplus \gamma^5\delta^{27} \oplus \gamma^6\delta^{31} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^{11} \oplus \gamma^1\delta^{15} \oplus \gamma^2\delta^{19} \oplus \gamma^3\delta^{23} \oplus \gamma^4\delta^{27} \oplus \gamma^5\delta^{31} \oplus \gamma^6\delta^{35} \oplus \gamma^7\delta^{+\infty} \end{pmatrix}.$$

L'état estimé par l'observateur est :

$$x_e = \begin{pmatrix} \gamma^0\delta^2 \oplus \gamma^1\delta^5 \oplus \gamma^2\delta^8 \oplus \gamma^3\delta^{11} \oplus \gamma^4\delta^{14} \oplus \gamma^5\delta^{17} \oplus \gamma^6\delta^{20} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^5 \oplus \gamma^1\delta^8 \oplus \gamma^2\delta^{11} \oplus \gamma^3\delta^{14} \oplus \gamma^4\delta^{17} \oplus \gamma^5\delta^{20} \oplus \gamma^6\delta^{23} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^2 \oplus \gamma^1\delta^6 \oplus \gamma^2\delta^{10} \oplus \gamma^3\delta^{14} \oplus \gamma^4\delta^{18} \oplus \gamma^5\delta^{22} \oplus \gamma^6\delta^{26} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^6 \oplus \gamma^1\delta^{10} \oplus \gamma^2\delta^{14} \oplus \gamma^3\delta^{18} \oplus \gamma^4\delta^{22} \oplus \gamma^5\delta^{26} \oplus \gamma^6\delta^{30} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^7 \oplus \gamma^1\delta^{12} \oplus \gamma^2\delta^{16} \oplus \gamma^3\delta^{20} \oplus \gamma^4\delta^{24} \oplus \gamma^5\delta^{28} \oplus \gamma^6\delta^{32} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^{12} \oplus \gamma^1\delta^{16} \oplus \gamma^2\delta^{20} \oplus \gamma^3\delta^{24} \oplus \gamma^4\delta^{28} \oplus \gamma^5\delta^{32} \oplus \gamma^6\delta^{36} \oplus \gamma^7\delta^{+\infty} \end{pmatrix}.$$

Les indicateurs obtenus pour cet exemple sont alors :

$$\begin{aligned} I_{x_1}(u, y_o) &= \text{faux} & \text{car } \Delta(x_{e1}, x_{s1}) &= [0, 0], \\ I_{x_2}(u, y_o) &= \text{faux} & \text{car } \Delta(x_{e2}, x_{s2}) &= [0, 0], \\ I_{x_3}(u, y_o) &= \text{faux} & \text{car } \Delta(x_{e3}, x_{s3}) &= [0, 0], \\ I_{x_4}(u, y_o) &= \text{faux} & \text{car } \Delta(x_{e4}, x_{s4}) &= [0, 0], \\ I_{x_5}(u, y_o) &= \text{vrai} & \text{car } \Delta(x_{e5}, x_{s5}) &= [0, 1], \\ I_{x_6}(u, y_o) &= \text{vrai} & \text{car } \Delta(x_{e6}, x_{s6}) &= [1, 1]. \end{aligned}$$

Puisque certains indicateurs sont égaux à vrai (les intervalles sont différents de $[0, 0]$), la dérive temporelle est détectée. Pour ce GET, la matrice d'observation L , donnée dans l'exemple 6.3, contient des lignes remplies de ε , donc les états estimés x_{ei} avec $i \in \{1, 2, 3, 4\}$ ne dépendent pas des perturbations. Nous obtenons donc des indicateurs qui retournent faux (car les intervalles sont égaux à $[0, 0]$) alors que dans l'exemple précédent, pour la même dérive temporelle, les indicateurs sur les états x_1 et x_2 retournaient vrai. D'autre part, dans l'exemple précédent, l'intervalle de décalage temporel associé à l'état x_5 était égal à $[1, 1]$ tandis que dans cet exemple, l'intervalle est de $[0, 1]$. Cette différence s'explique par le fait que le premier monôme de x_{e5} est identique au premier monôme de x_{s5} ce qui permet d'obtenir la borne inférieure de l'intervalle égale à 0. Ainsi, l'effet de la dérive temporelle n'apparaît pas immédiatement dans l'estimation de l'état x_5 (il n'apparaît qu'à partir du second monôme).

Cet exemple met en évidence des différences entre les estimations et la réalité. La conséquence de ces phénomènes est que l'indicateur peut générer des faux négatifs (c'est-à-dire qu'une dérive temporelle peut avoir lieu alors que l'indicateur n'est pas levé) et ne pas être suffisamment réactif par rapport aux dérives. La section suivante propose un raffinement de l'estimation des états pour pallier ces problèmes.

6.2.3 Raffinement de l'état estimé

L'indicateur donné par la définition 6.1 peut générer des faux négatifs lors la détection des dérives temporelles. Comme l'effet de la dérive temporelle peut ne pas être immédiat, les premiers monômes de l'estimation x_e peuvent ne pas être affectés par la perturbation. Cela a pour conséquence que l'indicateur montre un comportement sans dérive temporelle alors qu'elle a lieu ou que la dérive n'est pas prise en compte dès son apparition. Pour filtrer ces informations, un raffinement des états x_e est proposé pour ne garder que les monômes de la série affectés par la perturbation pour obtenir un état estimé *sensible-aux-perturbations* dit SAP.

Remarque. Pour rappel, dans ce qui suit, pour une matrice donnée M , M_{ij} désigne l'élément de M à la ligne i , et à la colonne j ; $M_{i\bullet}$ est la $i^{\text{ème}}$ ligne de M et $M_{\bullet i}$ sa $i^{\text{ème}}$ colonne.

La proposition ci-dessous donne le numéro de l'événement de l'état estimé à partir duquel l'effet de la perturbation est pris en compte. Deux notions sont nécessaires à cette proposition :

- $(n_0)_{ij}$: Soit un élément de la matrice L sous la forme canonique $L_{ij} = \bigoplus_{k=0}^K \gamma^{n_k} \delta^{t_k}$, $(n_0)_{ij}$ désigne le plus petit exposant de γ dans la série L_{ij} . Il est possible que $(n_0)_{ij}$ soit différent de 0 si des jetons se trouvent déjà présents dans la place représentée par $\gamma^{n_0} \delta^{t_0}$ dans L_{ij} .
- N_i : Soit $N_i = \min_{1 \leq j \leq q} \{(n_0)_{ij}\}$ avec q le nombre de sorties du système. N_i représente le plus petit $(n_0)_{ij}$ parmi les colonnes j de la ligne i de la matrice L .

Proposition 6.3. Soit $x_{ei} = \bigoplus_{\kappa=0}^K \gamma^{\nu_\kappa} \delta^{\tau_\kappa}$ la trajectoire estimée de x_i sous sa forme canonique, la première occurrence de l'événement x_{ei} qui prend en compte les perturbations temporelles de w_i est l'occurrence d'exposant ν_d telle que :

$$\nu_d = N_i + \nu_0^b + \nu_0^{w_i}$$

où ν_0^b et $\nu_0^{w_i}$ sont les plus petits exposants de γ des séries $b = C_{\bullet i} A_{i\bullet}^* R_{\bullet i}$ et w_i .

Démonstration. Soit l'état estimé :

$$\begin{aligned} x_{ei} &= (A_{i\bullet} \oplus L_{i\bullet} C_{\bullet i})^* B_{\bullet i} u \oplus (A_{i\bullet} \oplus L_{i\bullet} C_{\bullet i})^* L_{i\bullet} C_{\bullet i} A_{i\bullet}^* R_{\bullet i} w_i \\ &= (A_{i\bullet} \oplus L_{i\bullet} C_{\bullet i})^* (B_{\bullet i} u \oplus L_{i\bullet} C_{\bullet i} A_{i\bullet}^* R_{\bullet i} w_i) \\ &= (A_{i\bullet} \oplus L_{i\bullet} C_{\bullet i})^* (au \oplus L_{i\bullet} bw_i) \end{aligned}$$

où $a = B_{\bullet i}$ et $b = C_{\bullet i} A_{i\bullet}^* R_{\bullet i}$ sont des séries de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$. Dans la dernière équation, la seule partie dépendante de la perturbation w_i est la série $L_{i\bullet} b$. On cherche donc dans un premier temps, le plus petit exposant de γ dans la forme canonique de la série $L_{i\bullet} bw_i$, exposant noté ν_d . On note $L_{i\bullet} = (L_{i1} \ \dots \ L_{iq}) = \left(\bigoplus_{k=0}^{K_1} \gamma^{n_k^1} \delta^{t_k^1} \ \dots \ \bigoplus_{k=0}^{K_q} \gamma^{n_k^q} \delta^{t_k^q} \right)$ (avec q le nombre de sorties du système) et $b = \bigoplus_{h=0}^{K_b} \gamma^{\nu_h^b} \delta^{\tau_h^b}$ les représentations canoniques de ces séries. Commençons par le

détail du calcul de $L_{i_\bullet} \otimes b$:

$$\begin{aligned}
 L_{i_\bullet} \otimes b &= \left(\bigoplus_{k=0}^{K_1} \gamma^{n_k^1} \delta^{t_k^1} \cdots \bigoplus_{k=0}^{K_q} \gamma^{n_k^q} \delta^{t_k^q} \right) \otimes \bigoplus_{h=0}^{K_b} \gamma^{\nu_h^b} \delta^{\tau_h^b} \\
 &= \bigoplus_{k=0}^{K_1} \gamma^{n_k^1} \delta^{t_k^1} \otimes \bigoplus_{h=0}^{K_b} \gamma^{\nu_h^b} \delta^{\tau_h^b} \oplus \cdots \oplus \bigoplus_{k=0}^{K_q} \gamma^{n_k^q} \delta^{t_k^q} \otimes \bigoplus_{h=0}^{K_b} \gamma^{\nu_h^b} \delta^{\tau_h^b} \\
 &= \bigoplus_{k=0}^{K_1} \bigoplus_{h=0}^{K_b} \gamma^{n_k^1} \delta^{t_k^1} \otimes \gamma^{\nu_h^b} \delta^{\tau_h^b} \oplus \cdots \oplus \bigoplus_{k=0}^{K_q} \bigoplus_{h=0}^{K_b} \gamma^{n_k^q} \delta^{t_k^q} \otimes \gamma^{\nu_h^b} \delta^{\tau_h^b} \\
 &= \bigoplus_{k=0}^{K_1} \bigoplus_{h=0}^{K_b} \gamma^{n_k^1 + \nu_h^b} \delta^{t_k^1 + \tau_h^b} \oplus \cdots \oplus \bigoplus_{k=0}^{K_q} \bigoplus_{h=0}^{K_b} \gamma^{n_k^q + \nu_h^b} \delta^{t_k^q + \tau_h^b} \\
 &= \bigoplus_{j=1}^q \bigoplus_{k=0}^{K_j} \bigoplus_{h=0}^{K_b} \gamma^{n_k^j + \nu_h^b} \delta^{t_k^j + \tau_h^b},
 \end{aligned}$$

auquel nous ajoutons la perturbation w_i (donnée elle aussi sous sa forme canonique) :

$$\begin{aligned}
 L_{i_\bullet} \otimes b \otimes w_i &= \bigoplus_{j=1}^q \bigoplus_{k=0}^{K_j} \bigoplus_{h=0}^{K_b} \gamma^{n_k^j + \nu_h^b} \delta^{t_k^j + \tau_h^b} \otimes \bigoplus_{f=0}^{K_{w_i}} \gamma^{\nu_f^{w_i}} \delta^{\tau_f^{w_i}} \\
 &= \bigoplus_{j=1}^q \bigoplus_{k=0}^{K_j} \bigoplus_{h=0}^{K_b} \bigoplus_{f=0}^{K_{w_i}} \gamma^{n_k^j + \nu_h^b + \nu_f^{w_i}} \delta^{t_k^j + \tau_h^b + \tau_f^{w_i}}.
 \end{aligned}$$

Ainsi, tout exposant de γ dans la forme canonique de $L_{i_\bullet} \otimes b \otimes w_i$ est de la forme $n_k^j + \nu_h^b + \nu_f^{w_i}$ où n_k^j est issu de L_{i_\bullet} , ν_h^b est issu de b et $\nu_f^{w_i}$ est issu de w_i . L'exposant ν_d que l'on cherche à déterminer est donc le plus petit d'entre eux. Selon les règles de calcul de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$, ce plus petit exposant ν_d est :

$$\nu_d = \min_{1 \leq j \leq q} \{n_0^j\} + \nu_0^b + \nu_0^{w_i} = \min_{1 \leq j \leq q} \{(n_0)_{ij}\} + \nu_0^b + \nu_0^{w_i} = N_i + \nu_0^b + \nu_0^{w_i}.$$

□

Dans notre méthode de détection de dérives temporelles, nous supposons que les perturbations w_i ont une forme particulière car elles caractérisent les dérives temporelles que nous cherchons à déceler. Comme expliqué dans la section 6.2.1, ces perturbations sont des entrées exprimées en fonction des tirs des transitions auxquelles elles sont connectées (w_i dépend de x_i). Les numéros de tir de w_i , donc les exposants des γ de la série, sont connus car égaux aux numéros de tir de x_i . En revanche, si une dérive temporelle est effectivement présente sur une place précédant x_i , les dates de tirs de w_i , et donc les exposants des δ de w_i , ne sont pas connus. Ainsi, bien que w_i ne soit pas entièrement connue, son composant $\nu_0^{w_i}$ l'est, et donc l'occurrence ν_d est bien connue.

D'autre part, lorsque la matrice L n'est pas causale (voir exemple 6.5), le calcul de N_i n'est pas possible. En effet, il peut arriver que certains exposants de γ ou δ des séries de L soient négatifs. De tels exposants n'ont pas de sens physique dans les GET et nous ne savons alors pas quand la perturbation est prise en compte et donc, à partir de quel exposant de γ on peut se référer pour le calcul de l'état estimé SAP. Il faut donc rendre L causale¹ avant de calculer ν_d .

Exemple 6.9. Reprenons l'exemple 6.8 (1 seule sortie) avec pour entrée :

$$u_1 = u_2 = \gamma^0 \delta^1 \oplus \gamma^1 \delta^2 \oplus \gamma^2 \delta^3 \oplus \gamma^3 \delta^4 \oplus \gamma^4 \delta^5 \oplus \gamma^5 \delta^6 \oplus \gamma^6 \delta^7 \oplus \gamma^7 \delta^{+\infty},$$

et pour matrice L :

$$\begin{pmatrix} \vdots \\ \vdots \\ \vdots \\ \gamma^1 \delta^0 (\gamma^1 \delta^4)^* \\ \gamma^0 \delta^0 (\gamma^1 \delta^4)^* \end{pmatrix}$$

Le calcul de $b = CA^*R$ est :

$$CA^*R = (\gamma^0 \delta^9 (\gamma^1 \delta^4)^* \quad \gamma^0 \delta^6 (\gamma^1 \delta^4)^* \quad \gamma^0 \delta^9 (\gamma^1 \delta^4)^* \quad \gamma^0 \delta^5 (\gamma^1 \delta^4)^* \quad \gamma^0 \delta^4 (\gamma^1 \delta^4)^* \quad \gamma^0 \delta^0 (\gamma^1 \delta^4)^*).$$

Pour l'état x_1 , puisque $L_{1_\bullet} = \varepsilon = \gamma^{+\infty} \delta^{-\infty}$, $N_1 = +\infty$. D'après la série $b = CA^*R$, le plus petit exposant de γ de la colonne 1 donne $\nu_0^b = 0$. Enfin, grâce à l'entrée u , on obtient les exposants de γ de l'état x_1 et donc les exposants de γ de w_1 , soit $\nu_0^{w_1} = 0$. Donc, $\nu_d = +\infty + 0 + 0 = +\infty$. On réitère le procédé pour les autres états afin d'obtenir le tableau suivant.

1. En revanche, le calcul de x_{ei} utilise la matrice L d'origine et non la matrice causale L^+ .

Transition	N_i	ν_0^b	$\nu_0^{w_i}$	ν_d
x_1	$+\infty$	0	0	$+\infty$
x_2	$+\infty$	0	0	$+\infty$
x_3	$+\infty$	0	0	$+\infty$
x_4	$+\infty$	0	0	$+\infty$
x_5	1	0	0	1
x_6	0	0	0	0

TABLE 6.1 – Tableau récapitulatif des ν_d par états estimés

Exemple 6.10. Reprenons maintenant l'exemple 6.4 (3 sorties) avec pour entrée :

$$u_1 = u_2 = \gamma^0 \delta^1 \oplus \gamma^1 \delta^2 \oplus \gamma^2 \delta^3 \oplus \gamma^3 \delta^4 \oplus \gamma^4 \delta^5 \oplus \gamma^5 \delta^6 \oplus \gamma^6 \delta^7 \oplus \gamma^7 \delta^{+\infty},$$

et pour matrice L :

$$L = \begin{pmatrix} \cdot & \gamma^1 \delta^0 (\gamma^1 \delta^3)^* & \cdot \\ \cdot & \gamma^0 \delta^0 (\gamma^1 \delta^3)^* & \cdot \\ \cdot & \cdot & \gamma^1 \delta^0 (\gamma^1 \delta^4)^* \\ \cdot & \cdot & \gamma^0 \delta^0 (\gamma^1 \delta^4)^* \\ \gamma^1 \delta^0 (\gamma^1 \delta^4)^* & \gamma^0 \delta^2 (\gamma^1 \delta^4)^* & \gamma^0 \delta^1 (\gamma^1 \delta^4)^* \\ \gamma^0 \delta^0 (\gamma^1 \delta^4)^* & \gamma^0 \delta^6 (\gamma^1 \delta^4)^* & \gamma^0 \delta^5 (\gamma^1 \delta^4)^* \end{pmatrix}$$

Le calcul de $b = CA^*R$ est :

$$CA^*R = \begin{pmatrix} \gamma^0 \delta^9 (\gamma^1 \delta^4)^* & \gamma^0 \delta^6 (\gamma^1 \delta^4)^* & \gamma^0 \delta^9 (\gamma^1 \delta^4)^* & \gamma^0 \delta^5 (\gamma^1 \delta^4)^* & \gamma^0 \delta^4 (\gamma^1 \delta^4)^* & \gamma^0 \delta^0 (\gamma^1 \delta^4)^* \\ \gamma^0 \delta^3 (\gamma^1 \delta^3)^* & \gamma^0 \delta^0 (\gamma^1 \delta^3)^* & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \gamma^0 \delta^4 (\gamma^1 \delta^4)^* & \gamma^0 \delta^0 (\gamma^1 \delta^4)^* & \cdot & \cdot \end{pmatrix}$$

Pour l'état x_1 , d'après la ligne 1 de la matrice L , $N_1 = 1$. D'après la série $b = CA^*R$, $\nu_0^b = 0$. Enfin, grâce à l'entrée u , on obtient les exposants de γ de l'état x_1 et donc les exposants de γ de w_1 , soit $\nu_0^{w_1} = 0$. Donc, $\nu_d = 1 + 0 + 0 = 1$. On réitère le procédé pour les autres états afin d'obtenir le tableau suivant.

Transition	N_i	ν_0^b	$\nu_0^{w_i}$	ν_d
x_1	1	0	0	1
x_2	0	0	0	0
x_3	1	0	0	1
x_4	0	0	0	0
x_5	0	0	0	0
x_6	0	0	0	0

TABLE 6.2 – Tableau récapitulatif des ν_d par états estimés

L'état estimé *sensible-aux-perturbations* SAP x_{ed} est obtenu en ne conservant que les monômes dont les exposants de γ de l'état x_e prennent en compte la perturbation. Ces exposants dépendent du ν_d de la proposition précédente.

Définition 6.2. L'état estimé *sensible-aux-perturbations* SAP x_{edi} est défini à partir de l'état estimé $x_{ei} = \bigoplus_{\kappa=0}^K \gamma^{\nu_\kappa} \delta^{\tau_\kappa}$ par

$$x_{edi} = \bigoplus_{k=d}^K \gamma^{\nu_k} \delta^{\tau_k}$$

où ν_d est défini par la proposition 6.3.

Exemple 6.11. Dans le GET de la figure 6.3 page 85 (exemples 6.8 et 6.9), les états estimés sont :

$$x_e = \begin{pmatrix} \gamma^0 \delta^2 \oplus \gamma^1 \delta^5 \oplus \gamma^2 \delta^8 \oplus \gamma^3 \delta^{11} \oplus \gamma^4 \delta^{14} \oplus \gamma^5 \delta^{17} \oplus \gamma^6 \delta^{20} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^5 \oplus \gamma^1 \delta^8 \oplus \gamma^2 \delta^{11} \oplus \gamma^3 \delta^{14} \oplus \gamma^4 \delta^{17} \oplus \gamma^5 \delta^{20} \oplus \gamma^6 \delta^{23} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^2 \oplus \gamma^1 \delta^6 \oplus \gamma^2 \delta^{10} \oplus \gamma^3 \delta^{14} \oplus \gamma^4 \delta^{18} \oplus \gamma^5 \delta^{22} \oplus \gamma^6 \delta^{26} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^6 \oplus \gamma^1 \delta^{10} \oplus \gamma^2 \delta^{14} \oplus \gamma^3 \delta^{18} \oplus \gamma^4 \delta^{22} \oplus \gamma^5 \delta^{26} \oplus \gamma^6 \delta^{30} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^7 \oplus \gamma^1 \delta^{12} \oplus \gamma^2 \delta^{16} \oplus \gamma^3 \delta^{20} \oplus \gamma^4 \delta^{24} \oplus \gamma^5 \delta^{28} \oplus \gamma^6 \delta^{32} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^{12} \oplus \gamma^1 \delta^{16} \oplus \gamma^2 \delta^{20} \oplus \gamma^3 \delta^{24} \oplus \gamma^4 \delta^{28} \oplus \gamma^5 \delta^{32} \oplus \gamma^6 \delta^{36} \oplus \gamma^7 \delta^{+\infty} \end{pmatrix}.$$

Pour x_i avec $i \in \{1, 2, 3, 4\}$, puisque $\nu_d = +\infty$, les états estimés SAP correspondants sont égaux à ε . Pour l'état x_5 , puisque $\nu_d = 1$, le premier monôme de la série de l'état estimé SAP est $\gamma^1 \delta^{12}$. Pour l'état x_6 , la perturbation est prise en compte dès le premier monôme de x_{e6} . Cela donne donc les états estimés SAP suivants :

$$x_{ed} = \begin{pmatrix} \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \gamma^1 \delta^{12} \oplus \gamma^2 \delta^{16} \oplus \gamma^3 \delta^{20} \oplus \gamma^4 \delta^{24} \oplus \gamma^5 \delta^{28} \oplus \gamma^6 \delta^{32} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^{12} \oplus \gamma^1 \delta^{16} \oplus \gamma^2 \delta^{20} \oplus \gamma^3 \delta^{24} \oplus \gamma^4 \delta^{28} \oplus \gamma^5 \delta^{32} \oplus \gamma^6 \delta^{36} \oplus \gamma^7 \delta^{+\infty} \end{pmatrix}.$$

Exemple 6.12. Dans le GET de la figure 6.5 page 89 (exemples 6.4 et 6.10), pour les états estimés suivants :

$$x_e = \begin{pmatrix} \gamma^0 \delta^2 \oplus \gamma^1 \delta^6 \oplus \gamma^2 \delta^{10} \oplus \gamma^3 \delta^{14} \oplus \gamma^4 \delta^{18} \oplus \gamma^5 \delta^{22} \oplus \gamma^6 \delta^{26} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^6 \oplus \gamma^1 \delta^{10} \oplus \gamma^2 \delta^{14} \oplus \gamma^3 \delta^{18} \oplus \gamma^4 \delta^{22} \oplus \gamma^5 \delta^{26} \oplus \gamma^6 \delta^{30} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^2 \oplus \gamma^1 \delta^6 \oplus \gamma^2 \delta^{10} \oplus \gamma^3 \delta^{14} \oplus \gamma^4 \delta^{18} \oplus \gamma^5 \delta^{22} \oplus \gamma^6 \delta^{26} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^6 \oplus \gamma^1 \delta^{10} \oplus \gamma^2 \delta^{14} \oplus \gamma^3 \delta^{18} \oplus \gamma^4 \delta^{22} \oplus \gamma^5 \delta^{26} \oplus \gamma^6 \delta^{30} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^8 \oplus \gamma^1 \delta^{12} \oplus \gamma^2 \delta^{16} \oplus \gamma^3 \delta^{20} \oplus \gamma^4 \delta^{24} \oplus \gamma^5 \delta^{28} \oplus \gamma^6 \delta^{32} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^{12} \oplus \gamma^1 \delta^{16} \oplus \gamma^2 \delta^{20} \oplus \gamma^3 \delta^{24} \oplus \gamma^4 \delta^{28} \oplus \gamma^5 \delta^{32} \oplus \gamma^6 \delta^{36} \oplus \gamma^7 \delta^{+\infty} \end{pmatrix},$$

les états estimés SAP sont :

$$x_{ed} = \begin{pmatrix} \gamma^1 \delta^6 \oplus \gamma^2 \delta^{10} \oplus \gamma^3 \delta^{14} \oplus \gamma^4 \delta^{18} \oplus \gamma^5 \delta^{22} \oplus \gamma^6 \delta^{26} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^6 \oplus \gamma^1 \delta^{10} \oplus \gamma^2 \delta^{14} \oplus \gamma^3 \delta^{18} \oplus \gamma^4 \delta^{22} \oplus \gamma^5 \delta^{26} \oplus \gamma^6 \delta^{30} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^1 \delta^6 \oplus \gamma^2 \delta^{10} \oplus \gamma^3 \delta^{14} \oplus \gamma^4 \delta^{18} \oplus \gamma^5 \delta^{22} \oplus \gamma^6 \delta^{26} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^6 \oplus \gamma^1 \delta^{10} \oplus \gamma^2 \delta^{14} \oplus \gamma^3 \delta^{18} \oplus \gamma^4 \delta^{22} \oplus \gamma^5 \delta^{26} \oplus \gamma^6 \delta^{30} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^8 \oplus \gamma^1 \delta^{12} \oplus \gamma^2 \delta^{16} \oplus \gamma^3 \delta^{20} \oplus \gamma^4 \delta^{24} \oplus \gamma^5 \delta^{28} \oplus \gamma^6 \delta^{32} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^{12} \oplus \gamma^1 \delta^{16} \oplus \gamma^2 \delta^{20} \oplus \gamma^3 \delta^{24} \oplus \gamma^4 \delta^{28} \oplus \gamma^5 \delta^{32} \oplus \gamma^6 \delta^{36} \oplus \gamma^7 \delta^{+\infty} \end{pmatrix}.$$

Seuls les états x_1 et x_3 sont impactés par les ν_d différents de 0.

6.2.4 Indicateur avec les états estimés SAP

Avec le raffinement de l'état estimé SAP x_{ed} de la section précédente, un nouvel indicateur de dérive temporelle est proposé dans cette section. Les hypothèses du précédent indicateur ne changent pas. En revanche, pour effectuer une comparaison cohérente entre x_{ed} et x_s , les occurrences d'événements de x_s qui ont lieu avant ν_d doivent également être enlevées.

Définition 6.3. Soit $x_{si} = A^* B u = \bigoplus_{k=0}^K \gamma^{\nu_k} \delta^{\tau_k}$ l'état du modèle sans faute, la série x_{si} désigne l'état du modèle sans faute comparable à x_{edi} :

$$x_{si} = \bigoplus_{k=d}^K \gamma^{\nu_k} \delta^{\tau_k}.$$

Exemple 6.13. À partir de l'exemple 6.11, l'état du système sans faute est le suivant :

$$x_s = \begin{pmatrix} \gamma^0 \delta^2 \oplus \gamma^1 \delta^5 \oplus \gamma^2 \delta^8 \oplus \gamma^3 \delta^{11} \oplus \gamma^4 \delta^{14} \oplus \gamma^5 \delta^{17} \oplus \gamma^6 \delta^{20} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^5 \oplus \gamma^1 \delta^8 \oplus \gamma^2 \delta^{11} \oplus \gamma^3 \delta^{14} \oplus \gamma^4 \delta^{17} \oplus \gamma^5 \delta^{20} \oplus \gamma^6 \delta^{23} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^2 \oplus \gamma^1 \delta^6 \oplus \gamma^2 \delta^{10} \oplus \gamma^3 \delta^{14} \oplus \gamma^4 \delta^{18} \oplus \gamma^5 \delta^{22} \oplus \gamma^6 \delta^{26} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^6 \oplus \gamma^1 \delta^{10} \oplus \gamma^2 \delta^{14} \oplus \gamma^3 \delta^{18} \oplus \gamma^4 \delta^{22} \oplus \gamma^5 \delta^{26} \oplus \gamma^6 \delta^{30} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^7 \oplus \gamma^1 \delta^{11} \oplus \gamma^2 \delta^{15} \oplus \gamma^3 \delta^{19} \oplus \gamma^4 \delta^{23} \oplus \gamma^5 \delta^{27} \oplus \gamma^6 \delta^{31} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^{11} \oplus \gamma^1 \delta^{15} \oplus \gamma^2 \delta^{19} \oplus \gamma^3 \delta^{23} \oplus \gamma^4 \delta^{27} \oplus \gamma^5 \delta^{31} \oplus \gamma^6 \delta^{35} \oplus \gamma^7 \delta^{+\infty} \end{pmatrix}.$$

D'après les valeurs des ν_d calculées, les états du modèle sans faute comparable à x_{ed} sont :

$$x_s' = \begin{pmatrix} \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \gamma^1 \delta^{11} \oplus \gamma^2 \delta^{15} \oplus \gamma^3 \delta^{19} \oplus \gamma^4 \delta^{23} \oplus \gamma^5 \delta^{27} \oplus \gamma^6 \delta^{31} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^{11} \oplus \gamma^1 \delta^{15} \oplus \gamma^2 \delta^{19} \oplus \gamma^3 \delta^{23} \oplus \gamma^4 \delta^{27} \oplus \gamma^5 \delta^{31} \oplus \gamma^6 \delta^{35} \oplus \gamma^7 \delta^{+\infty} \end{pmatrix}.$$

Exemple 6.14. À partir de l'exemple 6.12, l'état du système sans faute est le suivant :

$$x_s = \begin{pmatrix} \gamma^0\delta^2 \oplus \gamma^1\delta^5 \oplus \gamma^2\delta^8 \oplus \gamma^3\delta^{11} \oplus \gamma^4\delta^{14} \oplus \gamma^5\delta^{17} \oplus \gamma^6\delta^{20} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^5 \oplus \gamma^1\delta^8 \oplus \gamma^2\delta^{11} \oplus \gamma^3\delta^{14} \oplus \gamma^4\delta^{17} \oplus \gamma^5\delta^{20} \oplus \gamma^6\delta^{23} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^2 \oplus \gamma^1\delta^6 \oplus \gamma^2\delta^{10} \oplus \gamma^3\delta^{14} \oplus \gamma^4\delta^{18} \oplus \gamma^5\delta^{22} \oplus \gamma^6\delta^{26} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^6 \oplus \gamma^1\delta^{10} \oplus \gamma^2\delta^{14} \oplus \gamma^3\delta^{18} \oplus \gamma^4\delta^{22} \oplus \gamma^5\delta^{26} \oplus \gamma^6\delta^{30} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^7 \oplus \gamma^1\delta^{11} \oplus \gamma^2\delta^{15} \oplus \gamma^3\delta^{19} \oplus \gamma^4\delta^{23} \oplus \gamma^5\delta^{27} \oplus \gamma^6\delta^{31} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^{11} \oplus \gamma^1\delta^{15} \oplus \gamma^2\delta^{19} \oplus \gamma^3\delta^{23} \oplus \gamma^4\delta^{27} \oplus \gamma^5\delta^{31} \oplus \gamma^6\delta^{35} \oplus \gamma^7\delta^{+\infty} \end{pmatrix}.$$

D'après les valeurs des ν_d calculées, les états du modèle sans faute comparable à x_{ed} sont :

$$x'_s = \begin{pmatrix} \gamma^1\delta^5 \oplus \gamma^2\delta^8 \oplus \gamma^3\delta^{11} \oplus \gamma^4\delta^{14} \oplus \gamma^5\delta^{17} \oplus \gamma^6\delta^{20} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^5 \oplus \gamma^1\delta^8 \oplus \gamma^2\delta^{11} \oplus \gamma^3\delta^{14} \oplus \gamma^4\delta^{17} \oplus \gamma^5\delta^{20} \oplus \gamma^6\delta^{23} \oplus \gamma^7\delta^{+\infty} \\ \gamma^1\delta^6 \oplus \gamma^2\delta^{10} \oplus \gamma^3\delta^{14} \oplus \gamma^4\delta^{18} \oplus \gamma^5\delta^{22} \oplus \gamma^6\delta^{26} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^6 \oplus \gamma^1\delta^{10} \oplus \gamma^2\delta^{14} \oplus \gamma^3\delta^{18} \oplus \gamma^4\delta^{22} \oplus \gamma^5\delta^{26} \oplus \gamma^6\delta^{30} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^7 \oplus \gamma^1\delta^{11} \oplus \gamma^2\delta^{15} \oplus \gamma^3\delta^{19} \oplus \gamma^4\delta^{23} \oplus \gamma^5\delta^{27} \oplus \gamma^6\delta^{31} \oplus \gamma^7\delta^{+\infty} \\ \gamma^0\delta^{11} \oplus \gamma^1\delta^{15} \oplus \gamma^2\delta^{19} \oplus \gamma^3\delta^{23} \oplus \gamma^4\delta^{27} \oplus \gamma^5\delta^{31} \oplus \gamma^6\delta^{35} \oplus \gamma^7\delta^{+\infty} \end{pmatrix}.$$

Définition 6.4. Soit x_{edi} l'état estimé SAP de x_i , différent de ε , et x'_{si} l'état comparable du modèle sans faute (donc également différent de ε), l'indicateur $I'_{x_i}(u, y_o)$ de l'état x_i est défini par la fonction booléenne :

$$I'_{x_i}(u, y_o) = \begin{cases} \text{faux si } \Delta(x_{edi}, x'_{si}) = [0; 0], \\ \text{vrai sinon,} \end{cases}$$

avec $\Delta(x_{edi}, x'_{si}) = [\mathcal{D}_{x_{edi} \neq x'_{si}}(0); -\mathcal{D}_{x'_{si} \neq x_{edi}}(0)]$.

Dans cette définition, si l'état estimé SAP est égal à ε , l'indicateur n'existe tout simplement pas.

Proposition 6.4. L'indicateur I'_{x_i} renvoie vrai si et seulement si une dérive temporelle impacte la trajectoire de l'état x_i par rapport à son comportement sans faute.

Démonstration. Tout d'abord, rappelons qu'un indicateur I' n'existe pour un état que si cet état est estimé SAP, donc prend en compte les perturbations. En effet, d'après la proposition 6.1, si la ligne de la matrice L pour cet état n'est composée que de ε , l'état n'est pas estimé SAP donc l'indicateur $I'_{x_i}(u, y_o)$ ne peut pas être calculé.

Ensuite, pour les indicateurs calculés, la preuve pour l'implication (\Rightarrow) est la même que celle de la proposition 6.2 en remplaçant x_e par x_{ed} et x_s par x'_s . Cela établit que si l'indicateur renvoie vrai, alors une dérive temporelle a lieu car elle impacte la trajectoire de l'état x_i .

Pour l'implication (\Leftarrow) , si une dérive temporelle a lieu sur une place p , alors d'après la définition 4.4 du chapitre 4, au moins une sortie observée en aval de p ($p \rightsquigarrow y_{oi}$) est impactée par la dérive. Or, si les dates de tirs d'une transition de sortie y_{oi} sont différentes des dates de tirs lorsqu'aucune dérive n'est présente dans le système, cela signifie qu'au moins un des états en amont de cette sortie ($x_i \rightsquigarrow y_{oi}$) est nécessairement impacté également. Les états estimés SAP prenant obligatoirement en compte les perturbations, et donc les dérives, l'état estimé SAP concerné x_{edi} voit alors ses dates de tir modifiées par rapport à celles sans dérive. L'indicateur renvoie vrai. \square

Exemple 6.15. D'après les exemples 6.11-6.13, les différents indicateurs sont :

$$\begin{aligned} I'_{x_5}(u, y_o) &= \text{vrai car } \Delta(x_{ed5}, x'_{s5}) = [1, 1], \\ I'_{x_6}(u, y_o) &= \text{vrai car } \Delta(x_{ed6}, x'_{s6}) = [1, 1]. \end{aligned}$$

Les indicateurs pour les états x_i avec $i \in \{1, 2, 3, 4\}$ n'existent pas car les perturbations ne sont pas prises en compte pour ces états (les états estimés SAP sont égaux à ε). Puisque les indicateurs sur x_5 et x_6 sont égaux à vrai, la dérive temporelle est détectée.

Exemple 6.16. D'après les exemples 6.12-6.14, les différents indicateurs sont :

$$\begin{aligned} I'_{x_1}(u, y_o) &= \text{vrai car } \Delta(x_{ed1}, x'_{s1}) = [1, 6], \\ I'_{x_2}(u, y_o) &= \text{vrai car } \Delta(x_{ed2}, x'_{s2}) = [1, 7], \\ I'_{x_3}(u, y_o) &= \text{faux car } \Delta(x_{ed3}, x'_{s3}) = [0, 0], \\ I'_{x_4}(u, y_o) &= \text{faux car } \Delta(x_{ed4}, x'_{s4}) = [0, 0], \\ I'_{x_5}(u, y_o) &= \text{vrai car } \Delta(x_{ed5}, x'_{s5}) = [1, 1], \\ I'_{x_6}(u, y_o) &= \text{vrai car } \Delta(x_{ed6}, x'_{s6}) = [1, 1]. \end{aligned}$$

Dans cet exemple, les perturbations sont prises en compte pour tous les états, nous avons donc des valeurs pour tous les indicateurs. Puisque certains indicateurs sont égaux à vrai, la dérive temporelle est détectée.

Complexité algorithmique

Pour déterminer la complexité de l'indicateur, on doit dans un premier temps distinguer les deux parties avec les entrées u et w du calcul de l'état estimé $x_e = (A \oplus LC)^*Bu \oplus (A \oplus LC)^*LCA^*Rw$ et l'état du modèle sans faute $x_s = A^*Bu$.

Tout d'abord, on se place dans la partie du calcul qui ne considère pas les entrées u et w . Ces parties sont A^*B pour l'état du modèle sans faute x_s , $(A \oplus LC)^*B$ et $(A \oplus LC)^*LCA^*R$ pour l'état estimé x_e . Pour chaque partie, la complexité dans le pire des cas est basée sur le calcul de A^* et $(A \oplus LC)^*$. Les matrices A et $(A \oplus LC)$ sont composées de monômes de dimension n_T des transitions internes du GET. Le calcul des $*$ est un algorithme implémenté dans la bibliothèque [CLHB00] et est un nombre linéaire de produits, chaque produit ayant une complexité de $O(n_T^2 \log(n_T))$. Donc la partie de ce calcul est de complexité $O(n_T^3 \log(n_T))$.

Ensuite, les entrées sont une suite de monômes n_u et n_w correspondant à un nombre fini d'événements. La complexité de $x_s = A^*Bu$ et de $x_e = (A \oplus LC)^*Bu \oplus (A \oplus LC)^*LCA^*Rw$ est donnée par le calcul de A^* et $(A \oplus LC)^*$ qui ont une complexité de $O(n_u \times n_T^3 \log(n_T))$ et de $O(n_w \times n_T^3 \log(n_T))$. Les autres calculs sont les résiduations $x_s \neq x_e$ et $x_e \neq x_s$ qui impliquent un calcul sur des polynômes. Comme les états x_e et x_s dépendent du nombre de monômes des entrées n_u cela donne la complexité suivante $O(n_w \times n_u \log(n_w \times n_u))$.

6.3 Localisation de dérives temporelles

Cette section présente la localisation de dérives temporelles dans les GET à partir des résultats des indicateurs présentés dans la section précédente. L'hypothèse principale spécifique à la localisation est qu'à tout instant, il n'y a qu'une seule dérive temporelle permanente θ présente dans le système. Cette étape de localisation se fait par structure élémentaire présentée dans la figure 4.3 du chapitre 4 : tandem, parallélisme, synchronisation et boucle. Enfin, des exemples d'association des résultats locaux par structure pour des GET complets sont proposés.

Pour l'étude par structure élémentaire, différentes configurations sont prises en compte. Une configuration est un ensemble d'intervales de décalage temporel issus des indicateurs *calculés* (voir exemples 6.15 et 6.16) peu importe leurs valeurs (que ces indicateurs renvoient vrai ou faux). Une configuration correspond donc à un ensemble d'états estimés SAP fourni par l'observateur. De plus, pour les structures tandem, parallélisme et synchronisation, les résultats sont donnés pour toute valeur de ν_d . En revanche, pour la structure boucle, les résultats sont donnés uniquement pour $\nu_d = 0$ (les résultats associés à d'autres valeurs de ν_d sont dans les perspectives de ce travail).

6.3.1 Tandem

Cette partie présente les résultats obtenus sur la localisation pour la structure élémentaire tandem rappelé dans la figure 6.8 et représentant une simple séquence de places et de transitions mises en série.

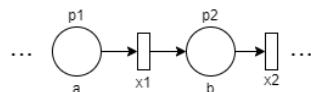


FIGURE 6.8 – Structure élémentaire tandem

Le tableau 6.3 donne les résultats de localisation d'une dérive temporelle présente dans cette structure tandem selon toutes les configurations possibles. Chaque ligne correspond à une configuration des intervalles des indicateurs (si l'indicateur est calculé, l'intervalle est donné; s'il ne l'est pas, l'état n'est pas estimé SAP et n'apparaît pas dans la ligne). La dernière colonne donne la conclusion sur la localisation de la dérive temporelle dans la structure analysée. Dans ce tableau (et par la suite), la notation (p_i) signifie que la dérive temporelle concerne soit la place p_i , soit l'une des places situées dans le chemin qui mène de l'une des entrées à cette place (place en amont

de p_i). La notation p_i signifie quant à elle que la dérive temporelle impacte directement cette place. Enfin, \emptyset indique qu'il n'y a pas de dérive temporelle.

	$\Delta(x_{ed1}, x'_{s1})$	$\Delta(x_{ed2}, x'_{s2})$	Localisation
c_1	$[0,0]$	$[0,0]$	\emptyset
c_2	$[\theta, \theta]$	$[\theta, \theta]$	(p_1)
c_3	$[0,0]$	$[\theta, \theta]$	p_2
	$\Delta(x_{ed2}, x'_{s2})$		Localisation
c_4	$[0,0]$		\emptyset
c_5	$[\theta, \theta]$		(p_2)

TABLE 6.3 – Résultat de localisation de la dérive temporelle pour une structure tandem

Le premier ensemble de configurations (lignes c_1-c_3) correspond aux cas où x_1 et x_2 sont associés à un indicateur chacun, ils sont donc estimés SAP. Prenons l'exemple de la **ligne c₁** pour laquelle les indicateurs renvoient faux car $\Delta(x_{ed1}, x'_{s1}) = [0, 0]$ et $\Delta(x_{ed2}, x'_{s2}) = [0, 0]$. Ces valeurs d'intervalles signifient qu'il n'y a pas de dérive temporelle avant la transition x_2 . En revanche, dans la **ligne c₂**, les indicateurs renvoient vrai avec $\Delta(x_{ed1}, x'_{s1}) = [\theta, \theta]$ et $\Delta(x_{ed2}, x'_{s2}) = [\theta, \theta]$. La dérive temporelle de valeur θ est donc localisée soit sur la place p_1 , soit sur une place en amont comme le montre la démonstration suivante.

Démonstration. Soient $x'_{s1} = \bigoplus_{k=0}^K \gamma^{n_k} \delta^{t_k+a}$ et $x'_{s2} = \bigoplus_{k=0}^K \gamma^{n_k} \delta^{t_k+a+b}$ les trajectoires des transitions du modèle sans faute exprimées sous leurs formes canoniques. Si une dérive temporelle θ a lieu sur p_1 ou une place en amont, la trajectoire de la transition x_1 est affectée par la dérive, alors $x_{ed1} = \bigoplus_{k=d_1}^K \gamma^{n_k} \delta^{t_k+a+\theta}$. Compte tenu de la structure tandem, la trajectoire de la transition x_2 l'est également $x_{ed2} = \bigoplus_{k=d_2}^K \gamma^{n_k} \delta^{t_k+a+\theta+b}$. Seule la valeur de la dérive temporelle θ fait la différence entre x'_{s1} et x_{ed1} , de même pour la transition x_2 . \square

Enfin, dans la **ligne c₃**, les valeurs des intervalles sont $\Delta(x_{ed1}, x'_{s1}) = [0, 0]$ et $\Delta(x_{ed2}, x'_{s2}) = [\theta, \theta]$. Cela signifie que la dérive temporelle θ est sur la place p_2 car l'indicateur de l'état x_1 renvoie faux ($\Delta(x_{ed1}, x'_{s1}) = [0, 0]$, il n'y a pas de dérive temporelle sur p_1 ou en amont).

À présent, le deuxième ensemble de configurations (lignes c_4-c_5) correspond au cas où seul l'état x_2 du tandem est associé à un indicateur donc les conclusions sont différentes des lignes précédentes. Par exemple, comme le montre la **ligne c₅**, l'indicateur renvoie vrai avec $\Delta(x_{ed2}, x'_{s2}) = [\theta, \theta]$ mais nous n'avons pas d'informations sur la transition x_1 . La distinction faite entre les lignes c_2 et c_3 n'est donc plus possible. Ainsi, le résultat de localisation est une association entre les résultats (p_1) et p_2 : la localisation de la dérive est soit sur la place p_2 , soit sur une place en amont, donc (p_2).

Exemple 6.17. D'après le GET de la figure 6.8 avec les durées suivantes pour les places p_1 et p_2 : $a = 1$ et $b = 2$. Soit une dérive temporelle $\theta = 2$ sur la place p_2 . Avec l'entrée suivante :

$$u = \gamma^0 \delta^1 \oplus \gamma^1 \delta^2 \oplus \gamma^2 \delta^3 \oplus \gamma^3 \delta^4 \oplus \gamma^4 \delta^5 \oplus \gamma^5 \delta^6 \oplus \gamma^6 \delta^7 \oplus \gamma^7 \delta^{+\infty},$$

les différents indicateurs sont :

$$\begin{aligned} I'_{x_1}(u, y_0) &= \text{faux} \quad \text{car} \quad \Delta(x_{ed1}, x'_{s1}) = [0, 0], \\ I'_{x_2}(u, y_0) &= \text{vrai} \quad \text{car} \quad \Delta(x_{ed2}, x'_{s2}) = [2, 2]. \end{aligned}$$

D'après la ligne c_3 du tableau 6.3, la dérive temporelle de valeur $\theta = 2$ est bien localisée sur la place p_2 .

6.3.2 Parallélisme

Cette partie présente les résultats de localisation pour la structure élémentaire de parallélisme rappelée sur la figure 6.9. Cette structure correspond simplement à plusieurs structures tandem partant d'une transition commune.

La tableau 6.4 donne les résultats obtenus pour la structure parallèle pour toutes les configurations d'estimation d'états SAP.

Dans la configuration des lignes c_1-c_4 , les trois transitions sont estimées SAP. La **ligne c₁** qui indique $\Delta(x_{ed1}, x'_{s1}) = [0, 0]$, $\Delta(x_{ed2}, x'_{s2}) = [0, 0]$ et $\Delta(x_{ed3}, x'_{s3}) = [0, 0]$ signifie qu'il n'y a pas de

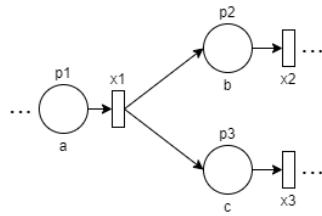


FIGURE 6.9 – Structure élémentaire de parallélisme

	$\Delta(x_{ed1}, x'_{s1})$	$\Delta(x_{ed2}, x'_{s2})$	$\Delta(x_{ed3}, x'_{s3})$	Localisation
c₁	[0,0]	[0,0]	[0,0]	\emptyset
c₂	[0,0]	$[\theta, \theta]$	[0,0]	p_2
c₃	[0,0]	[0,0]	$[\theta, \theta]$	p_3
c₄	$[\theta, \theta]$	$[\theta, \theta]$	$[\theta, \theta]$	(p_1)
	$\Delta(x_{ed2}, x'_{s2})$	$\Delta(x_{ed3}, x'_{s3})$	Localisation	
c₅	[0,0]	[0,0]	\emptyset	
c₆	$[\theta, \theta]$	[0,0]	p_2	
c₇	[0,0]	$[\theta, \theta]$	p_3	
c₈	$[\theta, \theta]$	$[\theta, \theta]$	(p_1)	

TABLE 6.4 – Résultat de localisation de la dérive temporelle pour une structure parallèle

dérive temporelle. Pour la **ligne c₂**, les intervalles associés à x_1 et x_3 sont égaux à $[0, 0]$ tandis que l'intervalle associé à x_2 est égal à $[\theta, \theta]$. Cela signifie que la dérive temporelle est localisée sur la place p_2 car l'enchaînement des transitions x_1 et x_2 correspond à la structure élémentaire du tandem. L'analyse de la **ligne c₃** est la même pour la place p_3 . Enfin, dans la **ligne c₄**, puisque tous les indicateurs retournent vrai, la dérive temporelle est située sur ou en amont de p_1 et son effet est propagé par la structure sur les transitions en aval.

Pour la configuration des **lignes c₅-c₈**, seuls les indicateurs des états x_2 et x_3 sont calculés. Les conclusions sur la localisation de la dérive découlent naturellement des conclusions précédentes (tandem et parallélisme).

Exemple 6.18. Soit le GET de la figure 6.9 avec les durées suivantes pour les places : $a = 1$, $b = 2$ et $c = 2$ et soit une dérive temporelle $\theta = 3$ sur la place p_2 . Avec l'entrée suivante :

$$u = \gamma^0 \delta^1 \oplus \gamma^1 \delta^2 \oplus \gamma^2 \delta^3 \oplus \gamma^3 \delta^4 \oplus \gamma^4 \delta^5 \oplus \gamma^5 \delta^6 \oplus \gamma^6 \delta^7 \oplus \gamma^7 \delta^{+\infty},$$

les différents indicateurs sont :

$$\begin{aligned} I'_{x_1}(u, y_o) &= \text{faux car } \Delta(x_{ed1}, x'_{s1}) = [0, 0], \\ I'_{x_2}(u, y_o) &= \text{vrai car } \Delta(x_{ed2}, x'_{s2}) = [3, 3], \\ I'_{x_3}(u, y_o) &= \text{faux car } \Delta(x_{ed3}, x'_{s3}) = [0, 0]. \end{aligned}$$

La ligne c_2 du tableau 6.4 est la ligne qui correspond à ces résultats. Elle indique que la dérive temporelle de valeur $\theta = 3$ est correctement localisée sur la place p_2 .

6.3.3 Synchronisation

La structure élémentaire de la synchronisation est rappelée dans la figure 6.10 et correspond à plusieurs tandems réunis sur une transition commune. Dans cette figure, les transitions d'entrée u_1 et u_2 , reliées aux places p_1, p_2 , ne font pas partie de la structure de synchronisation analysée dans cette section mais sont nécessaires aux résultats. En effet, la localisation dépend de la façon dont les jetons arrivent dans p_1 et p_2 . Pour effectuer l'analyse exhaustive, ces deux entrées sont données par $u_1 = \bigoplus_{n \in \mathbb{Z}} \gamma^n \delta^{D_{u_1}(n)}$ et $u_2 = \bigoplus_{n \in \mathbb{Z}} \gamma^n \delta^{D_{u_2}(n)}$ avec $D_{u_1}(n)$ et $D_{u_2}(n)$ leur fonction dateur respective (voir la définition 5.28).

Trois configurations d'estimation SAP sont détaillées dans trois tableaux différents :

- le tableau 6.5 regroupe les **lignes c₁-c₅** pour lesquelles tous les x_i sont estimés SAP,
- le tableau 6.6 présente les **lignes c₆-c₉** pour lesquelles x_3 et $\{x_1 \text{ ou } x_2\}$ sont estimés SAP,
- seul l'état x_3 est estimé SAP dans les **lignes c₁₀-c₁₂** du tableau 6.7.

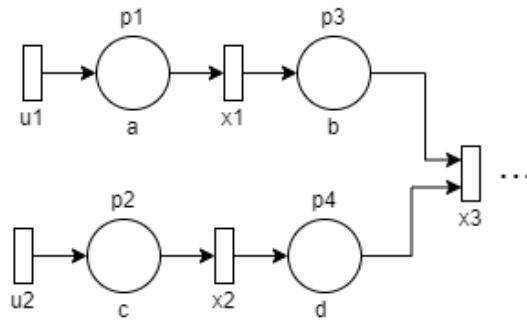


FIGURE 6.10 – Structure élémentaire de synchronisation

Tableau 6.5 (lignes c_1-c_5) : tous les x_i sont estimés SAP

	$\Delta(x_{ed1}, x'_{s1})$	$\Delta(x_{ed2}, x'_{s2})$	$\Delta(x_{ed3}, x'_{s3})$	Localisation
c_1	[0,0]	[0,0]	[0,0]	\emptyset
c_2	[θ, θ]	[0,0]	[0,0] ou [0, θ] ou [θ, θ]	(p_1)
c_3	[0,0]	[θ, θ]	[0,0] ou [0, θ] ou [θ, θ]	(p_2)
c_4	[0,0]	[0,0]	[θ, θ]	$\forall n \in \mathbb{Z}, t_1 = D_{u_1}(n), t_2 = D_{u_2}(n)$ p_3 si $a + b + \theta + t_1 > c + d + t_2$ p_4 si $a + b + t_1 < c + d + \theta + t_2$
c_5	[0,0]	[0,0]	[0, θ]	$\exists n \in \mathbb{Z}, t_1 = D_{u_1}(n), t_2 = D_{u_2}(n)$ p_3 si $a + b + \theta + t_1 > c + d + t_2$ p_4 si $a + b + t_1 < c + d + \theta + t_2$

TABLE 6.5 – Lignes c_1-c_5 : tous les x_i estimés SAP

Dans la **ligne c_1** pour laquelle les indicateurs renvoient faux car les intervalles sont tous égaux à [0,0], il n'y a pas de dérive temporelle. Pour la **ligne c_2** , l'indicateur renvoie faux pour l'état x_2 . Cette partie de la structure n'est donc pas concernée par la dérive temporelle. En revanche, l'indicateur associé à l'état x_1 renvoie vrai, tandis que l'indicateur associé à l'état x_3 renvoie plusieurs combinaisons possibles (faux, vrai, avec plusieurs valeurs d'intervalles possibles quand l'indicateur renvoie vrai). Cela signifie que la dérive temporelle θ est localisée sur la place p_1 ou en amont et que son effet se propage dans la structure sur les places en aval de p_1 et donc sur l'état x_3 (comme pour la ligne c_2 du tableau 6.3 de la structure tandem pour laquelle l'intervalle associé à x_3 est alors égal à [θ, θ]) ou non (absorption de l'effet par la synchronisation, l'intervalle associé à x_3 est alors égal à [0,0]) ou les deux (absorption puis propagation de l'effet, l'intervalle associé à x_3 est alors égal à [0, θ], ce phénomène est expliqué dans la ligne c_5 du tableau 6.5). La réflexion est identique pour la **ligne c_3** avec la partie supérieure de la structure.

Dans la **ligne c_4** , les intervalles associés à x_1 et x_2 sont égaux à [0,0], la dérive n'est donc pas située sur la place p_1 ou p_2 (voir résultats de la structure tandem). Ensuite, l'intervalle associé à x_3 est égal à [θ, θ], deux situations sont alors possibles pour tout $n \in \mathbb{Z}$, soit $t_1 = D_{u_1}(n)$, $t_2 = D_{u_2}(n)$:

- Lorsque $a + \theta + b + t_1 > c + d + t_2$, cela traduit le fait que la dérive temporelle se situe sur le chemin $u_1 \rightsquigarrow x_3$ (chemin qui contient les places p_1 et p_3), chemin le plus long par rapport au chemin $u_2 \rightsquigarrow x_3$ (qui contient les places p_2 et p_4). Comme la dérive temporelle θ n'est pas située sur p_1 ($\Delta(x_{ed1}, x'_{s1}) = [0,0]$), elle est nécessairement sur la place p_3 .
- Lorsque $a + b + t_1 < c + d + \theta + t_2$, la réflexion est symétrique, la dérive temporelle θ est nécessairement sur la place p_4 .

La **ligne c_5** décrit le cas où l'effet de la dérive temporelle θ est intermittent, ce qui conduit à $\Delta(x_{ed3}, x'_{s3}) = [0, \theta]$. Dans cet intervalle, la présence d'une borne égale à 0 signifie en effet que la dérive n'est pas continuellement détectée. Les dates de tir des transitions d'entrée u_1 et u_2 font que le chemin le plus long entre $u_1 \rightsquigarrow x_3$ et $u_2 \rightsquigarrow x_3$ varie et que la dérive est masquée ou non par la synchronisation. Ainsi, les résultats de localisation sont identiques à ceux de la ligne c_4 sous la condition "il existe une entrée" ($\exists n \in \mathbb{Z}$) au lieu de la condition "pour toute entrée" ($\forall n \in \mathbb{Z}$).

Exemple 6.19. Soit le GET de la figure 6.10 avec les durées : $a = 1, b = 2, c = 3$ et $d = 4$ et une dérive temporelle $\theta = 1$ sur la place p_4 . Avec l'entrée suivante :

$$u_1 = u_2 = \gamma^0\delta^1 \oplus \gamma^1\delta^2 \oplus \gamma^2\delta^3 \oplus \gamma^3\delta^4 \oplus \gamma^4\delta^5 \oplus \gamma^5\delta^6 \oplus \gamma^6\delta^7 \oplus \gamma^7\delta^{+\infty},$$

les différents indicateurs sont :

$$\begin{aligned} I'_{x_1}(u, y_o) &= \text{faux car } \Delta(x_{ed1}, x'_{s1}) = [0, 0], \\ I'_{x_2}(u, y_o) &= \text{faux car } \Delta(x_{ed2}, x'_{s2}) = [0, 0], \\ I'_{x_3}(u, y_o) &= \text{vrai car } \Delta(x_{ed1}, x'_{s1}) = [1, 1]. \end{aligned}$$

La ligne c_4 du tableau 6.5 indique que la dérive temporelle peut être sur la place p_4 . En effet, pour chaque événement des entrées u_1 et u_2 , le chemin $u_2 \rightsquigarrow x_3$ est le plus long parmi les deux chemins menant à x_3 depuis ces entrées, même sans la dérive temporelle. Ainsi, pour $t_1 = t_2 = 1$ (premières occurrences de u_1 et u_2), on a $a + b + t_1 < c + d + t_2 \Rightarrow 1 + 2 + 1 < 3 + 4 + 1 \Rightarrow 4 < 8$. La dérive temporelle sur p_4 accentue cette différence : $a + b + t_1 < c + d + \theta + t_2 \Rightarrow 1 + 2 + 1 < 3 + 4 + 1 + 1 \Rightarrow 4 < 9$.

Tableau 6.6 (lignes c_6-c_9) : x_3 et $\{x_1 \text{ ou } x_2\}$ sont estimés SAP

Dans ce tableau, les états estimés SAP sont soit le couple x_1, x_3 , soit le couple x_2, x_3 . La notation / indique quelle configuration est prise en compte dans la localisation. Par exemple dans la ligne c_7 , (p_1) est la localisation obtenue lorsque x_1 et x_3 sont estimés SAP, tandis que (p_2) est celle obtenue pour x_2, x_3 .

	$\Delta(x_{ed1}, x'_{s1}) / \Delta(x_{ed2}, x'_{s2})$	$\Delta(x_{ed3}, x'_{s3})$	Localisation
c_6	[0,0]	[0,0]	\emptyset
c_7	$[\theta, \theta]$	$[0,0]$ ou $[0, \theta]$ ou $[\theta, \theta]$	$(p_1) / (p_2)$
c_8	[0,0]	$[\theta, \theta]$	$\forall n \in \mathbb{Z}, t_1 = D_{u_1}(n), t_2 = D_{u_2}(n)$ $p_3 / (p_3)$ si $a + b + \theta + t_1 > c + d + t_2$ $(p_4) / p_4$ si $a + b + t_1 < c + d + \theta + t_2$
c_9	[0,0]	$[0, \theta]$	$\exists n \in \mathbb{Z}, t_1 = D_{u_1}(n), t_2 = D_{u_2}(n)$ $p_3 / (p_3)$ si $a + b + \theta + t_1 > c + d + t_2$ $(p_4) / p_4$ si $a + b + t_1 < c + d + \theta + t_2$

TABLE 6.6 – Lignes c_6-c_9 : x_3 et $\{x_1 \text{ ou } x_2\}$ estimés SAP

Dans la **ligne c_6** , les indicateurs renvoient faux donc il n'y a pas de dérive temporelle. Dans la **ligne c_7** , l'indicateur renvoie vrai pour x_1/x_2 avec l'intervalle $[\theta, \theta]$. Cela signifie que la dérive temporelle θ est située soit sur la place p_1 ou sur une place en amont (si c'est l'état x_1 qui est estimé SAP), soit sur p_2 ou en amont (si x_2 est estimé SAP, voir résultats de la structure tandem). Les valeurs de l'intervalle $\Delta(x_{ed3}, x'_{s3})$ de la transition x_3 peuvent varier, cela ne modifie pas la localisation de la dérive.

Les **lignes c_8 et c_9** reprennent les résultats des lignes c_4 et c_5 du tableau 6.5 mais sont moins précis car le nombre d'états estimés SAP est plus petit. Par exemple, pour la ligne c_8 , si x_1 est estimé SAP, la dérive est localisée sur p_3 . Elle ne peut pas être sur p_1 car $\Delta(x_{ed1}, x'_{s1}) = [0, 0]$. De plus, elle ne peut pas être sur le chemin $u_2 \rightsquigarrow x_3$ car $a + b + \theta + t_1 > c + d + t_2$ comme expliqué pour le tableau précédent. Si x_2 est estimé SAP, la localisation est moins précise et donne comme résultat p_3 ou une place en amont de p_3 . En effet, seule l'information du chemin le plus long est disponible (chemin $u_1 \rightsquigarrow x_3$ qui contient p_3).

Tableau 6.7 (lignes $c_{10}-c_{12}$) : seul x_3 est estimé SAP

Dans ce tableau où seul l'état x_3 est estimé SAP, les résultats des tableaux précédents sont repris et généralisés.

Puisque l'indicateur associé à x_3 renvoie faux pour la **ligne c_{10}** , il n'y a pas de dérive temporelle. Les résultats de localisation des **lignes c_{11} et c_{12}** reprennent les informations des lignes c_8 et c_9 du tableau précédent en ne conservant que les éléments entre parenthèses. A nouveau, la localisation est moins précise car un seul état est estimé SAP.

	$\Delta(x_{ed3}, x'_{s3})$	Localisation
c_{10}	$[0,0]$	\emptyset
c_{11}	$[\theta, \theta]$	$\forall n \in \mathbb{Z}, t_1 = D_{u_1}(n), t_2 = D_{u_2}(n)$ (p_3) si $a + b + \theta + t_1 > c + d + t_2$ (p_4) si $a + b + t_1 < c + d + \theta + t_2$
c_{12}	$[0, \theta]$	$\exists n \in \mathbb{Z}, t_1 = D_{u_1}(n), t_2 = D_{u_2}(n)$ (p_3) si $a + b + \theta + t_1 > c + d + t_2$ (p_4) si $a + b + t_1 < c + d + \theta + t_2$

TABLE 6.7 – Lignes c_{10} - c_{12} : x_3 estimé SAP

6.3.4 Boucle sur une même transition

La structure boucle est divisée en deux catégories : boucle sur une même transition et boucle entre deux transitions différentes. Commençons par étudier la boucle sur une même transition représentée par la figure 6.11 et dans laquelle il y a une transition d'entrée qui ne fait pas partie de la structure mais qui est nécessaire pour faire la localisation.

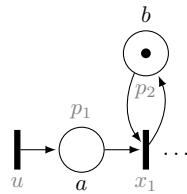


FIGURE 6.11 – Structure élémentaire de boucle sur une même transition

Le tableau 6.8 détaille l'ensemble des résultats possibles selon les valeurs d'intervales de l'indicateur associé à la transition x_1 .

	$\Delta(x_{ed1}, x'_{s1})$	Localisation
c_1	$[0,0]$	\emptyset
c_2	$[\theta, \theta]$	(p_1)
c_3	$[0, (n_K - 1) \times \theta]$	p_2

TABLE 6.8 – Résultat de localisation de la dérive temporelle pour une structure de boucle sur une même transition

Sur la **ligne c₁**, l'indicateur renvoie faux car $\Delta(x_{ed1}, x'_{s1}) = [0, 0]$. Cela signifie qu'il n'y a pas de dérive temporelle.

Pour les lignes suivantes, soient $u = \bigoplus_{k=0}^K \gamma^{n_k} \delta^{t_k}$ la forme canonique de l'entrée u et celle de la transition $x_1 = \bigoplus_{k=0}^K \gamma^{n_k^1} \delta^{t_k^1}$ à laquelle on ajoute le monôme ε afin de prendre en compte le jeton déjà présent dans la place p_2 à $t = 0$ (ce jeton est considéré présent depuis la date $-\infty$ soit depuis l'origine des temps). Ainsi :

$$\begin{aligned} x_1 &= \bigoplus_{k=0}^K \gamma^{n_k^1} \delta^{t_k^1} \oplus \varepsilon, \\ &= \bigoplus_{k=0}^K \gamma^{n_k^1} \delta^{t_k^1} \oplus \gamma^{+\infty} \delta^{-\infty}, \\ &= \bigoplus_{k=0}^K \gamma^{n_k^1} \delta^{t_k^1} \oplus \gamma^{n_{-1}^1} \delta^{t_{-1}^1}. \end{aligned}$$

Cet ajout, numéroté par l'indice -1 des exposants de γ et δ (n_{-1}^1 et t_{-1}^1) ne change pas la valeur de la série de x_1 (ε est neutre pour l'addition) et doit se faire sur la transition située juste après la place contenant le jeton présent à l'origine pour que le temps d'arrivée du jeton dans la place (temps considéré égal à $-\infty$) soit pris en compte dans les calculs des dates de tir de cette transition.

La **ligne c₂** indique $\Delta(x_{ed1}, x'_{s1}) = [\theta, \theta]$ ce qui signifie que la dérive temporelle est sur la place p_1 ou en amont.

Démonstration. L'obtention d'un intervalle avec des bornes identiques montre que l'effet de la dérive temporelle est toujours le même. Si cette dérive est située sur p_1 ou en amont, toutes les

dates de tir de x_1 sont :

$$t_k^1 = \max(t_k + a + \theta, t_{k-1}^1 + b) = t_k + a + \theta.$$

Chaque date numérotée k de tir de x_1 est retardée par la dérive θ pour chaque date numérotée k de tir de u , à laquelle on ajoute le temps de séjour a de la place p_1 . La dérive est tout de suite prise en compte et son effet n'évolue pas au cours du temps. Les seules places candidates qui permettent d'obtenir un tel résultat sont les places situées en amont et en dehors de la boucle, soit la place p_1 ou une place en amont. \square

Dans la **ligne c₃**, l'indicateur renvoie vrai car $\Delta(x_{ed1}, x_{s1}) = [0, (n_K - 1) \times \theta]$. Cela indique que la dérive temporelle est sur la place p_2 .

Démonstration. L'une des bornes de l'intervalle étant égale à 0, l'effet de la dérive temporelle n'apparaît pas immédiatement à cause des jetons déjà présents dans la structure. En effet, si la dérive temporelle est située sur p_2 , le jeton déjà présent dans cette place à $t = 0$ signifie que son temps de séjour n'est pas impacté par cette dérive. Il est donc directement disponible pour le premier tir de x_1 . La date du premier tir de x_1 dépend alors uniquement de la date du premier tir de u et la dérive n'est pas détectée :

$$t_0^1 = \max(t_0 + a, t_{-1}^1 + b) = \max(t_0 + a, -\infty + b) = t_0 + a.$$

L'analyse de l'autre borne (celle différente de 0) montre que la dérive est prise en compte et se retrouve dans les dates des tirs suivants de x_1 :

$$\begin{aligned} t_1^1 &= \max(t_1 + a, t_0^1 + b + \theta) = t_0^1 + b + \theta, \\ t_2^1 &= \max(t_2 + a, t_1^1 + b + \theta) = t_1^1 + b + \theta = (t_0^1 + b + \theta) + b + \theta, \\ t_3^1 &= \max(t_3 + a, t_2^1 + b + \theta) = t_2^1 + b + \theta = (t_0^1 + b + \theta + b + \theta) + b + \theta, \\ &\dots \\ t_k^1 &= \max(t_k + a, t_{k-1}^1 + b + ((n_K - 1) \times \theta)) = t_{k-1}^1 + b + ((n_K - 1) \times \theta). \end{aligned}$$

L'accumulation de la valeur de la dérive sur les dates de tir de x_1 dépend du nombre n_K de tirs de u (qui est identique au nombre de tirs de x_1) moins un. Ainsi, la seule place candidate permettant d'obtenir l'intervalle $[0, (n_K - 1) \times \theta]$ est la place p_2 , seule place de la boucle avec un jeton déjà présent à $t = 0$. \square

Exemple 6.20. Soit le GET de la figure 6.11 avec les durées $a = 1$ et $b = 4$ et soit une dérive temporelle $\theta = 2$ sur la place p_2 . Avec l'entrée suivante :

$$u_1 = \gamma^0 \delta^1 \oplus \gamma^1 \delta^2 \oplus \gamma^2 \delta^3 \oplus \gamma^3 \delta^4 \oplus \gamma^4 \delta^5 \oplus \gamma^5 \delta^6 \oplus \gamma^6 \delta^7 \oplus \gamma^7 \delta^{+\infty},$$

l'unique indicateur est :

$$I'_{x_1}(u, y_o) = \text{vrai car } \Delta(x_{ed1}, x_{s1}) = [0, 12].$$

Donc d'après le tableau 6.8, la ligne c_3 indique que la dérive temporelle est sur la place p_2 . Sa valeur est bien $\theta = 2$ car il y a eu 7 tirs de la transition u (= nombre de tirs de x_1) ce qui donne : $(n_K - 1) \times \theta = 12 \Rightarrow \theta = 12 \div (n_K - 1) = 12 \div (7 - 1) = 12 \div 6 = 2$.

6.3.5 Boucle entre deux transitions

La figure 6.12 représente la structure élémentaire d'une boucle entre deux transitions dans laquelle à nouveau une transition d'entrée permet d'obtenir les informations nécessaires pour la localisation.

Le tableau 6.9 regroupe les résultats de localisation selon la configuration pour laquelle x_1 et x_2 sont estimés SAP. La dernière ligne est séparée des autres car il s'agit d'un cas particulier de cette structure.

La **ligne c₁** de ce tableau montre qu'il n'y a pas de dérive temporelle.

Pour les autres lignes, soient $u = \bigoplus_{k=0}^K \gamma^{n_k} \delta^{t_k}$ la forme canonique de l'entrée u , celle de la transition $x_1 = \bigoplus_{k=0}^K \gamma^{n_k^1} \delta^{t_k^1}$ et celle de la transition x_2 à laquelle on ajoute le monôme ε afin de prendre en compte le jeton déjà présent dans la place p_3 à $t = 0$:

$$\begin{aligned} x_2 &= \bigoplus_{k=0}^K \gamma^{n_k^2} \delta^{t_k^2} \oplus \varepsilon, \\ &= \bigoplus_{k=0}^K \gamma^{n_k^2} \delta^{t_k^2} \oplus \gamma^{+\infty} \delta^{-\infty}, \\ &= \bigoplus_{k=0}^K \gamma^{n_k^2} \delta^{t_k^2} \oplus \gamma^{n_{-1}^2} \delta^{t_{-1}^2}. \end{aligned}$$

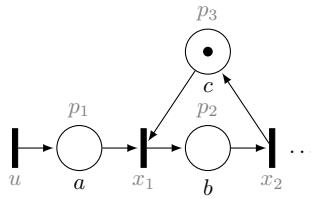


FIGURE 6.12 – Structure élémentaire de boucle entre deux transitions

	$\Delta(x_{ed1}, x'_{s1})$	$\Delta(x_{ed2}, x'_{s2})$	Localisation
c_1	$[0,0]$	$[0,0]$	\emptyset
c_2	$[\theta, \theta]$	$[\theta, \theta]$	(p_1)
c_3	$[0, (n_K - 1) \times \theta]$	$[0, (n_K - 1) \times \theta]$	p_3
c_4	$[0, (n_K - 1) \times \theta]$	$[\theta, n_K \times \theta]$	p_2
c_5	$[r_1, s_1]$	$[r_2, s_2]$	p_2 ou p_3

TABLE 6.9 – Résultat de localisation de la dérive temporelle pour une structure de boucle entre deux transitions

La **ligne c₂** indique que les intervalles associés aux transitions x_1 et x_2 sont égaux à $[\theta, \theta]$. Cela signifie que la dérive temporelle θ a lieu soit sur la place p_1 , soit en amont.

Démonstration. Comme pour la ligne c_2 du tableau 6.8, toutes les dates de tir de x_1 sont données par :

$$t_k^1 = \max(t_k + a + \theta, t_{k-1}^2 + c) = t_k + a + \theta.$$

L'effet de la dérive est toujours le même et montre qu'elle n'est pas localisée dans la boucle mais sur la place p_1 ou en amont. \square

Pour la **ligne c₃**, les indicateurs renvoient vrai et les deux intervalles obtenus sont identiques : $[0, (n_K - 1) \times \theta]$. La dérive temporelle θ est localisée sur p_3 , seule place de la boucle contenant un jeton à $t = 0$.

Démonstration. La preuve est identique à celle de la ligne c_3 du tableau 6.8. \square

Enfin pour la **ligne c₄**, les intervalles associés à x_1 et x_2 ne sont pas identiques et ont des bornes différentes. La dérive temporelle θ est localisée sur p_2 .

Démonstration. Comme pour la ligne précédente (ligne c_3), la présence de l'unique jeton de la boucle dans la place située juste avant la transition x_1 permet d'obtenir l'intervalle $[0, (n_K - 1) \times \theta]$ pour cette transition, peu importe où est située la dérive entre p_2 et p_3 . En revanche, si la dérive est située sur p_2 , la date du premier tir de x_2 est :

$$\begin{aligned} t_0^2 &= t_0^1 + b + \theta, \\ &= \max(t_0 + a, t_{-1}^2 + c) + b + \theta = (t_0 + a) + b + \theta, \end{aligned}$$

tandis que les dates suivantes sont :

$$\begin{aligned} t_1^2 &= t_1^1 + b + \theta \\ &= \max(t_1 + a, t_0^2 + c) + b + \theta = (t_0^2 + c) + b + \theta \\ &= (t_0 + a + b + \theta + c) + b + \theta, \\ t_2^2 &= t_2^1 + b + \theta \\ &= \max(t_2 + a, t_1^2 + c) + b + \theta = (t_1^2 + c) + b + \theta \\ &= ((t_0 + a + b + \theta + c) + b + \theta + c) + b + \theta, \\ &\dots \\ t_k^2 &= t_k^1 + b + \theta \\ &= \max(t_k + a, t_{k-1}^2 + c) + b + \theta = (t_{k-1}^2 + c) + b + \theta \\ &= (t_0 + a) + (n_K \times b) + ((n_K - 1) \times c) + (n_K \times \theta). \end{aligned}$$

La valeur de la dérive apparaît dès le premier tir de x_2 et est conservée dans la borne minimum de l'intervalle associé. Sa valeur est ajoutée à chaque nouveau tir donc la borne supérieure de l'intervalle est égale à $(n_K \times \theta)$. \square

Enfin, la **ligne c_5** est un cas particulier de cette localisation car elle correspond à l'hypothèse ² où $\nu_d \neq 0$ avec, pour $i \in \{1, 2\}$, $r_i \neq s_i$ et $r_i \neq 0$. Dans ce cas, la dérive est dans la structure boucle mais la localisation ne permet pas de différencier sur quelle place. Il s'agit d'une généralisation des lignes c_3 et c_4 .

Démonstration. Les intervalles obtenus pour x_1 et x_2 étant avec des bornes différentes, la dérive n'est pas située sur p_1 ou en amont (ligne c_2 du tableau). Elle est donc nécessairement située entre les places p_2 et p_3 . \square

Exemple 6.21. Soit le GET de la figure 6.12 avec les durées $a = 1, b = 2$ et $c = 3$ et soit une dérive temporelle $\theta = 1$ sur la place p_2 . Avec l'entrée suivante :

$$u_1 = \gamma^0 \delta^1 \oplus \gamma^1 \delta^2 \oplus \gamma^2 \delta^3 \oplus \gamma^3 \delta^4 \oplus \gamma^4 \delta^5 \oplus \gamma^5 \delta^6 \oplus \gamma^6 \delta^7 \oplus \gamma^7 \delta^{+\infty},$$

les différents indicateurs sont :

$$\begin{aligned} I'_{x_1}(u, y_o) &= \text{vrai} \quad \text{car} \quad \Delta(x_{ed1}, x'_{s1}) = [0, 6], \\ I'_{x_2}(u, y_o) &= \text{vrai} \quad \text{car} \quad \Delta(x_{ed2}, x'_{s2}) = [1, 7]. \end{aligned}$$

D'après le tableau 6.9, la ligne c_4 indique que la dérive temporelle $\theta = 1$ peut être sur la place p_2 .

6.3.6 Exemples d'associations de structures élémentaires

Cette section présente des exemples d'associations de structures élémentaires pour faire de la localisation dans des GET plus complexes. Chaque exemple est découpé en trois étapes : décomposer le GET global par les différentes structures élémentaires qui le composent grâce à la connaissance structurelle du système étudié, appliquer les résultats de localisation pour chaque structure et enfin regrouper ces informations pour fournir un ensemble de places candidates sous l'hypothèse d'au plus une seule dérive temporelle présente à tout instant dans le système.

Pour chaque GET complet, un tableau présente les résultats de localisation par structure élémentaire.

- Celui-ci liste les différentes structures élémentaires qui composent le GET principal ainsi que les transitions qui appartiennent à chaque structure et les places potentiellement concernées par la dérive (colonnes "Structure" et "Trans. et Pl." pour Transitions et Places). Certaines transitions peuvent appartenir à plusieurs structures (par exemple une transition impliquée dans une structure tandem et une structure boucle).
- Les intervalles issus de l'étape de détection et obtenus pour chaque état sont récapitulés dans la colonne "Intervalles".
- Chaque ligne du tableau fournit un ensemble de places candidates $\{p_i\}$ pour la localisation de la dérive par structure (dernière colonne "Localisation"). Si cet ensemble est vide, la ligne contient à nouveau le symbole \emptyset .

Une analyse de ces ensembles est ensuite réalisée pour fournir un unique ensemble de places candidates noté \mathcal{PC} pour la localisation générale. Pour les exemples traités dans cette section, sous l'hypothèse d'une seule dérive temporelle présente dans le système à tout instant, cet ensemble est égal à l'intersection des ensembles obtenus par structure.

Remarque. Il est important de noter que chaque structure étant associée à une ou plusieurs autres structures (un tandem suivi d'une synchronisation, une boucle sur un tandem...), l'intervalle obtenu pour un état dans les sections précédentes n'apparaît potentiellement plus sous la même forme que celui présenté dans les tableaux des résultats par structure. Par exemple, une transition impliquée dans un tandem et une boucle n'aura pas un intervalle associé avec des bornes identiques comme c'est le cas pour la structure tandem seule.

Remarque. Rappelons que les indicateurs de l'étape de détection ne fournissent des informations que sur les états du système. Toutes les analyses de localisation ne sont donc faites que sur les transitions internes (une synchronisation entre une entrée et une transition interne ne sera pas analysée, de même pour un parallélisme entre une sortie et une transition interne...).

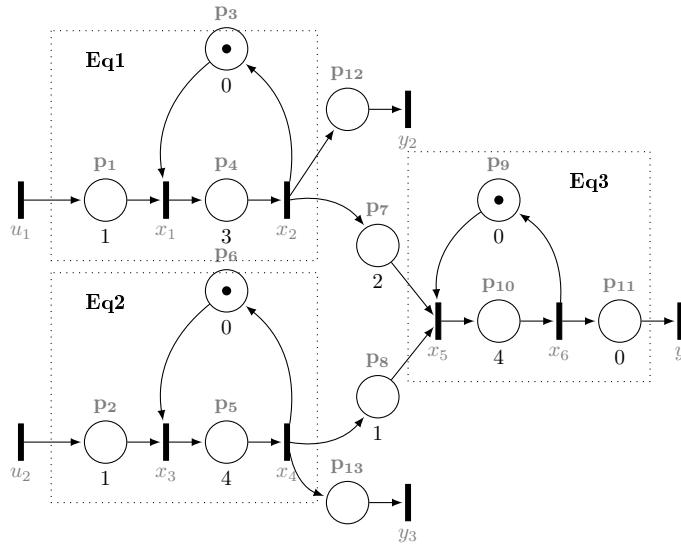


FIGURE 6.13 – GET avec 2 entrées et 3 sorties

Exemple 6.22. Reprenons l'exemple 6.4 dont le GET est repris dans la figure 6.13 ci-dessous sans représentation des entrées de perturbations.

Le tableau 6.10 récapitule les informations de localisation par structure de ce GET lorsqu'une dérive temporelle $\theta = 1$ est présente sur p_4 .

	Trans. et Pl.	Structure	Intervalles	Localisation
1	x_1, x_2 p_1, p_3, p_4	Boucle	$\Delta(x_{ed1}, x'_{s1}) = [1, 6]$ $\Delta(x_{ed2}, x'_{s2}) = [1, 7]$	$\{p_3, p_4\}$
2	x_3, x_4 p_2, p_5, p_6	Boucle	$\Delta(x_{ed3}, x'_{s3}) = [0, 0]$ $\Delta(x_{ed4}, x'_{s4}) = [0, 0]$	\emptyset
3	x_2, x_4, x_5, x_6 $p_1, p_3, p_4, p_7,$ $p_2, p_5, p_6, p_8,$ p_9, p_{10}	Synchronisation	$\Delta(x_{ed2}, x'_{s2}) = [1, 7]$ $\Delta(x_{ed4}, x'_{s4}) = [0, 0]$ $\Delta(x_{ed5}, x'_{s5}) = [1, 1]$ $\Delta(x_{ed6}, x'_{s6}) = [1, 1]$	$\{p_1, p_3, p_4\}$
4	x_5, x_6 $\forall p_i \text{ avec } i \in \{1, 10\}$	Boucle	$\Delta(x_{ed5}, x'_{s5}) = [1, 1]$ $\Delta(x_{ed6}, x'_{s6}) = [1, 1]$	$\{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8\}$

TABLE 6.10 – Localisation par structure du GET de la figure 6.13

La **ligne 1** du tableau correspond aux transitions x_1 et x_2 impliquées dans une structure élémentaire de boucle entre deux transitions. Les places considérées dans l'analyse de cette structure sont celles de la boucle plus celle située juste avant la boucle soit $\{p_1, p_3, p_4\}$. Les valeurs des intervalles obtenus par l'étape de détection ($\Delta(x_{ed1}, x'_{s1}) = [1, 6]$ et $\Delta(x_{ed2}, x'_{s2}) = [1, 7]$) correspondent à la ligne c_5 du tableau 6.9 ce qui donne une localisation de la dérive sur l'une des places de la boucle : p_3 ou p_4 .

Remarque. En fait, de manière plus précise, les transitions x_1 et x_2 sont impliquées dans deux structures tandem ($x_1 \rightsquigarrow x_2$ et $x_2 \rightsquigarrow x_1$) qui composent une structure boucle. D'ailleurs, les valeurs des intervalles obtenus par l'étape de détection pour ces transitions ont des bornes différentes ce qui est typique d'une structure boucle.

Le découpage et l'analyse de la **ligne 2** est similaire à la ligne c_1 avec les transitions x_3 et x_4 . En revanche, d'après les valeurs des intervalles, il n'y a pas de dérive temporelle dans cette partie du GET, donc pas de dérive sur les places p_2, p_5 ni p_6 . La **ligne 3** implique les transitions x_2, x_4, x_5 et x_6 qui forment une structure de synchronisation. Les places concernées par cette structure sont toutes celles situées en amont de la transition de synchronisation x_5 soit $\{p_1, p_3, p_4, p_7, p_2, p_5, p_6, p_8, p_9, p_{10}\}$. D'après les résultats de localisation pour une telle structure et

2. Pour les autres lignes de ce tableau, $\nu_d = 0$.

de telles valeurs d'intervalles (ligne c_2 du tableau 6.5), les places candidates pour la dérive temporelle sont $\{p_1, p_3, p_4\}$. Enfin, la **ligne 4** présente une autre structure boucle entre les transitions x_5 et x_6 qui impliquent les places de la boucle $\{p_9, p_{10}\}$ ainsi que toutes les places en amont $\{p_1, p_2, \dots, p_7, p_8\}$. Les intervalles associés aux transitions sont identiques ce qui correspond à la ligne c_2 du tableau 6.9. Ainsi, la dérive n'est pas localisée dans la boucle mais en amont de la boucle, c'est-à-dire sur l'une des places de l'ensemble : $\{p_1, p_2, \dots, p_7, p_8\}$.

Les différents éléments de la localisation doivent maintenant être analysés et regroupés afin de ne proposer qu'un ensemble cohérent de places candidates pour ce GET complet et ce premier jeu de valeurs d'intervalles obtenus par la détection. L'intersection de tous les ensembles de la colonne Localisation donne cet ensemble \mathcal{PC} de place candidate :

$$\begin{aligned}\mathcal{PC} &= \{p_3, p_4\} \cap \{p_1, p_3, p_4\} \cap \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8\}, \\ &= \{p_3, p_4\}.\end{aligned}$$

La dérive temporelle est donc localisée soit sur la place p_3 , soit sur la place p_4 .

Exemple 6.23. Le tableau 6.11 contient un deuxième ensemble de valeurs d'intervalles obtenus par l'étape de détection de dérive pour le GET de la figure 6.13. Cette fois-ci, la dérive $\theta = 1$ est présente sur p_2 .

	Trans. et Pl.	Structure	Intervalles	Localisation
1	x_1, x_2 p_1, p_3, p_4	Boucle	$\Delta(x_{ed1}, x'_{s1}) = [0, 0]$ $\Delta(x_{ed2}, x'_{s2}) = [0, 0]$	\emptyset
2	x_3, x_4 p_2, p_5, p_6	Boucle	$\Delta(x_{ed3}, x'_{s3}) = [1, 1]$ $\Delta(x_{ed4}, x'_{s4}) = [1, 1]$	$\{p_2\}$
3	x_2, x_4, x_5 p_1, p_3, p_4, p_7 p_2, p_5, p_6, p_8 p_9, p_{10}	Synchronisation	$\Delta(x_{ed2}, x'_{s2}) = [0, 0]$ $\Delta(x_{ed4}, x'_{s4}) = [1, 1]$ $\Delta(x_{ed5}, x'_{s5}) = [1, 1]$ $\Delta(x_{ed6}, x'_{s6}) = [1, 1]$	$\{p_2, p_5, p_6\}$
4	x_5, x_6 $\forall p_i \text{ avec } i \in \{1, 10\}$	Boucle	$\Delta(x_{ed5}, x'_{s5}) = [1, 1]$ $\Delta(x_{ed6}, x'_{s6}) = [1, 1]$	$\{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8\}$

TABLE 6.11 – Deuxième jeu d'intervalles pour la localisation par structure du GET de la figure 6.13

L'analyse est identique à celle de l'exemple précédent. Ainsi, d'après ce tableau, la dérive temporelle est localisée sur la place p_2 :

$$\begin{aligned}\mathcal{PC} &= \{p_2\} \cap \{p_2, p_5, p_6\} \cap \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8\}, \\ &= \{p_2\}.\end{aligned}$$

Exemple 6.24. Le tableau 6.12 récapitule les informations liées à un troisième ensemble de valeurs d'intervalles pour le GET de la figure 6.13 avec une dérive $\theta = 1$ sur p_{10} .

	Trans. et Pl.	Structure	Intervalles	Localisation
1	x_1, x_2 p_1, p_3, p_4	Boucle	$\Delta(x_{ed1}, x'_{s1}) = [0, 0]$ $\Delta(x_{ed2}, x'_{s2}) = [0, 0]$	\emptyset
2	x_3, x_4 p_2, p_5, p_6	Boucle	$\Delta(x_{ed3}, x'_{s3}) = [0, 0]$ $\Delta(x_{ed4}, x'_{s4}) = [0, 0]$	\emptyset
3	x_2, x_4, x_5 $p_1, p_3, p_4, p_7,$ $p_2, p_5, p_6, p_8,$ p_9, p_{10}	Synchronisation	$\Delta(x_{ed2}, x'_{s2}) = [0, 0]$ $\Delta(x_{ed4}, x'_{s4}) = [0, 0]$ $\Delta(x_{ed5}, x'_{s5}) = [0, 6]$ $\Delta(x_{ed6}, x'_{s6}) = [1, 7]$	$\{p_9, p_{10}\}$
4	x_5, x_6 $\forall p_i \text{ avec } i \in \{1, 10\}$	Boucle	$\Delta(x_{ed5}, x'_{s5}) = [0, 6]$ $\Delta(x_{ed6}, x'_{s6}) = [1, 7]$	$\{p_{10}\}$

TABLE 6.12 – Troisième jeu d'intervalles pour la localisation par structure du GET de la figure 6.13

Puisque les **lignes 1-2** du tableau ne contiennent que des intervalles égaux à $[0, 0]$, il n'y a pas de dérive temporelle dans les places $\{p_1, p_2, p_3, p_4, p_5, p_6\}$. Pour la **ligne 3**, le raisonnement est identique à celui de la ligne c_3 du tableau 6.10 (avec le premier ensemble de valeurs d'intervalles).

Cette fois-ci, les places concernées par la dérive et situées en amont de x_5 sont les places de la boucle entre x_5 et x_6 soit $\{p_9, p_{10}\}$. Enfin, les valeurs de la **ligne 4** font référence à la ligne c_4 du tableau 6.9 et permettent de définir une unique place candidate $\{p_{10}\}$.

L'analyse globale peut maintenant être faite en effectuant l'intersection des ensembles trouvés de places candidates :

$$\begin{aligned}\mathcal{PC} &= \{p_9, p_{10}\} \cap \{p_{10}\}, \\ &= \{p_{10}\}.\end{aligned}$$

La dérive temporelle est localisée sur la place p_{10} .

Exemple 6.25. Soit le GET de la figure 6.14 avec 2 entrées u_1, u_2 et 4 sorties y_1, y_2, y_3, y_4 .

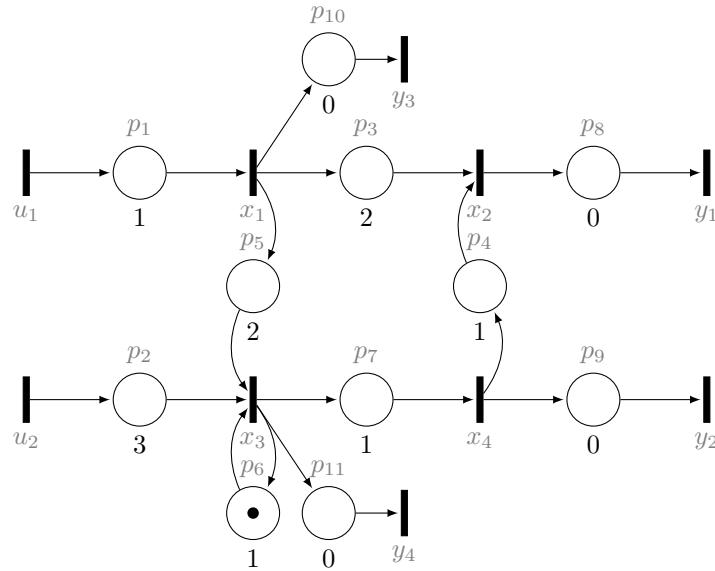


FIGURE 6.14 – GET avec 2 entrées et 4 sorties

Les différentes matrices du système sont :

$$A = \begin{pmatrix} \gamma^0 \delta^2 & \cdot & \cdot & \gamma^0 \delta^1 \\ \gamma^0 \delta^2 & \cdot & \gamma^1 \delta^1 & \cdot \\ \cdot & \cdot & \gamma^0 \delta^1 & \cdot \end{pmatrix}, B = \begin{pmatrix} \gamma^0 \delta^1 & \cdot \\ \cdot & \gamma^0 \delta^3 \end{pmatrix}, C = \begin{pmatrix} \cdot & \gamma^0 \delta^0 & \cdot & \gamma^0 \delta^0 \\ \gamma^0 \delta^0 & \cdot & \cdot & \gamma^0 \delta^0 \\ \cdot & \cdot & \gamma^0 \delta^0 & \cdot \\ \gamma^0 \delta^0 & \gamma^0 \delta^3 & \gamma^0 \delta^1 & \cdot \end{pmatrix},$$

tandis que la matrice L est :

$$L = \begin{pmatrix} \gamma^0 \delta^0 & \gamma^0 \delta^1 & \gamma^0 \delta^4 (\gamma^1 \delta^1)^* & \gamma^0 \delta^2 (\gamma^1 \delta^1)^* \\ \cdot & \cdot & \gamma^0 \delta^2 (\gamma^1 \delta^1)^* & \gamma^0 \delta^0 (\gamma^1 \delta^1)^* \\ \cdot & \gamma^0 \delta^0 & \gamma^0 \delta^3 (\gamma^1 \delta^1)^* & \gamma^0 \delta^1 (\gamma^1 \delta^1)^* \end{pmatrix}$$

Si les entrées suivantes sont appliquées au GET :

$$u_1 = u_2 = \gamma^0 \delta^1 \oplus \gamma^1 \delta^2 \oplus \gamma^2 \delta^3 \oplus \gamma^3 \delta^4 \oplus \gamma^4 \delta^5 \oplus \gamma^5 \delta^6 \oplus \gamma^6 \delta^7 \oplus \gamma^7 \delta^{+\infty}.$$

on obtient les états estimés SAP x_{ed} :

$$x_{ed} = \begin{pmatrix} \gamma^0 \delta^2 \oplus \gamma^1 \delta^3 \oplus \gamma^2 \delta^4 \oplus \gamma^3 \delta^5 \oplus \gamma^4 \delta^6 \oplus \gamma^5 \delta^7 \oplus \gamma^6 \delta^8 \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^6 \oplus \gamma^1 \delta^8 \oplus \gamma^2 \delta^{10} \oplus \gamma^3 \delta^{12} \oplus \gamma^4 \delta^{14} \oplus \gamma^5 \delta^{16} \oplus \gamma^6 \delta^{18} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^4 \oplus \gamma^1 \delta^6 \oplus \gamma^2 \delta^8 \oplus \gamma^3 \delta^{10} \oplus \gamma^4 \delta^{12} \oplus \gamma^5 \delta^{14} \oplus \gamma^6 \delta^{16} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^5 \oplus \gamma^1 \delta^7 \oplus \gamma^2 \delta^9 \oplus \gamma^3 \delta^{11} \oplus \gamma^4 \delta^{13} \oplus \gamma^5 \delta^{15} \oplus \gamma^6 \delta^{17} \oplus \gamma^7 \delta^{+\infty}, \end{pmatrix}$$

et les états x_s du modèle sans faute comparable aux états x_{ed} :

$$x_s = \begin{pmatrix} \gamma^0 \delta^2 \oplus \gamma^1 \delta^3 \oplus \gamma^2 \delta^4 \oplus \gamma^3 \delta^5 \oplus \gamma^4 \delta^6 \oplus \gamma^5 \delta^7 \oplus \gamma^6 \delta^8 \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^6 \oplus \gamma^1 \delta^7 \oplus \gamma^2 \delta^8 \oplus \gamma^3 \delta^9 \oplus \gamma^4 \delta^{10} \oplus \gamma^5 \delta^{11} \oplus \gamma^6 \delta^{12} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^4 \oplus \gamma^1 \delta^5 \oplus \gamma^2 \delta^6 \oplus \gamma^3 \delta^7 \oplus \gamma^4 \delta^8 \oplus \gamma^5 \delta^9 \oplus \gamma^6 \delta^{10} \oplus \gamma^7 \delta^{+\infty} \\ \gamma^0 \delta^5 \oplus \gamma^1 \delta^6 \oplus \gamma^2 \delta^7 \oplus \gamma^3 \delta^8 \oplus \gamma^4 \delta^9 \oplus \gamma^5 \delta^{10} \oplus \gamma^6 \delta^{11} \oplus \gamma^7 \delta^{+\infty}. \end{pmatrix}$$

Le tableau 6.13 récapitule les informations de localisation par structure de ce GET avec une dérive $\theta = 1$ sur p_6 .

	Trans. et Pl.	Structure	Intervalles	Localisation
1	x_1, x_2, x_4 $p_1, p_3, p_2, p_5, p_6, p_7, p_4$	Synchronisation	$\Delta(x_{ed1}, x_{s1}) = [0, 0]$ $\Delta(x_{ed2}, x_{s2}) = [0, 6]$ $\Delta(x_{ed4}, x_{s4}) = [0, 6]$	$\{p_1, p_2, p_5, p_6, p_7\}$
2	x_3 p_1, p_5, p_2, p_6	Boucle	$\Delta(x_{ed3}, x_{s3}) = [0, 6]$	$\{p_6\}$
3	x_3, x_1 p_1, p_5, p_6	Synchronisation	$\Delta(x_{ed3}, x_{s3}) = [0, 6]$ $\Delta(x_{ed1}, x_{s1}) = [0, 0]$	$\{p_6\}$
4	x_3 et x_4 p_1, p_5, p_2, p_6, p_7	Tandem	$\Delta(x_{ed3}, x_{s3}) = [0, 6]$ $\Delta(x_{ed4}, x_{s4}) = [0, 6]$	$\{p_1, p_2, p_5, p_6\}$

TABLE 6.13 – Résumé des différentes structures dans le GET général

La **ligne 1** du tableau correspond aux transitions x_1 , x_2 et x_4 qui sont en structure élémentaire de synchronisation. Les places concernées sont toutes celles situées en amont de la transition de synchronisation x_2 soit $\{p_1, p_3, p_2, p_5, p_6, p_7, p_4\}$. D'après les résultats de localisation pour une telle structure et de telles valeurs d'intervalles (ligne c_3 du tableau 6.5), les places candidates pour la dérive temporelle sont $\{p_2, p_3, p_4, p_5, p_6, p_7\}$. La **ligne 2** correspond à une boucle sur une même transition x_3 . D'après la ligne c_3 du tableau 6.8 de la localisation d'une telle structure, il n'y a qu'une seule place candidate, il s'agit de la place de la boucle p_6 . La **ligne 3** est une synchronisation sur la transition x_3 entre elle-même et la transition x_1 . Comme indiqué en remarque au début de cette section, la place p_2 n'est pas prise en compte dans cette structure car elle relie la transition x_3 à la transition d'entrée u_2 . D'après la ligne c_3 du tableau 6.5, la faute est localisée sur p_6 ou en amont ce qui donne une seule place candidate, la place p_6 . Enfin, la **ligne 4** récapitule les informations du tandem entre les transitions x_3 et x_4 . D'après la ligne c_2 du tableau 6.3, l'ensemble des places candidates sont toutes les places en amont de la transition x_3 .

Les différents éléments de la localisation sont regroupés afin de proposer un ensemble \mathcal{PC} de places candidates à la dérive temporelle pour ce GET. L'intersection de tous les ensembles de la colonne Localisation est la suivante :

$$\begin{aligned} \mathcal{PC} &= \{p_1, p_2, p_5, p_6, p_7\} \cap \{p_6\} \cap \{p_2, p_4, p_5, p_6, p_7\}, \\ &= \{p_6\}. \end{aligned}$$

La dérive temporelle est située sur la place p_6 .

Les différents exemples présentés dans cette section montrent qu'une décomposition d'un GET en plusieurs structures élémentaires, suivi d'une analyse des résultats de localisation par structure permet une localisation globale de la dérive temporelle dans le GET. Cependant, cette localisation globale demande pour l'instant beaucoup d'analyse et de vérification manuelle d'un expert. Le processus peut être amélioré et de nombreux éléments doivent encore être étudiés. Une première amélioration est l'automatisation de la décomposition du GET. Pour le moment la décomposition d'un GET complet se fait en observant la structure du système. Une analyse algorithmique de la matrice A , qui contient les liens entre les transitions internes, permettrait d'obtenir directement cette décomposition en parcourant ses différents monômes afin de définir quelles sont les structures élémentaires qui les unissent. De plus, comme nous avons pu le constater dans les exemples, certaines structures, telles que la boucle, sont plus englobantes que d'autres dans les valeurs des intervalles. Un classement des structures doit donc être réalisé pour affiner la décomposition via la matrice A . Il faut également s'assurer que l'analyse globale réalisée dans les exemples traités par une simple intersection des localisations par structure est toujours possible et correcte pour obtenir l'ensemble \mathcal{PC} de places candidates à la dérive temporelle dans le GET complet.

Remarque. Il est possible que les valeurs des intervalles ne correspondent pas à une localisation cohérente dans les résultats par structure élémentaire. Une explication possible pour ce type de résultats et que l'hypothèse d'une seule dérive temporelle présente dans le système à tout instant ne tient plus. Le système peut contenir plusieurs dérives temporelles et l'étape de localisation ne peut se faire telle que présentée dans ce chapitre.

6.4 Synthèse du chapitre

Ce chapitre présente la méthode de détection et de localisation de dérives temporelles sur des systèmes $(\max, +)$ -linéaires avec observateurs. Dans un premier temps, on introduit les systèmes $(\max, +)$ -linéaires avec perturbations et la mise en place d'un observateur sur ce type de système.

Dans un second temps, une méthode pour caractériser les dérives temporelles comme des perturbations est proposée afin de pouvoir exploiter l'observateur. Puis, à partir de cet observateur, on met en place un premier indicateur de dérives temporelles sur les états du GET.

Pour améliorer la détection des dérives temporelles sur les états, on met ensuite en place un filtrage des états pour ne prendre en compte que les perturbations. Grâce aux filtrages des états, l'indicateur est plus précis ce qui permet de définir des règles systématiques fondées sur les structures élémentaires du GET pour établir la localisation de la dérive temporelle sous l'hypothèse qu'elle est unique.

Chapitre 7

Détection de dérives temporelles dans des systèmes intervalles

Ce chapitre présente une méthode de détection de dérives temporelles dans les systèmes $(\max, +)$ -linéaires à temps intervalles. Il s'agit de systèmes dont les temps manipulés ne sont plus des temps fixes mais appartiennent à des intervalles. Ces systèmes permettent de représenter des durées d'opérations soumises à des variations (par exemple des temps de transport de pièces entre deux équipements, des temps de processus qui diffèrent selon l'équipement utilisé...) et dont ne connaît exactement qu'une borne inférieure et une borne supérieure. Ces systèmes peuvent être représentés mathématiquement par des dioïdes intervalles. Les GET des chapitres 5 et 6 peuvent alors être vus comme un cas particulier des GET intervalles avec des intervalles dits "dégénérés", c'est-à-dire avec des bornes inférieures et supérieures identiques.

Ce chapitre propose une approche pour la détection de dérives temporelles dans ces GET intervalles. La définition d'une dérive temporelle détectable est ici renforcée par rapport à la définition du chapitre 4 utilisée dans le chapitre 6, mais le terme dérive temporelle est toujours utilisé pour désigner une *dérive temporelle détectable*. Les hypothèses de la détection sont les suivantes : il y a plusieurs dérives temporelles possibles au même instant dans le GET (hypothèse identique à celle de la partie détection du chapitre 6), les dérives temporelles peuvent être des avances ou des retards (contrairement au chapitre précédent qui ne traite que des retards), les entrées et sorties sont mesurées et connues (hypothèse identique à la partie détection du chapitre précédent), un indicateur booléen est proposé pour chaque sortie observée y_o en fonction de la connaissance du système (matrice de transfert H) et de l'entrée u (pas d'utilisation d'observateur $(\max, +)$ pour estimer les états). Cet indicateur utilise les résultats de la comparaison entre la sortie observée et la sortie du modèle sans faute y_s , comparaison réalisée à partir de la résiduation. Il utilise des ensembles des sorties acceptables pour un système donné et est levé si une dérive temporelle est détectée.

7.1 Théorie des dioïdes intervalles

Les dioïdes intervalles sont une extension des dioïdes présentés dans le chapitre 5. Ils représentent le fait qu'un élément du dioïde n'est plus un scalaire mais un intervalle muni de deux bornes : une borne minimale et une borne maximale. Ces deux bornes peuvent être identiques, on parle alors d'intervalle dit *dégénéré* (cela revient alors à manipuler les dioïdes du chapitre 5).

Cette section présente les différentes définitions nécessaires aux dioïdes intervalles d'après les documents [Lho03], [LS01] et [BHRM13]. Ces éléments sont utilisés dans la section suivante pour définir les systèmes $(\max, +)$ -linéaires à temps intervalles et les GET intervalles. Tout comme dans les chapitres précédents, le dioïde principalement utilisé est le dioïde intervalle issu de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ noté : $I(\mathcal{M}_{in}^{ax}[\gamma, \delta])$.

Définition 7.1 (Intervalle dans un dioïde). Un intervalle fermé¹ dans un dioïde \mathcal{D} s'écrit :

$$\mathbf{x} = [\underline{x}, \bar{x}] \quad \text{avec } \underline{x}, \bar{x} \in \mathcal{D}$$

1. L'utilisation des intervalles fermés est utilisée pour avoir une analyse usuelle des intervalles. De plus, du point de vue industriel, cela garantit qu'une opération se fait bien dans l'intervalle de temps donné.

et satisfait :

$$\mathbf{x} = \{x \in \mathcal{D} \mid \underline{x} \preceq x \preceq \bar{x}\}.$$

Remarque. Dans cette définition et par la suite, la notation en gras est utilisée pour désigner un intervalle.

Remarque. Un intervalle *dégénéré* représente un intervalle pour lequel la borne inférieure est la même que la borne supérieure, soit $\mathbf{x} = [\underline{x}, \bar{x}]$ avec $\underline{x} = \bar{x}$. Un tel intervalle est noté plus simplement x (sans la mise en forme en gras).

Définition 7.2 (Ensemble des intervalles d'un dioïde). L'ensemble des intervalles fermés d'un dioïde \mathcal{D} est noté $I(\mathcal{D})$. Soient $\mathbf{x} = [\underline{x}, \bar{x}]$ et $\mathbf{y} = [\underline{y}, \bar{y}]$ deux intervalles appartenant à $I(\mathcal{D})$, les opérateurs internes de cet ensemble sont :

$$\begin{aligned} \mathbf{x} \overline{\oplus} \mathbf{y} &= [\underline{x} \oplus \underline{y}, \bar{x} \oplus \bar{y}], \\ \mathbf{x} \overline{\otimes} \mathbf{y} &= [\underline{x} \otimes \underline{y}, \bar{x} \otimes \bar{y}]. \end{aligned} \quad (7.1)$$

Les éléments neutres sont $\mathbf{e} = [\varepsilon, \varepsilon]$ pour la somme $\overline{\oplus}$ et $\mathbf{e} = [e, e]$ pour le produit $\overline{\otimes}$.

Théorème 7.1 ([Lho03]). *L'ensemble des intervalles fermés $I(\mathcal{D})$ muni des opérateurs $\overline{\oplus}$ et $\overline{\otimes}$ est un dioïde.*

Définition 7.3 (Relation d'ordre). L'opérateur $\overline{\oplus}$ engendre une relation d'ordre canonique \preceq dans $I(\mathcal{D})$ telle que :

$$\forall \mathbf{a}, \mathbf{b} \in I(\mathcal{D}), \quad \mathbf{a} \preceq \mathbf{b} \Leftrightarrow \underline{a} \preceq \underline{b} \text{ et } \bar{a} \preceq \bar{b}.$$

Théorème 7.2 ([LS01],[Lho03]). *Pour tout $\mathbf{x}, \mathbf{y} \in I(\mathcal{D})$ l'intervalle $\mathbf{x} \overline{\oplus} \mathbf{y}$ est le plus petit intervalle de $I(\mathcal{D})$ contenant l'ensemble $\{x \oplus y \mid x \in \mathbf{x}, y \in \mathbf{y}\}$. De même, l'intervalle $\mathbf{x} \overline{\otimes} \mathbf{y}$ est le plus petit intervalle de $I(\mathcal{D})$ contenant l'ensemble $\{x \otimes y \mid x \in \mathbf{x}, y \in \mathbf{y}\}$.*

Définition 7.4. Soit Π une application isotone définie sur un dioïde \mathcal{D} , l'application Π s'applique sur le dioïde $I(\mathcal{D})$ telle que $\Pi(\mathbf{x}) = \{\Pi(x) \mid x \in \mathbf{x}\} = [\Pi(\underline{x}), \Pi(\bar{x})]$.

Définition 7.5. Soient \mathcal{D} un dioïde complet et un sous-ensemble $\mathcal{C} \subseteq I(\mathcal{D})$. La définition de la somme infinie est la suivante :

$$\overline{\bigoplus}_{\mathbf{x} \in \mathcal{C}} \mathbf{x} = [\bigoplus_{\mathbf{x} \in \mathcal{C}} \underline{x}, \bigoplus_{\mathbf{x} \in \mathcal{C}} \bar{x}].$$

Définition 7.6. Si \mathcal{D} est un dioïde complet alors $I(\mathcal{D})$ est un dioïde complet.

Définition 7.7. L'opérateur étoile de Kleene (définition 5.17) défini sur $I(\mathcal{D})$ est tel que :

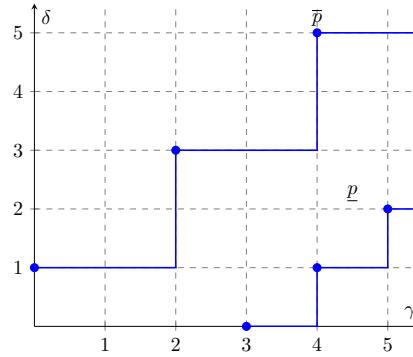
$$\forall \mathbf{a} \in I(\mathcal{D}), \quad \mathbf{a}^* = [\underline{a}^*, \bar{a}^*].$$

En s'appuyant sur les définitions et résultats précédents, le dioïde intervalle $I(\mathcal{M}_{in}^{ax}[\gamma, \delta])$ est présenté. Il permet de représenter les systèmes (max, +)-linéaires intervalles et les GET intervalles correspondants.

Définition 7.8 (Dioïde intervalle $I(\mathcal{M}_{in}^{ax}[\gamma, \delta])$). Le dioïde intervalle $I(\mathcal{M}_{in}^{ax}[\gamma, \delta])$ est formé de l'ensemble des intervalles fermés \mathbf{x} sur $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ muni des opérateurs $\overline{\otimes}$ et $\overline{\oplus}$. D'après la définition 7.6, le dioïde $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ étant complet, le dioïde $I(\mathcal{M}_{in}^{ax}[\gamma, \delta])$ est complet.

Un élément de $I(\mathcal{M}_{in}^{ax}[\gamma, \delta])$ correspond à l'ensemble des séries comprises entre la série de la borne minimale et la série de la borne maximale de l'intervalle. Rappelons qu'une série de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ peut être représentée graphiquement par l'association de ses fonctions dateur et compteur (voir figure 5.9 du chapitre 5). C'est cette représentation qui est choisie dans ce chapitre pour le tracé des bornes minimale et maximale d'un intervalle de $I(\mathcal{M}_{in}^{ax}[\gamma, \delta])$.

Exemple 7.1. Soit la série intervalle $\mathbf{p} = [p, \bar{p}] = [\gamma^3 \delta^0 \oplus \gamma^4 \delta^1 \oplus \gamma^5 \delta^2, \gamma^0 \delta^1 \oplus \gamma^2 \delta^3 \oplus \gamma^4 \delta^5] \in I(\mathcal{M}_{in}^{ax}[\gamma, \delta])$ représentée graphiquement sur la figure 7.1. Les points de coordonnées (0,1), (2,3) et (4,5) correspondent à la borne maximale \bar{p} de l'intervalle \mathbf{p} . Les points de coordonnées (3,0), (4,1) et (5,2) correspondent à sa borne minimale \underline{p} .

FIGURE 7.1 – Représentation de l'intervalle $\mathbf{p} = [\underline{p}, \bar{p}] \in I(\mathcal{M}_{in}^{ax}[\gamma, \delta])$

7.2 Modélisation des systèmes $(\max, +)$ -linaires à temps intervalles

Les systèmes $(\max, +)$ -linaires à temps intervalles sont des systèmes dont les temps ne sont plus des temps fixes mais appartiennent à des intervalles. La représentation d'état d'un tel système est identique à celle de l'équation (5.23) du chapitre 5 mais définie dans le dioïde intervalle $I(\mathcal{M}_{in}^{ax}[\gamma, \delta])$ telle que :

$$\begin{cases} \mathbf{x} = \mathbf{Ax} \mathbin{\overline{\oplus}} \mathbf{Bu}, \\ \mathbf{y} = \mathbf{Cx}, \end{cases}$$

avec $\mathbf{u} \in I(\mathcal{M}_{in}^{ax}[\gamma, \delta])^{p \times 1}$ le vecteur d'entrée (p entrées), $\mathbf{x} \in I(\mathcal{M}_{in}^{ax}[\gamma, \delta])^{n \times 1}$ le vecteur d'état, $\mathbf{y} \in I(\mathcal{M}_{in}^{ax}[\gamma, \delta])^{q \times 1}$ le vecteur de sortie (q sorties), $\mathbf{A} \in I(\mathcal{M}_{in}^{ax}[\gamma, \delta])^{n \times n}$ la matrice de connexion entre les états, $\mathbf{B} \in I(\mathcal{M}_{in}^{ax}[\gamma, \delta])^{n \times p}$ la matrice de connexion entre les états et les entrées, et $\mathbf{C} \in I(\mathcal{M}_{in}^{ax}[\gamma, \delta])^{q \times n}$ la matrice de connexion entre les états et les sorties.

La fonction de transfert permettant de représenter l'influence des entrées sur les sorties du système peut s'exprimer par la matrice $\mathbf{H} \in I(\mathcal{M}_{in}^{ax}[\gamma, \delta])^{q \times p}$ telle que :

$$\mathbf{H} = \mathbf{CA}^* \mathbf{B}.$$

Cette matrice est calculée par la transformation de la représentation d'état grâce au théorème de l'étoile de Kleene (théorème 5.2 et définition 7.7) appliqué au dioïde intervalle et permettant d'obtenir cette relation entrée/sortie :

$$\mathbf{y} = \mathbf{CA}^* \mathbf{Bu}.$$

Exemple 7.2. Soit le GET représenté par la figure 7.2. Les matrices de sa représentation d'état sont les suivantes :

$$\mathbf{A} = \begin{pmatrix} [\gamma^1 \delta^0, \gamma^1 \delta^0] & & & & & & \\ [\gamma^0 \delta^2, \gamma^0 \delta^5] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & [\gamma^0 \delta^1, \gamma^0 \delta^3] & [\gamma^1 \delta^0, \gamma^1 \delta^0] & \cdot & \cdot & \cdot & \cdot \\ \cdot & [\gamma^0 \delta^1, \gamma^0 \delta^3] & [\gamma^0 \delta^2, \gamma^0 \delta^5] & [\gamma^1 \delta^0, \gamma^1 \delta^0] & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & [\gamma^0 \delta^1, \gamma^0 \delta^2] & [\gamma^1 \delta^0, \gamma^1 \delta^0] & \cdot \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} [\gamma^0 \delta^0, \gamma^0 \delta^0] & & & & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \\ \cdot & [\gamma^0 \delta^0, \gamma^0 \delta^0] & & & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & [\gamma^0 \delta^0, \gamma^0 \delta^0] \end{pmatrix}.$$

Tout comme pour les matrices de la représentation d'état d'un système classique dans le dioïde $\mathcal{M}_{in}^{ax}[\gamma, \delta]$, les exposants de γ représentent les décalages événementiels du système tandis que les

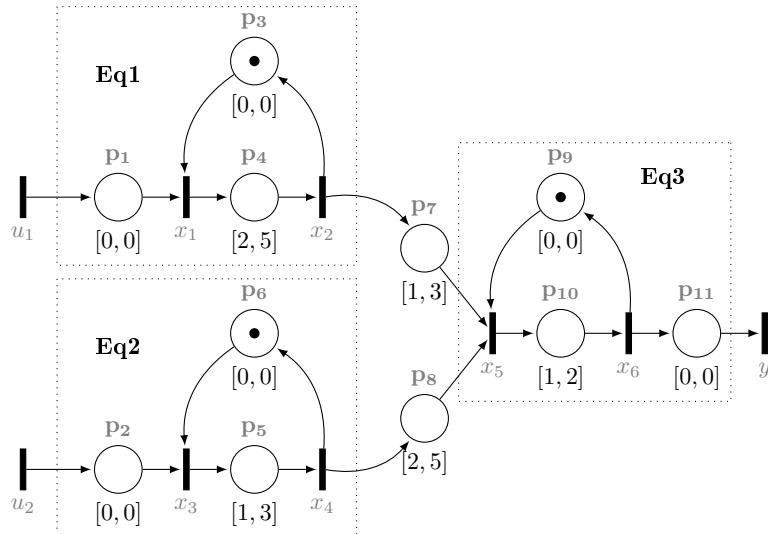


FIGURE 7.2 – Représentation d'un GET intervalle à 2 entrées et 1 sortie

exposants de δ représentent les décalages temporels. Plus précisément pour le temps, chaque borne inférieure des monômes, représentant chaque place du GET, contient la borne inférieure de l'intervalle temporel associé à la place. Cela correspond au temps de séjour minimal obligatoire que les jetons doivent subir en arrivant dans la place. Les bornes supérieures contiennent les bornes supérieures des intervalles temporels. Cela correspond au temps de séjour maximal possible pour cette place (sans prendre en compte les temps supplémentaires dus à d'éventuelles synchronisations). Ainsi, par exemple dans la matrice A , l'élément $A(1, 0) = [\gamma^0 \delta^2, \gamma^0 \delta^5]$ correspond à la place située entre la transition x_1 et la transition x_2 . L'intervalle temporel de cette place est $[2, 5]$. La valeur minimale 2 se retrouve dans l'exposant de δ de la borne minimale du monôme $\gamma^0 \delta^2$, la valeur maximale 5 se retrouve dans l'exposant de δ de la borne maximale du monôme $\gamma^0 \delta^5$.

La matrice de transfert du système est :

$$\mathbf{H} = \mathbf{C} \mathbf{A}^* \mathbf{B} = ([\gamma^0 \delta^4 (\gamma^1 \delta^2)^*, \gamma^0 \delta^{10} (\gamma^1 \delta^5)^*] \quad [\gamma^0 \delta^4 (\gamma^1 \delta^1)^*, \gamma^0 \delta^{10} (\gamma^1 \delta^3)^*]).$$

Si ce GET est soumis à l'intervalle dégénéré suivant comme les entrées :

$$\begin{aligned} \mathbf{u} &= [\underline{u}, \bar{u}] = \mathbf{u}_1 = \mathbf{u}_2 \text{ avec :} \\ \underline{u} &= \bar{u} = \gamma^0 \delta^1 \oplus \gamma^1 \delta^2 \oplus \gamma^2 \delta^3 \oplus \gamma^3 \delta^4 \oplus \gamma^4 \delta^5 \oplus \gamma^5 \delta^6 \oplus \gamma^6 \delta^7 \oplus \gamma^7 \delta^{+\infty}. \end{aligned}$$

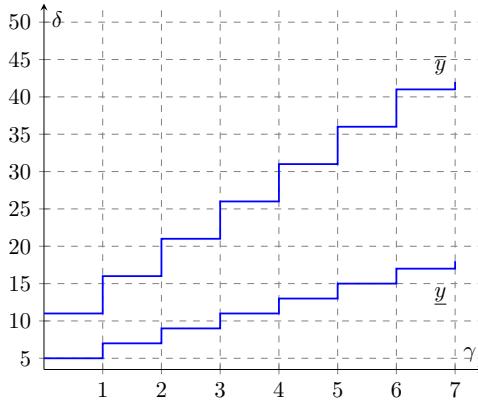
la sortie est alors (voir figure 7.3) :

$$\begin{aligned} \mathbf{y} &= [\underline{y}, \bar{y}] \\ &= [\gamma^0 \delta^5 \oplus \gamma^1 \delta^7 \oplus \gamma^2 \delta^9 \oplus \gamma^3 \delta^{11} \oplus \gamma^4 \delta^{13} \oplus \gamma^5 \delta^{15} \oplus \gamma^6 \delta^{17} \oplus \gamma^7 \delta^{+\infty}, \\ &\quad \gamma^0 \delta^{11} \oplus \gamma^1 \delta^{16} \oplus \gamma^2 \delta^{21} \oplus \gamma^3 \delta^{26} \oplus \gamma^4 \delta^{31} \oplus \gamma^5 \delta^{36} \oplus \gamma^6 \delta^{41} \oplus \gamma^7 \delta^{+\infty}]. \end{aligned}$$

La figure 7.3 représente les bornes $\underline{y} \preceq \bar{y}$ de la sortie \mathbf{y} . La borne inférieure \underline{y} est plus rapide que la borne supérieure \bar{y} car elle est celle qui prend en compte toutes les bornes inférieures des intervalles temporels situés sur les places du GET. La borne supérieure \bar{y} est quant à elle plus lente que la borne inférieure car elle prend en compte tous les temps maximaux des intervalles situés sur les places du GET.

7.3 Etude des sorties d'un système $(\max, +)$ -linéaire à temps intervalle

Cette section présente les définitions mises en place pour calculer la distance temporelle entre les bornes de la sortie $\mathbf{y} = [\underline{y}, \bar{y}]$ d'un système $(\max, +)$ -linéaire à temps intervalles. À partir de ces définitions, nous proposons un ensemble de sorties acceptables pour le système connaissant son entrée \mathbf{u} et sa fonction de transfert \mathbf{H} . Les conditions de ces ensembles sont utilisées dans les indicateurs de dérives temporelles sur les systèmes à temps intervalles dans la section suivante.

FIGURE 7.3 – Représentation de la sortie $\mathbf{y} = [\underline{y}, \bar{y}]$ du GET

7.3.1 Mesure de la distance temporelle du système

Dans un premier temps, la distance temporelle entre les bornes d'une sortie $\mathbf{y} = [\underline{y}, \bar{y}]$ d'un système (max, +)-linéaire à temps intervalles est calculée pour une entrée \mathbf{u} et une fonction de transfert \mathbf{H} données. Graphiquement, il s'agit de tous les écarts verticaux entre la borne minimale \underline{y} et la borne maximale \bar{y} , eux-mêmes bornés par l'intervalle de décalages temporels donné dans la définition 5.37 du chapitre 5.

Définition 7.9 (Distance temporelle d'un système). Soit la sortie intervalle $\mathbf{y} = \mathbf{H}\mathbf{u} = [\underline{y}, \bar{y}]$ avec $\mathbf{u} \in I(\mathcal{M}_{in}^{ax}[\gamma, \delta]^{p \times 1})$ et $\mathbf{H} \in I(\mathcal{M}_{in}^{ax}[\gamma, \delta]^{q \times p})$. La distance temporelle du système avec \bar{y} comme référence² est bornée par l'intervalle :

$$\Delta_{\bar{y}} = [\mathcal{D}_{\bar{y} \neq \bar{y}}(0); -\mathcal{D}_{\bar{y} \neq \bar{y}}(0)] = [\underline{\Delta}_{\bar{y}}, \bar{\Delta}_{\bar{y}}].$$

La distance temporelle du système avec \underline{y} comme référence est bornée par l'intervalle :

$$\Delta_{\underline{y}} = [\mathcal{D}_{\underline{y} \neq \underline{y}}(0); -\mathcal{D}_{\underline{y} \neq \underline{y}}(0)] = [\underline{\Delta}_{\underline{y}}, \bar{\Delta}_{\underline{y}}].$$

Remarque. La définition 7.9 est une distance donnée selon deux références différentes : \bar{y} et \underline{y} . La borne inférieure selon \bar{y} est l'opposé de la borne supérieure selon \underline{y} : $\underline{\Delta}_{\bar{y}} = -\bar{\Delta}_{\underline{y}}$; la borne supérieure selon \bar{y} est l'opposé de la borne inférieure selon \underline{y} : $\bar{\Delta}_{\bar{y}} = -\underline{\Delta}_{\underline{y}}$. Donc le calcul de l'une des deux distances suffit à obtenir l'autre distance.

Exemple 7.3. Soit le GET intervalle représenté par la figure 7.4.

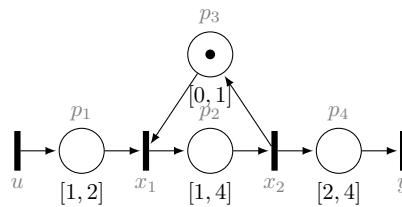


FIGURE 7.4 – Un GET intervalle avec 1 entrée et 1 sortie

Sa représentation d'état est :

$$\begin{cases} \mathbf{x} = \begin{pmatrix} [\varepsilon, \varepsilon] & [\gamma^1 \delta^0, \gamma^1 \delta^1] \\ [\gamma^0 \delta^1, \gamma^0 \delta^4] & [\varepsilon, \varepsilon] \end{pmatrix} \mathbf{x} \oplus \begin{pmatrix} [\gamma^0 \delta^1, \gamma^0 \delta^2] \\ [\varepsilon, \varepsilon] \end{pmatrix} \mathbf{u}, \\ \mathbf{y} = \begin{pmatrix} [\varepsilon, \varepsilon] & [\gamma^0 \delta^2, \gamma^0 \delta^4] \end{pmatrix} \mathbf{x}. \end{cases}$$

Ce qui donne comme fonction de transfert :

$$\mathbf{H} = ([\gamma^0 \delta^4 (\gamma^1 \delta^1)^*, \gamma^0 \delta^{10} (\gamma^1 \delta^5)^*]).$$

2. Élément à partir duquel le décalage est mesuré.

Si ce GET est soumis à l'intervalle dégénéré suivant comme entrée :

$$\begin{aligned}\mathbf{u} &= [\underline{u}, \bar{u}] \quad \text{avec :} \\ \underline{u} &= \bar{u} = \gamma^0 \delta^1 \oplus \gamma^1 \delta^2 \oplus \gamma^2 \delta^3 \oplus \gamma^3 \delta^4 \oplus \gamma^4 \delta^5 \oplus \gamma^5 \delta^6 \oplus \gamma^6 \delta^7 \oplus \gamma^7 \delta^{+\infty},\end{aligned}$$

la sortie, illustrée sur la figure 7.5a est alors :

$$\begin{aligned}\mathbf{y} &= [\underline{y}, \bar{y}] \\ &= [\gamma^0 \delta^5 \oplus \gamma^1 \delta^6 \oplus \gamma^2 \delta^7 \oplus \gamma^3 \delta^8 \oplus \gamma^4 \delta^9 \oplus \gamma^5 \delta^{10} \oplus \gamma^6 \delta^{11} \oplus \gamma^7 \delta^{+\infty}, \\ &\quad \gamma^0 \delta^{11} \oplus \gamma^1 \delta^{16} \oplus \gamma^2 \delta^{21} \oplus \gamma^3 \delta^{26} \oplus \gamma^4 \delta^{31} \oplus \gamma^5 \delta^{36} \oplus \gamma^6 \delta^{41} \oplus \gamma^7 \delta^{+\infty}].\end{aligned}$$

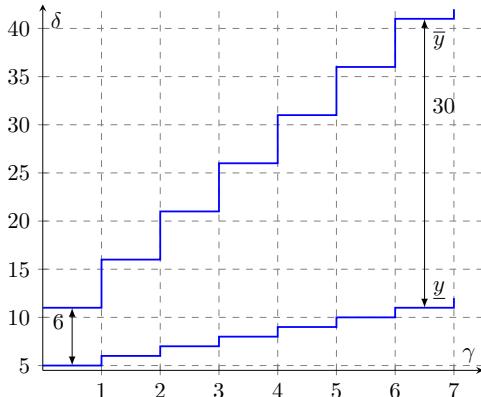
Les calculs de $\underline{y} \neq \bar{y}$ et de $\bar{y} \neq \underline{y}$ sont représentés figure 7.5b et donnent :

$$\begin{aligned}\underline{y} \neq \bar{y} &= \gamma^0 \delta^{-30} \oplus \gamma^1 \delta^{-25} \oplus \gamma^2 \delta^{-20} \oplus \gamma^3 \delta^{-15} \oplus \gamma^4 \delta^{-10} \oplus \gamma^5 \delta^{-5} \oplus \gamma^6 \delta^0 \oplus \gamma^7 \delta^{+\infty}, \\ \bar{y} \neq \underline{y} &= \gamma^0 \delta^6 \oplus \gamma^1 \delta^{11} \oplus \gamma^2 \delta^{16} \oplus \gamma^3 \delta^{21} \oplus \gamma^4 \delta^{26} \oplus \gamma^5 \delta^{31} \oplus \gamma^6 \delta^{36} \oplus \gamma^7 \delta^{+\infty}.\end{aligned}$$

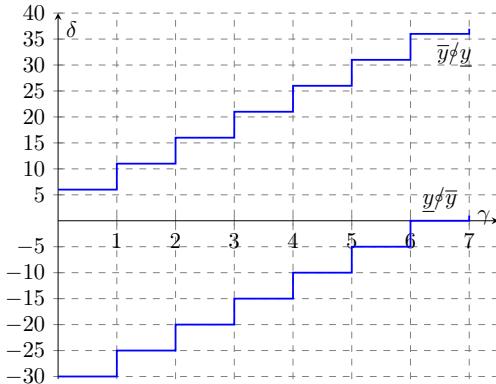
Ainsi, les distances temporelles du système sont :

$$\Delta_{\bar{y}} = [-30, -6], \quad \Delta_{\underline{y}} = [6, 30].$$

Les valeurs absolues des bornes sont identiques à un échange borne inférieure-borne supérieure près, ce qui est cohérent car elles représentent des écarts similaires. Il n'y a que la référence du calcul qui change. Avec \bar{y} comme référence, les bornes de $\Delta_{\bar{y}}$ sont négatives ($\bar{y} \succeq \underline{y}$), il s'agit bien de l'écart depuis la série \bar{y} vers la série \underline{y} donc une mesure faite de haut en bas. À l'inverse, avec \underline{y} comme référence, les bornes de $\Delta_{\underline{y}}$ sont positives ($\underline{y} \preceq \bar{y}$) car il s'agit de l'écart depuis \underline{y} vers \bar{y} (mesure de bas en haut).



(a) Représentation de \underline{y} et \bar{y}



(b) Représentation de $\underline{y} \neq \bar{y}$ et $\bar{y} \neq \underline{y}$

FIGURE 7.5 – Distance temporelle d'un système

7.3.2 Sorties acceptables d'un système

Après avoir défini la distance temporelle d'un système, c'est-à-dire l'intervalle contenant tous les décalages temporels possibles entre les bornes minimale et maximale de la sortie, l'ensemble des sorties acceptables permettant de caractériser le système est défini.

Définition 7.10 (Ensemble des sorties acceptables d'un système). L'ensemble des sorties acceptables d'un système est noté \mathcal{Y}_{acc} et regroupe toutes les sorties $y \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$ qui respectent la condition suivante :

$$\underline{y}_s \preceq y \preceq \bar{y}_s.$$

où $\mathbf{y}_s = [\underline{y}_s, \bar{y}_s] = \mathbf{H}\mathbf{u}$ avec $\mathbf{H} \in I(\mathcal{M}_{in}^{ax}[\gamma, \delta]^{q \times p})$ la fonction de transfert du système et $\mathbf{u} \in I(\mathcal{M}_{in}^{ax}[\gamma, \delta]^{p \times 1})$ une entrée connue.

Proposition 7.3. Une sortie $y \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$ est une sortie acceptable, et donc appartient à l'ensemble \mathcal{Y}_{acc} , si et seulement si les deux conditions suivantes sont vraies :

$$-\mathcal{D}_{\bar{y}_s \neq y}(0) \in [\Delta_{\bar{y}_s}, 0], \quad (7.2)$$

$$\mathcal{D}_{y \neq \underline{y}_s}(0) \in [0, \Delta_{\underline{y}_s}]. \quad (7.3)$$

Démonstration. Rappelons la définition 5.37 de l'intervalle de décalages temporels entre deux séries a et b de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ $\Delta(a, b) = [\mathcal{D}_{b \neq a}(0); -\mathcal{D}_{a \neq b}(0)]$ avec a comme série de référence; ainsi que la définition 7.9 de la distance temporelle du système entre les bornes \underline{y} et \bar{y} de sa sortie. L'équation (7.2) utilise cette distance temporelle avec \bar{y} comme référence tandis que l'équation (7.3) l'utilise avec \underline{y} comme référence.

(\Rightarrow) Considérons une sortie acceptable $y \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$. Lorsque $y = \bar{y}_s$:

- $-\mathcal{D}_{\bar{y}_s \neq y}(0) = -\mathcal{D}_{\bar{y}_s \neq \bar{y}_s}(0) = 0 \in [\bar{\Delta}_{\bar{y}_s}, 0]$ donc l'équation (7.2) est vérifiée.
- $\mathcal{D}_{y \neq \bar{y}_s}(0) = \mathcal{D}_{\bar{y}_s \neq \underline{y}_s}(0) = \underline{\Delta}_{\underline{y}_s} \in [0, \underline{\Delta}_{\underline{y}_s}]$ donc l'équation (7.3) est vérifiée.

Lorsque $y = \underline{y}_s$:

- $-\mathcal{D}_{\bar{y}_s \neq y}(0) = -\mathcal{D}_{\bar{y}_s \neq \underline{y}_s}(0) = \bar{\Delta}_{\bar{y}_s} \in [\bar{\Delta}_{\bar{y}_s}, 0]$ donc l'équation (7.2) est vérifiée.
- $\mathcal{D}_{y \neq \underline{y}_s}(0) = \mathcal{D}_{\underline{y}_s \neq \underline{y}_s}(0) = 0 \in [0, \underline{\Delta}_{\underline{y}_s}]$ donc l'équation (7.3) est vérifiée.

Enfin, lorsque $\underline{y}_s \preceq y \preceq \bar{y}_s$ (avec $y \neq \underline{y}_s, y \neq \bar{y}_s$) :

- Lorsque $y \neq \bar{y}_s$, $-\mathcal{D}_{\bar{y}_s \neq y}(0) \neq 0$. De plus, puisque $y \preceq \bar{y}_s$, y est situé graphiquement en-dessous de \bar{y}_s donc la distance temporelle avec \bar{y}_s comme référence sera à valeur négative soit $-\mathcal{D}_{\bar{y}_s \neq y}(0) < 0$.

D'autre part, lorsque $y \neq \underline{y}_s$, $-\mathcal{D}_{\bar{y}_s \neq y}(0) \neq \bar{\Delta}_{\bar{y}_s}$. De plus, puisque $\underline{y}_s \preceq y$, y est situé graphiquement au-dessus de \underline{y}_s donc la distance temporelle avec \underline{y}_s comme référence sera supérieure à $\bar{\Delta}_{\bar{y}_s}$ soit $-\mathcal{D}_{\bar{y}_s \neq y}(0) > \bar{\Delta}_{\bar{y}_s}$.

Par conséquent, si y évolue entre \underline{y}_s et \bar{y}_s , $\bar{\Delta}_{\bar{y}_s} < -\mathcal{D}_{\bar{y}_s \neq y}(0) < 0$. L'équation (7.2) est vérifiée.

- De la même manière, puisque y évolue entre \underline{y}_s et \bar{y}_s , $0 < \mathcal{D}_{y \neq \underline{y}_s}(0) < \underline{\Delta}_{\underline{y}_s}$. L'équation (7.3) est vérifiée.

(\Leftarrow)

- Si $\Delta(\bar{y}_s, y) = [\mathcal{D}_{y \neq \bar{y}_s}(0); -\mathcal{D}_{\bar{y}_s \neq y}(0)] = [0; 0]$, alors il n'y a pas de décalage temporel entre y et \bar{y}_s donc $y = \bar{y}_s$. Il s'agit d'une sortie acceptable vérifiant l'équation (7.2). Cela implique que $\mathcal{D}_{y \neq \underline{y}_s}(0) = \mathcal{D}_{\bar{y}_s \neq \underline{y}_s}(0) = \underline{\Delta}_{\underline{y}_s}$ donc y est une sortie acceptable vérifiant l'équation (7.3).

- Si $\Delta(\underline{y}_s, y) = [\mathcal{D}_{y \neq \underline{y}_s}(0); -\mathcal{D}_{\bar{y}_s \neq y}(0)] = [0; 0]$, alors il n'y a pas de décalage temporel entre y et \underline{y}_s donc $y = \underline{y}_s$. Il s'agit d'une sortie acceptable vérifiant l'équation (7.3). Cela implique que $-\mathcal{D}_{\bar{y}_s \neq y}(0) = -\mathcal{D}_{\bar{y}_s \neq \underline{y}_s}(0) = \bar{\Delta}_{\bar{y}_s}$ donc y est une sortie acceptable vérifiant l'équation (7.2).

- Si $-\mathcal{D}_{\bar{y}_s \neq y}(0) \in [\bar{\Delta}_{\bar{y}_s}, 0]$ (la condition (7.2) est vérifiée), alors $y \preceq \bar{y}_s$, et si $\mathcal{D}_{y \neq \underline{y}_s}(0) \notin [0, \underline{\Delta}_{\underline{y}_s}]$ (la condition (7.3) n'est pas vérifiée), alors ($y \preceq \underline{y}_s$ et $y \neq \underline{y}_s$). Par conséquent y est acceptable vis-à-vis de \bar{y}_s mais pas vis-à-vis de \underline{y}_s donc $y \notin [\underline{y}_s, \bar{y}_s]$.

De même, si $-\mathcal{D}_{\bar{y}_s \neq y}(0) \notin [\bar{\Delta}_{\bar{y}_s}, 0]$ (la condition (7.2) n'est pas vérifiée), alors ($y \succeq \bar{y}_s$ et $y \neq \bar{y}_s$), et si $\mathcal{D}_{y \neq \underline{y}_s}(0) \in [0, \underline{\Delta}_{\underline{y}_s}]$ (la condition (7.3) est vérifiée), alors $y \succeq \underline{y}_s$. Par conséquent y est acceptable vis-à-vis de \underline{y}_s mais pas vis-à-vis de \bar{y}_s donc $y \notin [\underline{y}_s, \bar{y}_s]$.

Ainsi, pour que $y \in [\underline{y}_s, \bar{y}_s]$, il faut que les conditions (7.2) et (7.3) soient vérifiées. Si l'une des deux est fausse, peu importe laquelle, y n'est pas dans l'intervalle $[\underline{y}_s, \bar{y}_s]$.

□

Exemple 7.4. Reprenons l'exemple 7.3 avec le GET de la figure 7.4 pour lequel les sorties sont représentées par les séries en bleu sur la figure 7.6. Les distances temporelles du système sont $\Delta_{\bar{y}_s} = [\underline{\Delta}_{\bar{y}_s}, \bar{\Delta}_{\bar{y}_s}] = [-30, -6]$ et $\Delta_{\underline{y}_s} = [\underline{\Delta}_{\underline{y}_s}, \bar{\Delta}_{\underline{y}_s}] = [6, 30]$. La sortie $y_1 = \gamma^0 \delta^7 \oplus \gamma^1 \delta^{10} \oplus \gamma^2 \delta^{12} \oplus \gamma^3 \delta^{15} \oplus \gamma^4 \delta^{20} \oplus \gamma^5 \delta^{25} \oplus \gamma^6 \delta^{30} \oplus \gamma^7 \delta^{+\infty}$ représentée en vert sur la figure est une sortie acceptable car :

$$\begin{aligned} -\mathcal{D}_{\bar{y}_s \neq y}(0) = -4 &\in [\bar{\Delta}_{\bar{y}_s}, 0] = [-6, 0] && \text{voir équation (7.2),} \\ \mathcal{D}_{y \neq \underline{y}_s}(0) = 2 &\in [0, \underline{\Delta}_{\underline{y}_s}] = [0, 6] && \text{voir équation (7.3).} \end{aligned}$$

La sortie $y_2 = \gamma^0\delta^4 \oplus \gamma^1\delta^5 \oplus \gamma^2\delta^6 \oplus \gamma^3\delta^7 \oplus \gamma^4\delta^8 \oplus \gamma^5\delta^9 \oplus \gamma^6\delta^{10} \oplus \gamma^7\delta^{+\infty}$ représentée en rouge sur la figure n'est pas une sortie acceptable car :

$$\begin{aligned} -\mathcal{D}_{\bar{y}_s \neq y}(0) &= -7 \notin [\bar{\Delta}_{\bar{y}_s}, 0] = [-6, 0] && \text{voir équation (7.2),} \\ \mathcal{D}_{y \neq \bar{y}_s}(0) &= -1 \notin [0, \Delta_{\bar{y}_s}] = [0, 6] && \text{voir équation (7.3).} \end{aligned}$$

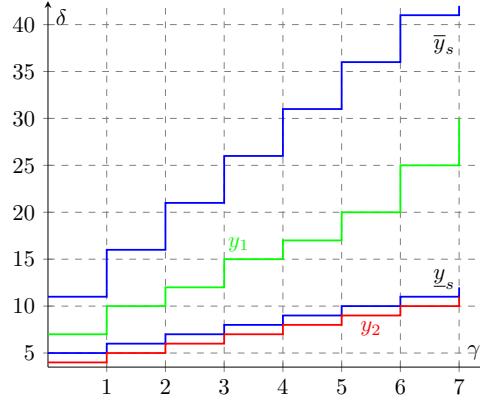


FIGURE 7.6 – Représentation de la sortie acceptable y_1 et de la sortie non acceptable y_2 par rapport à l'intervalle $\mathbf{y}_s = [\underline{y}_s, \bar{y}_s]$

La proposition suivante est une extension de la proposition 7.3 sur une sortie intervalle \mathbf{y} au lieu d'une sortie unique y .

Proposition 7.4. *Toutes les sorties contenues dans l'intervalle $\mathbf{y} = [y, \bar{y}] \in I(\mathcal{M}_{in}^{ax}[\gamma, \delta])$ sont des sorties acceptables, et donc appartiennent à l'ensemble \mathcal{Y}_{acc} , si et seulement si les deux conditions suivantes sont vraies :*

$$-\mathcal{D}_{\bar{y}_s \neq \bar{y}}(0) \in [\bar{\Delta}_{\bar{y}_s}, 0], \quad (7.4)$$

$$\mathcal{D}_{y \neq \underline{y}_s}(0) \in [0, \Delta_{\underline{y}_s}]. \quad (7.5)$$

Démonstration. Puisque $\underline{y} \preceq \bar{y}$, la démonstration de cette proposition est la même que celle de la proposition 7.3 avec :

- $y = \bar{y}$ pour l'équation (7.4) à la place de l'équation (7.2),
- $y = \underline{y}$ pour l'équation (7.5) à la place de l'équation (7.3).

□

7.4 Détection de dérives temporelles dans un système $(\max, +)$ -linéaire à temps intervalles

À partir des définitions des sorties acceptables pour un système $(\max, +)$ -linéaire à temps intervalles, deux fonctions de détection de dérives temporelles sont maintenant présentées : l'une pour une sortie observée *certaine* représentant une observation exacte de la sortie réelle du système, l'autre pour une sortie observée *incertaine* représentant un intervalle d'observations dont on ne connaît que les bornes minimale et maximale.

Une sortie *certaine* $\mathbf{y} = [\underline{y}, \bar{y}]$ est une sortie exprimée par un intervalle dégénéré de $I(\mathcal{M}_{in}^{ax}[\gamma, \delta])$, c'est-à-dire que la borne inférieure est identique à la borne supérieure $\underline{y} = \bar{y}$. Comme indiquée en début de section 7.1, une telle sortie est plus simplement notée y (sans la mise en forme en gras). Une sortie *incertaine* $\mathbf{y} = [\underline{y}, \bar{y}]$ est un intervalle classique où $\underline{y} \preceq \bar{y}$.

Les hypothèses faites pour les indicateurs sont les suivantes : l'entrée \mathbf{u} , entrée intervalle, et la fonction de transfert \mathbf{H} sont connues, les sorties du modèle sans faute sont calculées $\mathbf{y}_s = [\underline{y}_s, \bar{y}_s] = \mathbf{H}\mathbf{u}$, la mesure d'une sortie \mathbf{y}_o du système réel est une sortie certaine ou incertaine. Les indicateurs font la comparaison entre cette observation du système réel et la sortie du modèle sans faute du système. De plus, afin de définir les indicateurs de ce chapitre, la définition 4.4 d'une dérive temporelle détectable donnée dans le chapitre 4 doit être renforcée dans le cas des GET intervalles.

Définition 7.11 (Dérive temporelle détectable dans un GET intervalles). Une dérive temporelle $\theta \in \mathbb{Z}$ située sur une place p d'un GET intervalles est *détectable* si au moins l'une des transitions de sortie observable en aval de cette place (y_{oi} telle que $p \rightsquigarrow y_{oi}$) n'appartient plus à l'ensemble des sorties acceptables \mathcal{Y}_{acc} , soit $y_{oi} \notin [\underline{y}_s, \bar{y}_s]$.

Par la suite, nous continuons d'utiliser le simple terme de *dérive temporelle* pour désigner une *dérive temporelle détectable*.

7.4.1 Indicateur dans les cas d'une sortie certaine

Le premier indicateur de dérives temporelles présenté est celui dans le cas d'une sortie observée certaine noté y_o . Cet indicateur vérifie que la sortie y_o est acceptable d'après les conditions des équations (7.2) et (7.3).

Définition 7.12 (Indicateur pour une sortie certaine). Soient \mathbf{u} l'entrée appliquée au système et \mathbf{H} sa fonction de transfert, l'indicateur de dérive temporelle associé à la sortie certaine y_o est noté $I(\mathbf{u}, y_o)$ et est défini par la fonction booléenne suivante :

$$I(\mathbf{u}, y_o) = \begin{cases} \text{faux} & \text{si pour } \mathbf{y}_s = \mathbf{H}\mathbf{u}, \\ & -\mathcal{D}_{\bar{y}_s \neq y_o}(0) \in [\bar{\Delta}_{\bar{y}_s}, 0] \text{ et } \mathcal{D}_{y_o \neq \underline{y}_s}(0) \in [0, \underline{\Delta}_{\underline{y}_s}], \\ \text{vrai} & \text{sinon.} \end{cases}$$

Proposition 7.5. *L'indicateur $I(\mathbf{u}, y_o)$ renvoie vrai si et seulement si une dérive temporelle est présente dans le système.*

Démonstration. La proposition 7.3 assure que la sortie y_o est acceptable par rapport aux sorties du modèle sans faute $\mathbf{y}_s = \mathbf{H}\mathbf{u}$ si et seulement si les équations (7.2) et (7.3) sont exactes. Si la sortie est acceptable, il n'y a pas de dérive temporelle³ dans le système, l'indicateur renvoie faux. Si elle n'est pas acceptable (au moins l'une des deux conditions de (7.2) et (7.3) est fausse), une dérive temporelle est nécessairement présente, l'indicateur renvoie vrai. \square

Exemple 7.5. Reprenons l'exemple 7.3 avec le GET de la figure 7.4. Une première sortie certaine observée du système dans lequel aucune dérive temporelle n'est présente est :

$$y_{o1} = \gamma^0 \delta^7 \oplus \gamma^1 \delta^{10} \oplus \gamma^2 \delta^{13} \oplus \gamma^3 \delta^{16} \oplus \gamma^4 \delta^{19} \oplus \gamma^5 \delta^{22} \oplus \gamma^6 \delta^{25} \oplus \gamma^7 \delta^{+\infty}.$$

Elle est représentée figure 7.7 avec les sorties du modèle sans faute \mathbf{y}_s . Les calculs de $\bar{y}_s \neq y_{o1}$ et

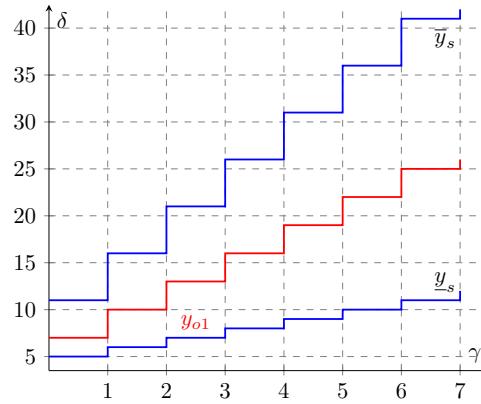


FIGURE 7.7 – Représentation de la sortie certaine y_{o1} et des sorties du modèle sans faute \mathbf{y}_s

$y_{o1} \neq \underline{y}_s$ donnent les résultats suivants :

$$\begin{aligned} \bar{y}_s \neq y_{o1} &= \gamma^0 \delta^4 \oplus \gamma^1 \delta^9 \oplus \gamma^2 \delta^{14} \oplus \gamma^3 \delta^{19} \oplus \gamma^4 \delta^{24} \oplus \gamma^5 \delta^{29} \oplus \gamma^6 \delta^{34} \oplus \gamma^7 \delta^{+\infty}, \\ y_{o1} \neq \underline{y}_s &= \gamma^0 \delta^2 \oplus \gamma^1 \delta^5 \oplus \gamma^2 \delta^8 \oplus \gamma^3 \delta^{11} \oplus \gamma^4 \delta^{14} \oplus \gamma^5 \delta^{17} \oplus \gamma^6 \delta^{20} \oplus \gamma^7 \delta^{+\infty}. \end{aligned}$$

3. Au sens de la définition 7.11.

Ainsi, l'indicateur obtenu pour cet exemple est :

$$I(\mathbf{u}, y_{o1}) = \text{faux} \quad \text{car} \quad -\mathcal{D}_{\bar{y}_s \neq y_{o1}}(0) = -4 \in [-6, 0] \quad \text{et} \quad \mathcal{D}_{y_{o1} \neq \underline{y}_s}(0) = 2 \in [0, 6].$$

Il n'y a pas de dérive temporelle détectée dans ce système pour cette sortie observée.

Exemple 7.6. Soit une seconde sortie observée certaine pour le GET de la figure 7.4 avec une dérive temporelle $\theta = 1$ sur p_2 :

$$y_{o2} = \gamma^0 \delta^{10} \oplus \gamma^1 \delta^{16} \oplus \gamma^2 \delta^{22} \oplus \gamma^3 \delta^{28} \oplus \gamma^4 \delta^{34} \oplus \gamma^5 \delta^{40} \oplus \gamma^6 \delta^{46} \oplus \gamma^7 \delta^{+\infty}$$

Elle est représentée figure 7.8 avec les sorties du modèle sans faute \mathbf{y}_s . Les calculs de $\bar{y}_s \neq y_{o2}$ et

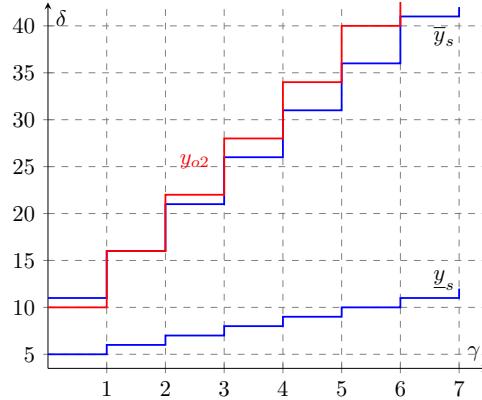


FIGURE 7.8 – Représentation de la sortie certaine y_{o2} et des sorties du modèle sans faute \mathbf{y}_s

$y_{o2} \neq \underline{y}_s$ donnent les résultats suivants :

$$\begin{aligned} \bar{y}_s \neq y_{o2} &= \gamma^0 \delta^{-5} \oplus \gamma^1 \delta^1 \oplus \gamma^2 \delta^7 \oplus \gamma^3 \delta^{13} \oplus \gamma^4 \delta^{19} \oplus \gamma^5 \delta^{25} \oplus \gamma^6 \delta^{31} \oplus \gamma^7 \delta^{+\infty}, \\ y_{o2} \neq \underline{y}_s &= \gamma^0 \delta^5 \oplus \gamma^1 \delta^{11} \oplus \gamma^2 \delta^{17} \oplus \gamma^3 \delta^{23} \oplus \gamma^4 \delta^{29} \oplus \gamma^5 \delta^{35} \oplus \gamma^6 \delta^{41} \oplus \gamma^7 \delta^{+\infty}. \end{aligned}$$

L'indicateur obtenu pour cet exemple est alors :

$$I(\mathbf{u}, y_{o2}) = \text{vrai} \quad \text{car} \quad -\mathcal{D}_{\bar{y}_s \neq y_{o2}}(0) = 5 \notin [-6, 0] \quad \text{et} \quad \mathcal{D}_{y_{o2} \neq \underline{y}_s}(0) = 5 \in [0, 6].$$

Une dérive temporelle est détectée.

Exemple 7.7. Reprenons à présent l'exemple 7.2 avec le GET de la figure 7.2. Les distances temporelles du système sont :

$$\begin{aligned} \Delta_{\bar{y}_s} &= [\underline{\Delta}_{\bar{y}_s}, \bar{\Delta}_{\bar{y}_s}] = [-24, -6], \\ \Delta_{\underline{y}_s} &= [\underline{\Delta}_{\underline{y}_s}, \bar{\Delta}_{\underline{y}_s}] = [6, 24], \end{aligned}$$

donc les intervalles à prendre en compte pour la détection de dérives temporelles sont :

$$\begin{aligned} [0, \underline{\Delta}_{\underline{y}_s}] &= [0, 6], \\ [\bar{\Delta}_{\bar{y}_s}, 0] &= [-6, 0]. \end{aligned}$$

Soit une première sortie observée certaine du système pour lequel aucune dérive temporelle n'est présente :

$$y_{o1} = \gamma^0 \delta^{10} \oplus \gamma^1 \delta^{14} \oplus \gamma^2 \delta^{18} \oplus \gamma^3 \delta^{22} \oplus \gamma^4 \delta^{26} \oplus \gamma^5 \delta^{30} \oplus \gamma^6 \delta^{34} \oplus \gamma^7 \delta^{+\infty}.$$

Elle est représentée figure 7.9 avec les sorties du modèle sans faute \mathbf{y}_s . L'indicateur obtenu pour cet exemple est alors :

$$I(\mathbf{u}, y_{o1}) = \text{faux} \quad \text{car} \quad -\mathcal{D}_{\bar{y}_s \neq y_{o1}}(0) = -1 \in [-6, 0] \quad \text{et} \quad \mathcal{D}_{y_{o1} \neq \underline{y}_s}(0) = 5 \in [0, 6].$$

Il n'y a pas de dérive temporelle dans ce système pour cette sortie y_{o1} .

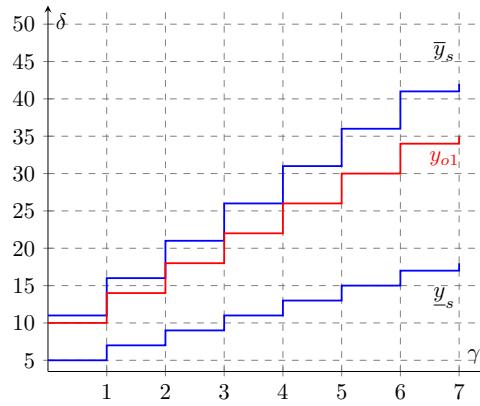


FIGURE 7.9 – Représentation de la sortie certaine y_{o1} et des sorties du modèle sans faute \mathbf{y}_s

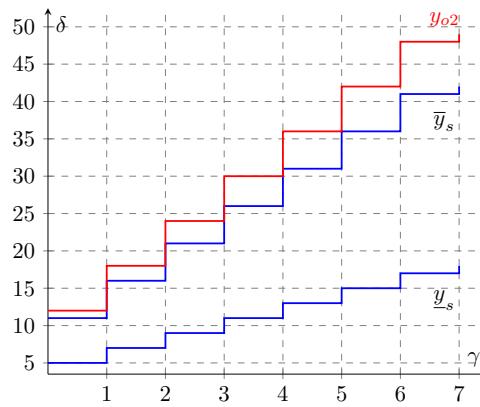


FIGURE 7.10 – Représentation de la sortie certaine y_{o2} et des sorties du modèle sans faute \mathbf{y}_s

Exemple 7.8. Soit une seconde sortie observée certaine pour le GET de la figure 7.2 avec une dérive temporelle $\theta = 1$ sur p_4 :

$$y_{o2} = \gamma^0 \delta^{12} \oplus \gamma^1 \delta^{18} \oplus \gamma^2 \delta^{24} \oplus \gamma^3 \delta^{30} \oplus \gamma^4 \delta^{36} \oplus \gamma^5 \delta^{42} \oplus \gamma^6 \delta^{48} \oplus \gamma^7 \delta^{+\infty}.$$

Elle est représentée figure 7.10 avec les sorties du modèle sans faute \mathbf{y}_s . L'indicateur obtenu pour cet exemple est alors :

$$I(\mathbf{u}, y_{o2}) = \text{vrai} \quad \text{car} \quad -\mathcal{D}_{\bar{y}_s \neq y_{o2}}(0) = 7 \notin [-6, 0] \quad \text{et} \quad \mathcal{D}_{y_{o2} \neq \bar{y}_s}(0) = 7 \notin [0, 6].$$

Une dérive temporelle est détectée.

7.4.2 Indicateur dans les cas d'une sortie incertaine

Le second indicateur de dérives temporelles est celui dans le cas d'une sortie observée incertaine notée $\mathbf{y}_o = [\underline{y}_o, \bar{y}_o]$. Cet indicateur, qui est une extension de l'indicateur défini en 7.12, vérifie que la sortie \mathbf{y}_o est acceptable d'après les conditions des équations (7.4) et (7.5).

Définition 7.13 (Indicateur pour une sortie incertaine). Soient \mathbf{u} l'entrée appliquée au système et \mathbf{H} sa fonction de transfert, l'indicateur de dérive temporelle associé à la sortie incertaine \mathbf{y}_o est noté $I(\mathbf{u}, \mathbf{y}_o)$ et est défini par la fonction booléenne suivante pour $\mathbf{y}_s = \mathbf{H}\mathbf{u}$:

$$I(\mathbf{u}, \mathbf{y}_o) = \begin{cases} \text{faux} & \text{si } -\mathcal{D}_{\bar{y}_s \neq \bar{y}_o}(0) \in [\bar{\Delta}_{\bar{y}_s}, 0] \text{ et } \mathcal{D}_{\underline{y}_o \neq \underline{y}_s}(0) \in [0, \bar{\Delta}_{\underline{y}_s}], \\ \text{vrai} & \text{si } -\mathcal{D}_{\bar{y}_s \neq \bar{y}_o}(0) \notin [\bar{\Delta}_{\bar{y}_s}, 0] \text{ et } \mathcal{D}_{\underline{y}_o \neq \underline{y}_s}(0) \notin [0, \bar{\Delta}_{\underline{y}_s}], \\ \text{incertain} & \text{sinon.} \end{cases}$$

Proposition 7.6. L'indicateur $I(\mathbf{u}, \mathbf{y}_o)$ renvoie vrai si une dérive temporelle est présente dans le système.

Démonstration. La proposition 7.4 assure que l'ensemble des sorties observées de l'intervalle $\mathbf{y}_o = [\underline{y}_o, \bar{y}_o]$ est acceptable par rapport aux sorties du modèle sans faute $\mathbf{y}_s = \mathbf{H}\mathbf{u}$ si et seulement si les équations (7.4) et (7.5) sont vérifiées. Si la sortie est acceptable, il n'y a pas de dérive temporelle⁴ dans le système, l'indicateur renvoie faux. Si elle n'est pas acceptable (les deux conditions de (7.4) et (7.5) sont fausses), une dérive temporelle est nécessairement présente, l'indicateur renvoie vrai. Si l'une des conditions parmi les équations (7.4) et (7.5) est fausse mais que l'autre est vraie, on ne peut pas conclure sur la présence ou non de la dérive temporelle. Les intervalles \mathbf{y}_o et \mathbf{y}_s se chevauchent, on ne sait pas où se situe la sortie réelle au sein de \mathbf{y}_o . L'indicateur renvoie alors la valeur "incertain".

□

Exemple 7.9. Reprenons le GET 7.2 de l'exemple 7.2 pour lequel $\Delta_{\bar{y}_s} = [-24, -6]$ et $\Delta_{\underline{y}_s} = [6, 24]$. Soit une première sortie observée incertaine du système pour lequel aucune dérive temporelle n'est présente :

$$\begin{aligned}\mathbf{y}_{o1} &= [\underline{y}_{o1}, \bar{y}_{o1}] \\ &= [\gamma^0 \delta^8 \oplus \gamma^1 \delta^{11} \oplus \gamma^2 \delta^{14} \oplus \gamma^3 \delta^{17} \oplus \gamma^4 \delta^{20} \oplus \gamma^5 \delta^{23} \oplus \gamma^6 \delta^{26} \oplus \gamma^7 \delta^{+\infty}, \\ &\quad \gamma^0 \delta^{10} \oplus \gamma^1 \delta^{14} \oplus \gamma^2 \delta^{18} \oplus \gamma^3 \delta^{22} \oplus \gamma^4 \delta^{26} \oplus \gamma^5 \delta^{30} \oplus \gamma^6 \delta^{34} \oplus \gamma^7 \delta^{+\infty}].\end{aligned}$$

Elle est représentée figure 7.11 avec les sorties du modèle sans faute \mathbf{y}_s . Les calculs de $\bar{y}_s \not\in \bar{y}_{o1}$ et

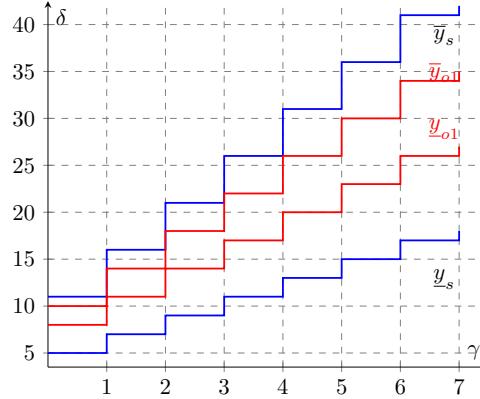


FIGURE 7.11 – Représentation de la sortie incertaine \mathbf{y}_{o1} et des sorties du modèle sans faute \mathbf{y}_s

$\underline{y}_{o1} \not\in \underline{y}_s$ donnent les résultats suivants :

$$\begin{aligned}\bar{y}_s \not\in \bar{y}_{o1} &= \gamma^0 \delta^5 \oplus \gamma^1 \delta^9 \oplus \gamma^2 \delta^{13} \oplus \gamma^3 \delta^{17} \oplus \gamma^4 \delta^{21} \oplus \gamma^5 \delta^{25} \oplus \gamma^6 \delta^{29} \oplus \gamma^7 \delta^{+\infty}, \\ \underline{y}_{o1} \not\in \underline{y}_s &= \gamma^0 \delta^3 \oplus \gamma^1 \delta^6 \oplus \gamma^2 \delta^9 \oplus \gamma^3 \delta^{12} \oplus \gamma^4 \delta^{15} \oplus \gamma^5 \delta^{18} \oplus \gamma^6 \delta^{21} \oplus \gamma^7 \delta^{+\infty}.\end{aligned}$$

L'indicateur obtenu pour cet exemple est alors :

$$I(\mathbf{u}, \mathbf{y}_{o1}) = \text{faux} \quad \text{car} \quad -\mathcal{D}_{\bar{y}_s \not\in \bar{y}_{o1}}(0) = -5 \in [-6, 0] \quad \text{et} \quad \mathcal{D}_{\underline{y}_{o1} \not\in \underline{y}_s}(0) = 3 \in [0, 6].$$

Il n'y a pas de dérive temporelle détectée.

Exemple 7.10. Soit une deuxième sortie observée incertaine pour le GET de la figure 7.2 avec deux dérives temporelles : $\theta = 2$ sur p_7 et $\theta = 3$ sur p_{10} :

$$\begin{aligned}\mathbf{y}_{o2} &= [\underline{y}_{o2}, \bar{y}_{o2}] \\ &= [\gamma^0 \delta^{13} \oplus \gamma^1 \delta^{18} \oplus \gamma^2 \delta^{23} \oplus \gamma^3 \delta^{28} \oplus \gamma^4 \delta^{33} \oplus \gamma^5 \delta^{38} \oplus \gamma^6 \delta^{43} \oplus \gamma^7 \delta^{+\infty}, \\ &\quad \gamma^0 \delta^{14} \oplus \gamma^1 \delta^{20} \oplus \gamma^2 \delta^{26} \oplus \gamma^3 \delta^{32} \oplus \gamma^4 \delta^{38} \oplus \gamma^5 \delta^{44} \oplus \gamma^6 \delta^{50} \oplus \gamma^7 \delta^{+\infty}].\end{aligned}$$

représentée figure 7.12 avec les sorties du modèle sans faute \mathbf{y}_s . Les calculs de $\bar{y}_s \not\in \bar{y}_{o2}$ et $\underline{y}_{o2} \not\in \underline{y}_s$ donnent les résultats suivants :

$$\begin{aligned}\bar{y}_s \not\in \bar{y}_{o2} &= \gamma^0 \delta^{-9} \oplus \gamma^1 \delta^{-3} \oplus \gamma^2 \delta^3 \oplus \gamma^3 \delta^9 \oplus \gamma^4 \delta^{15} \oplus \gamma^5 \delta^{21} \oplus \gamma^6 \delta^{27} \oplus \gamma^7 \delta^{+\infty}, \\ \underline{y}_{o2} \not\in \underline{y}_s &= \gamma^0 \delta^8 \oplus \gamma^1 \delta^{13} \oplus \gamma^2 \delta^{18} \oplus \gamma^3 \delta^{23} \oplus \gamma^4 \delta^{28} \oplus \gamma^5 \delta^{33} \oplus \gamma^6 \delta^{38} \oplus \gamma^7 \delta^{+\infty}.\end{aligned}$$

4. Au sens de la définition 7.11.

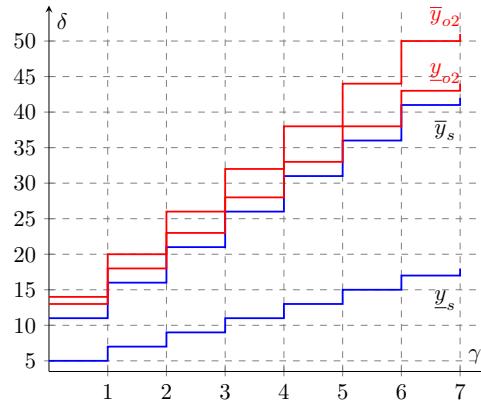


FIGURE 7.12 – Représentation de la sortie incertaine \mathbf{y}_{o2} et des sorties du modèle sans faute \mathbf{y}_s

L'indicateur obtenu pour cet exemple est alors :

$$I(\mathbf{u}, \mathbf{y}_{o2}) = \text{vrai} \quad \text{car} \quad -\mathcal{D}_{\bar{y}_s \not\subset \bar{y}_{o2}}(0) = 9 \notin [-6, 0] \quad \text{et} \quad \mathcal{D}_{\underline{y}_{o2} \not\subset \underline{y}_s}(0) = 8 \notin [0, 6].$$

Une dérive temporelle est détectée pour ce système avec cette sortie incertaine.

Exemple 7.11. Soit une troisième sortie observée incertaine, représentée figure 7.13, pour le GET de la figure 7.2 avec une dérive temporelle $\theta = 3$ sur p_{10} :

$$\begin{aligned} \mathbf{y}_{o3} &= [\underline{y}_{o3}, \bar{y}_{o3}] \\ &= [\gamma^0 \delta^{10} \oplus \gamma^1 \delta^{14} \oplus \gamma^2 \delta^{18} \oplus \gamma^3 \delta^{22} \oplus \gamma^4 \delta^{26} \oplus \gamma^5 \delta^{30} \oplus \gamma^6 \delta^{34} \oplus \gamma^7 \delta^{+\infty}, \\ &\quad \gamma^0 \delta^{14} \oplus \gamma^1 \delta^{20} \oplus \gamma^2 \delta^{26} \oplus \gamma^3 \delta^{32} \oplus \gamma^4 \delta^{38} \oplus \gamma^5 \delta^{44} \oplus \gamma^6 \delta^{50} \oplus \gamma^7 \delta^{+\infty}]. \end{aligned}$$

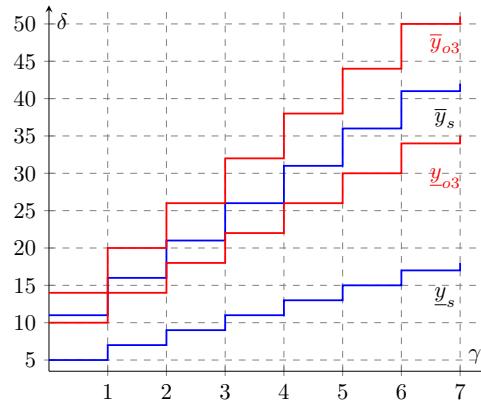


FIGURE 7.13 – Représentation de la sortie incertaine \mathbf{y}_{o3} et des sorties du modèle sans faute \mathbf{y}_s

Les calculs de $\bar{y}_s \not\subset \bar{y}_{o3}$ et $\underline{y}_{o3} \not\subset \underline{y}_s$ donnent les résultats suivants :

$$\begin{aligned} \bar{y}_s \not\subset \bar{y}_{o3} &= \gamma^0 \delta^{-9} \oplus \gamma^1 \delta^{-3} \oplus \gamma^2 \delta^3 \oplus \gamma^3 \delta^9 \oplus \gamma^4 \delta^{15} \oplus \gamma^5 \delta^{21} \oplus \gamma^6 \delta^{27} \oplus \gamma^7 \delta^{+\infty}, \\ \underline{y}_{o3} \not\subset \underline{y}_s &= \gamma^0 \delta^5 \oplus \gamma^1 \delta^9 \oplus \gamma^2 \delta^{13} \oplus \gamma^3 \delta^{17} \oplus \gamma^4 \delta^{21} \oplus \gamma^5 \delta^{25} \oplus \gamma^6 \delta^{29} \oplus \gamma^7 \delta^{+\infty}. \end{aligned}$$

L'indicateur obtenu pour cet exemple est alors :

$$I(\mathbf{u}, \mathbf{y}_{o3}) = \text{incertain} \quad \text{car} \quad -\mathcal{D}_{\bar{y}_s \not\subset \bar{y}_{o3}}(0) = 9 \notin [-6, 0] \quad \text{et} \quad \mathcal{D}_{\underline{y}_{o3} \not\subset \underline{y}_s}(0) = 3 \in [0, 6].$$

Cela signifie qu'on ne peut pas conclure si une dérive temporelle est présente dans le système ou non car l'une des bornes de la sortie observée \underline{y}_{o3} est dans l'intervalle des sorties acceptables alors que l'autre \bar{y}_{o3} ne l'est pas. L'incertitude sur la sortie ne nous permet pas de savoir avec exactitude où se situe la sortie réelle du système ni de détecter si une dérive temporelle est présente ou non.

7.5 Synthèse du chapitre

Ce chapitre présente les définitions liées aux dioïdes intervalles permettant de faire une modélisation des systèmes $(\max, +)$ -linéaires à temps intervalles. Ces systèmes représentés par des GET intervalles sont des systèmes dont les opérations (temps de transport, de processus...) sont soumises à des durées variables dont on ne connaît que la borne minimale, soit le comportement le plus rapide, et la borne maximale, soit le comportement le plus lent.

Les sorties acceptables caractérisant ces systèmes sont définies et utilisées dans l'indicateur de dérives temporelles pour une sortie certaine issue d'une observation exacte, et dans l'indicateur pour une sortie incertaine issue d'un intervalle d'observations. L'indicateur avec sortie certaine peut être utilisé pour faire une détection des dérives temporelles sur les modèles d'étapes que nous avons définis au chapitre 4.

Le chapitre suivant présente un exemple de l'utilisation de cette méthode de détection avec des données réelles de l'usine. Le modèle étudié est celui d'une opération constituée de deux étapes, étapes formellement décrite dans le chapitre 4. Cependant, la modélisation de ce problème industriel ne permet pas d'obtenir un unique système $(\max, +)$ -linéaire intervalle et son GET associé. En revanche, nous avons obtenu deux GET différents permettant de représenter deux comportements temporels de l'étape : l'un est utilisé comme borne minimale car il modélise la durée minimale de l'étape, soit son comportement le plus rapide; l'autre est utilisé comme borne maximale pour modéliser le comportement le plus lent. Nous pouvons ainsi appliquer notre méthode de détection de dérive temporelle proposée pour les systèmes intervalles.

Chapitre 8

Exemple de diagnostic de dérives temporelles pour une opération de l'usine

Ce chapitre présente un exemple d'une mise en place de nos méthodes de diagnostic de dérives temporelles pour une opération de l'usine en particulier sur une journée d'activité. Les dérives temporelles qu'on cherche à détecter et localiser sont les retards entre les dates planifiées de production des lots et les dates des lots effectivement produits. Ces méthodes sont appliquées sur les modèles GET d'une étape définis dans le chapitre 4. Les données de l'outil de planification OPERA et celles de l'historique de la production présentées dans le chapitre 2 sont utilisées pour représenter les trajectoires des entrées et sorties des GET.

Dans un premier temps, la méthode de détection par intervalles du chapitre 7 est appliquée. La complexité du problème industriel n'ayant pas permis d'obtenir un seul GET intervalle pour modéliser une étape mais deux GET classiques, ces deux GET sont utilisés en tant que borne minimale, comportement le plus rapide de l'étape, et borne maximale, comportement le plus lent, de la dynamique du système. Ensuite, afin de réaliser une localisation des dérives temporelles détectées, les observateurs présentés dans le chapitre 6 sont mis en oeuvre.

8.1 Présentation générale de l'exemple étudié

L'exemple étudié est une opération o sur des lots de plaques pour une journée d'activité. Cette opération d'une brique d'une route est constituée de la succession de deux étapes notées s_1 et s_2 . Elle se fait pour une technologie t_e donnée (voir sections 4.2.1 et 4.2.2 du chapitre 4). Les modèles utilisés pour représenter cette opération de deux étapes sont issus de la modélisation GET d'une étape définie en section 4.2.3 du chapitre 4.

Cette opération est étudiée pour une planification de lots à produire durant la journée du 15 novembre 2020. Cette journée fait partie d'une planification d'une semaine qui débute le 13 novembre 2020 (les planification sont faites à la semaine) et dont les informations sont issues du fichier LOTS PROJECTION d'OPERA (fichier défini dans la sous-section 2.1.2 du chapitre 2). De plus, il est important de préciser que pour STMicroelectronics, une journée démarre à 5h du matin et dure 24h. Donc l'origine temporel de la journée étudiée est le 15 novembre à 5h. Puisque l'étude porte sur une opération, les données historiques utilisées pour connaître la production réelle des lots sur cette opération sont issues du fichier MOUV_OUT qui agrège les informations du fichier TGVHistorique (fichier défini dans la sous-section 2.1.2 du chapitre 2) au niveau de l'opération.

Les méthodes de diagnostic de dérives temporelles des chapitres 6 et 7 sont définies pour des données connues et mesurées. Ainsi :

- Les entrées u des GET qui représentent l'opération sont les données issues de l'historique.
- Ces entrées permettent d'obtenir les sorties des modèles sans faute y_s de l'opération connaissant les fonctions de transfert H des GET.
- Enfin, les sorties mesurées y_o sont les données issues de l'historique de production, ce sont elles qui sont comparées aux sorties y_s .

8.2 Transformation des données de l'usine en séries de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$

Pour appliquer les méthodes de diagnostic des chapitres 6 et 7, les données issues des fichiers de l'usine doivent être transformées en séries du dioïde $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ pour servir d'entrée connue u et de sortie observée y_o des GET qui modélisent une étape.

8.2.1 Transformation des données de l'historique en entrée u

La colonne StepOut du fichier historique MOUV_OUT¹ fournit les dates de sortie d'une étape pour chaque lot. La sortie d'une étape pour un lot correspondant à son entrée dans l'étape suivante, ces informations vont nous servir pour représenter les dates et nombre de lots en entrée de l'étape s_1 et donc de l'opération o . À partir de cette colonne, la figure 8.1 donne pour chaque heure de la journée étudiée le nombre de lots en entrée de l'opération. Si une heure n'est pas représentée sur l'axe des abscisses (par exemple 7h, 9h, 11h..., 19h, 21h, 23h...), c'est qu'aucun lot n'entre dans l'opération pour cette heure-là. La première barre correspond à 1 lot en entrée le 15 novembre à 5h, la seconde montre que 2 lots entrent dans l'opération à 6h (aucun lot n'a été planifié à 7h), etc. Les informations de ce graphique sont utilisées pour modéliser l'entrée u historique des GET d'une opération en tant que série de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$.

Remarque. Si la méthode de détection par intervalle nécessite des séries u appartenant au dioïde $I(\mathcal{M}_{in}^{ax}[\gamma, \delta])$, les informations de l'historique pour cette opération sur cette journée étant uniques, elles peuvent être transformées en une série dégénérée du dioïde intervalle. Pour plus de simplicité, nous traiterons ces séries directement comme des séries de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$.

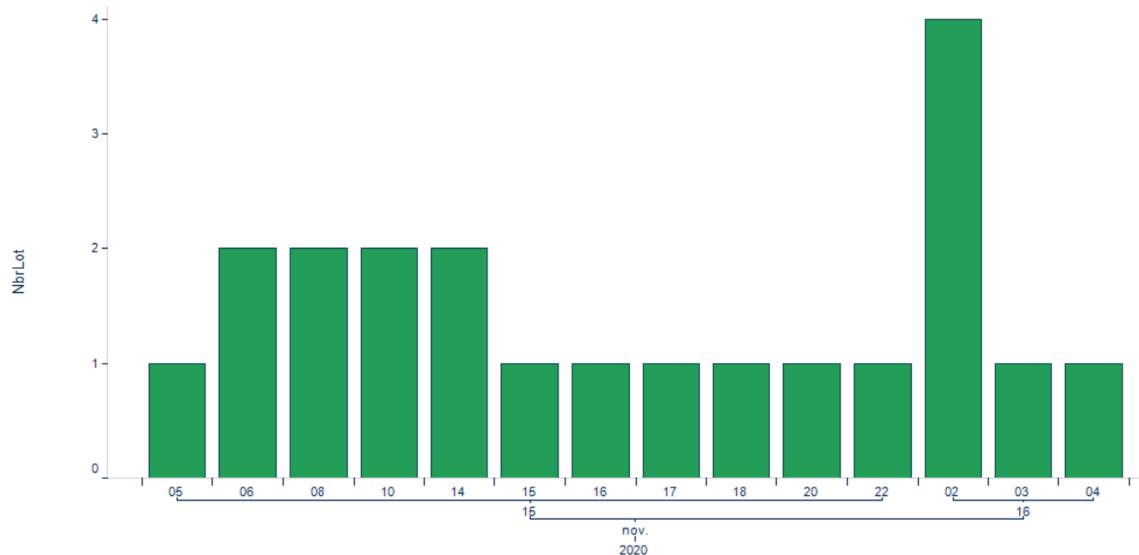


FIGURE 8.1 – Nombre de lots en entrée de l'opération o par heure, données historiques

Dans la série $u \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$ obtenue à partir de la figure, les valeurs des exposants de γ correspondent aux nombres de lots qui sont entrés dans l'opération et les valeurs des exposants de δ aux heures de ces entrées. L'historique étudié démarre le 15 novembre à 5h, heure à laquelle le premier lot entre dans l'opération. Donc le premier événement de la série u , numéroté 0, a pour date 0. Ensuite, le graphique indique que 2 lots, numérotés 1 et 2, entrent dans l'opération à 6h, soit à la date 1 (= 1h après le début de la journée du 15 novembre à 5h). Cela donne comme premiers monômes de la série :

$$\gamma^0 \delta^0 \oplus \gamma^1 \delta^1 \oplus \gamma^2 \delta^1.$$

Ensuite, deux autres lots, numérotés 3 et 4 entrent à nouveau dans l'opération à 8h (aucun lot n'entre à 7h) soit 2h après les lots précédents (événements numéros 1 et 2 de la série), soit à la date 3 = 1 + 2. Si on ajoute cette information aux monômes déjà obtenus, cela donne :

$$\gamma^0 \delta^0 \oplus \gamma^1 \delta^1 \oplus \gamma^2 \delta^1 \oplus \gamma^3 \delta^3 \oplus \gamma^4 \delta^3.$$

1. Données du fichier TGVHistorique agrégées au niveau de l'opération.

Ce début de série peut être simplifié (le monôme $\gamma^2\delta^1$ disparaît) grâce aux règles de la représentation minimale d'une série (définition 5.27) :

$$\gamma^0\delta^0 \oplus \gamma^1\delta^1 \oplus \gamma^3\delta^3 \oplus \gamma^4\delta^3.$$

Cette transformation et simplification s'appliquent pour tout le reste du graphique et permettent d'obtenir la série u suivante :

$$u = \gamma^0\delta^0 \oplus \gamma^1\delta^1 \oplus \gamma^3\delta^3 \oplus \gamma^5\delta^5 \oplus \gamma^7\delta^9 \oplus \gamma^9\delta^{10} \oplus \gamma^{10}\delta^{11} \oplus \gamma^{11}\delta^{12} \oplus \gamma^{12}\delta^{15} \oplus \gamma^{13}\delta^{17} \oplus \gamma^{14}\delta^{20} \oplus \gamma^{15}\delta^{21} \oplus \gamma^{19}\delta^{22} \oplus \gamma^{20}\delta^{23} \oplus \gamma^{21}\delta^{+\infty}.$$

On peut constater que les informations de la première barre du graphique sont réparties dans les deux premiers monômes de la série u : l'information temporelle de la date d'entrée se retrouve sur l'exposant de δ du premier monôme (0), l'information événementielle du nombre² de lots est l'exposant de γ du deuxième monôme (1). Les informations de la deuxième barre se retrouvent alors dans les monômes 2 et 3 de la série. La date d'entrée des lots est l'exposant de δ du deuxième monôme (1), le nombre de lots (2) se retrouve dans la différence entre les exposants de γ des monômes $\gamma^3\delta^3$ et $\gamma^1\delta^1$ soit $3 - 1 = 2$. Etc. pour la troisième barre... Enfin, cette journée compte 21 lots, information traduite en série de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ par le dernier monôme $\gamma^{21}\delta^{+\infty}$ (puisque les lots sont numérotés de 0 à 20).

8.2.2 Transformation des données de l'historique en sortie y_o

A nouveau, la colonne StepOut du fichier historique MOUV_OUT est utilisé pour retrouver les dates de sortie des lots par opération ainsi que le nombre de lots pour chaque date, afin de transformer ces données de sortie observable du système en série de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$. À partir de cette colonne, la figure 8.2 donne pour chaque heure de la journée étudiée le nombre de lots en sortie de l'opération o .

Pour la journée étudiée du 15 novembre 5h, le premier lot sort de l'opération à 6h, le second à 7h, puis 3 lots sortent à 8h, aucun lot ne sort entre 9h et 13h, puis 3 lots sortent à 14h, etc. Pour que le nombre de lots en sortie de l'opération soit identique au nombre de lots en entrée (21 lots sont entrés), l'historique doit être analysé jusqu'au 16 novembre à 5h (date à laquelle le 21ème lot sort de l'opération).

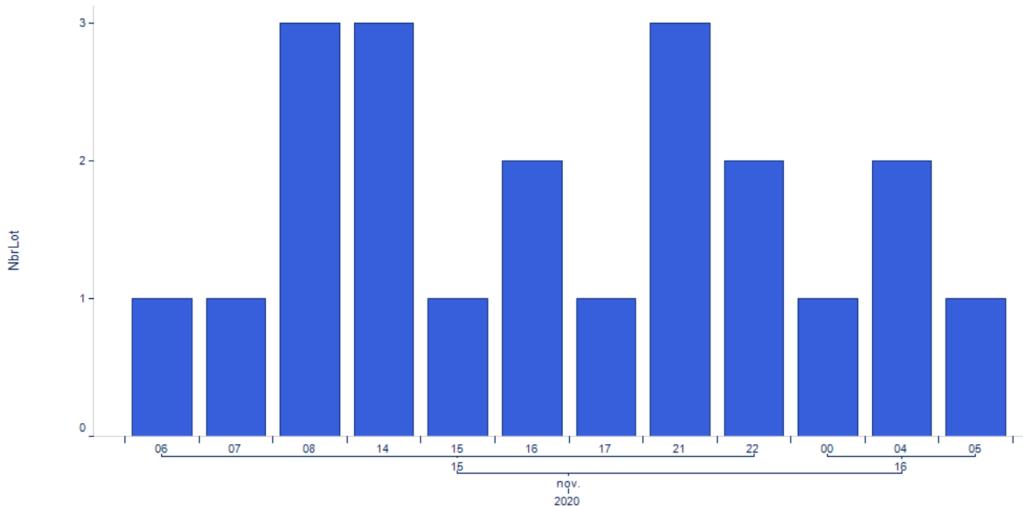


FIGURE 8.2 – Nombre de lots sortis de l'opération o par heure, données historiques

La méthode de transformation des données de l'usine en série de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ expliquée pour l'obtention des entrées u à partir de l'historique, est appliquée sur ce graphique pour obtenir la sortie $y_o \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$. Les valeurs des exposants de γ correspondent donc aux nombres de lots produits et les valeurs des exposants de δ aux heures de sortie de ces lots. Le premier monôme de la série est $\gamma^0\delta^1$ étant donné que le premier lot, numéroté 0, sort de l'opération à 6h, soit 1h après

2. Information différente du numéro du lot qui est l'exposant de γ du premier monôme (0).

le début de la journée étudiée (15 novembre à 5h) donc à la date 1. La série y_o obtenue est la série suivante :

$$y_o = \gamma^0 \delta^1 \oplus \gamma^1 \delta^2 \oplus \gamma^2 \delta^3 \oplus \gamma^5 \delta^9 \oplus \gamma^8 \delta^{10} \oplus \gamma^9 \delta^{11} \oplus \gamma^{11} \delta^{12} \oplus \gamma^{12} \delta^{16} \oplus \gamma^{15} \delta^{17} \oplus \gamma^{17} \delta^{19} \oplus \gamma^{18} \delta^{23} \oplus \gamma^{20} \delta^{24} \oplus \gamma^{21} \delta^{+\infty}.$$

8.3 Modélisation de l'opération

L'opération o étudiée dans l'exemple de ce chapitre est une opération constituée de deux étapes successives s_1 et s_2 . La modélisation par GET de cette opération se fait à partir des modèles définis dans la section 4.2.3 du chapitre 4. Ces modèles sont vérifiés grâce aux informations d'OPERA. A partir de l'entrée de planification u_{opera} , on calcule l'intervalle des sorties acceptables produites par ces GET $[y_v^{min}, y_v^{max}]$ et on vérifie si la sortie planifiée par OPERA y_{OPERA} (issue de l'entrée planifiée) est bien contenue dans cet intervalle. Enfin, les sorties y_s des modèles GET sont calculées à partir de l'entrée mesurée u (entrée de l'historique calculée dans la section 8.2.1) et connaissant les fonctions de transfert H des GET (H^{min} et H^{max}).

8.3.1 Modèles GET de l'opération

Commençons par les modèles GET de l'opération. Dans le chapitre 4, les GET 4.12 et 4.13, repris ci-dessous par les figures 8.3 et 8.4, représentent une étape d'une opération selon deux configurations temporelles différentes. Le premier prend en compte la durée moyenne minimale de l'étape CTMin. Ce modèle est celui que nous choisissons pour représenter le comportement le plus rapide de l'étape. Le second prend en compte le temps moyen d'attente CTWait et la durée moyenne de processus de l'étape CTTH. Ce fonctionnement est plus lent que le fonctionnement du premier modèle. Ce modèle est choisi pour représenter le comportement le plus lent de l'étape.

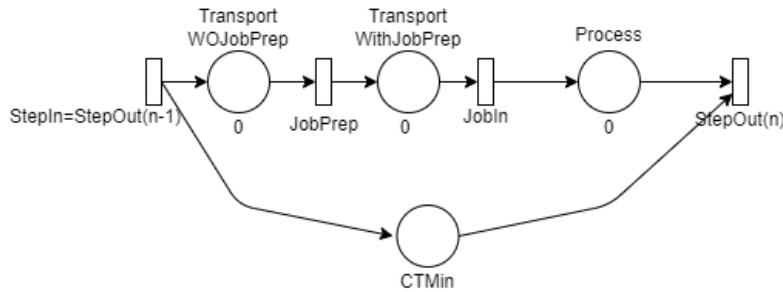


FIGURE 8.3 – GET d'une étape avec la durée CTMIN

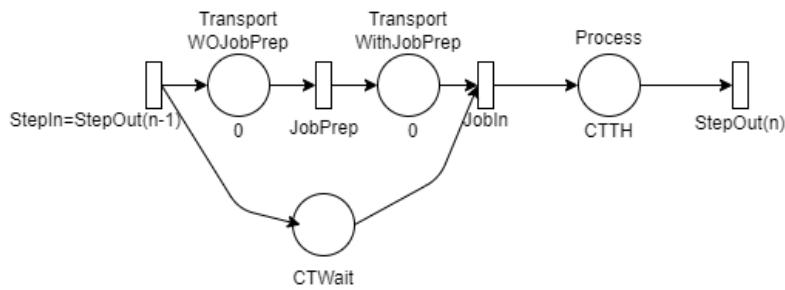
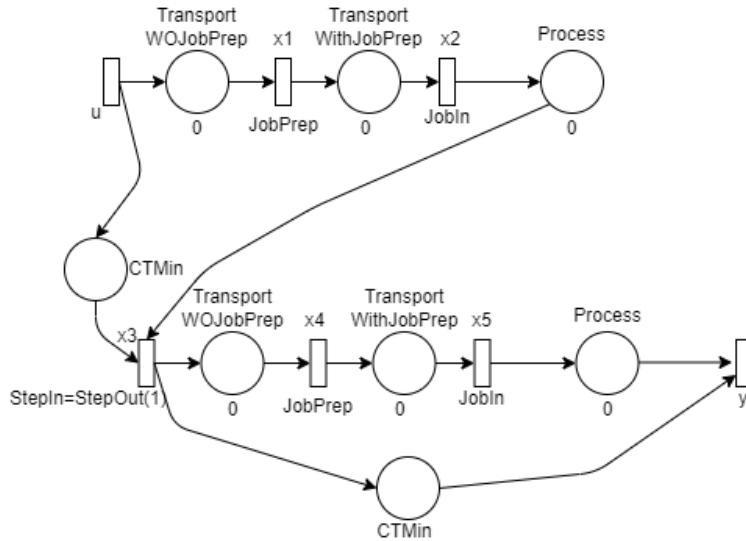
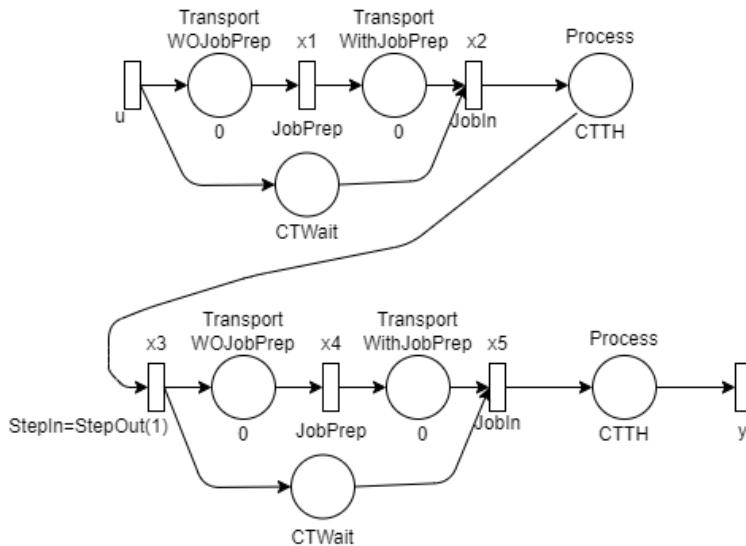


FIGURE 8.4 – GET d'une étape avec les durées CTWait et CTTH

L'opération o sera donc modélisée selon ces deux comportements temporels différents. Pour représenter la configuration des deux étapes s_1 et s_2 de l'opération avec la durée CTMin, on répète deux fois de suite le GET 8.3. Le GET obtenu est nommé GETmin et est représenté figure 8.5. L'entrée u est le StepIn de la première étape s_1 , sa sortie y est le StepOut de la seconde étape s_2 . Les transitions internes sont nommées par x_i . La transition StepOut de l'étape s_1 est fusionnée dans la transition x_3 avec la transition StepIn de l'étape s_2 .

FIGURE 8.5 – GETmin = GET de l'opération o avec les durées CTMIN

La même modélisation est faite pour la configuration avec les durées CTWait et CTTH. Le GET obtenu est nommé GETmax et est représenté sur la figure 8.6.

FIGURE 8.6 – GETmax = GET de l'opération o avec les durées CTWait et CTTH

D'après les données d'OPERA et les calculs réalisés sur ces données, les valeurs des durées CTMin, CTWait, CTTH des deux étapes s_1 et s_2 de l'opération o sont données en minutes dans le tableau 8.1. Si les séries présentées dans ce chapitre sont décrites en heures (u, y_o, \dots), elles sont traduites en minutes dans notre mise en oeuvre afin d'être cohérentes avec les modèles GET dont les durées sont des minutes.

Les matrices de transfert de ces GET 8.5 et 8.6 (de dimension (1×1) car une seule entrée et une seule sortie) sont notées respectivement H^{min} et H^{max} et ont pour valeur :

$$\begin{aligned} H^{min} &= (\gamma^0 \delta^{209}), \\ H^{max} &= (\gamma^0 \delta^{542}). \end{aligned} \quad (8.1)$$

8.3.2 Vérification de la modélisation

Nous vérifions ces modèles par les données de planification d'OPERA. L'idée est qu'à partir de l'entrée planifiée pour l'opération, nommée u_{OPERA} , l'intervalle des sorties acceptables des GET

	CTTH	CTWait	CTMin
s_1	30	202	76
s_2	53	257	133

TABLE 8.1 – Les différentes durées (en minutes) des étapes s_1 et s_2 de l'opération o

$[y_v^{min}, y_v^{max}]$ pour cette planification est calculée. Ensuite, la sortie planifiée, nommée y_{OPERA} , est comparée à ces sorties y_v^{min} et y_v^{max} afin de valider nos modèles GET.

Tout d'abord, la colonne StepOut du fichier LOTSPROJECTION qui fournit les dates planifiées de sortie d'une étape pour chaque lot est utilisée pour obtenir la série $u_{OPERA} \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$ (puisque la sortie d'une étape pour un lot correspond à son entrée dans l'étape suivante). Dans ce fichier, le premier lot planifié entre dans l'opération le 15 novembre à 12h, donc la date de ce premier lot, événement numéroté 0, est $7 = 12$ (heure de la sortie planifiée du premier lot) - 5 (début de la journée). La série obtenue par la méthode de transformation de la section 8.2.1 est :

$$u_{OPERA} = \gamma^0 \delta^7 \oplus \gamma^4 \delta^9 \oplus \gamma^5 \delta^{10} \oplus \gamma^6 \delta^{11} \oplus \gamma^8 \delta^{12} \oplus \gamma^{21} \delta^{+\infty}.$$

À partir des fonctions de transfert données en (8.1) et connaissant l'entrée u_{OPERA} , les sorties y_v^{min} et y_v^{max} correspondant aux sorties des modèles sans faute des GET des figures 8.5 et 8.6 sont calculées :

$$\begin{aligned} y_v^{min} &= H^{min} \otimes u_{OPERA} \\ &= \gamma^0 \delta^{10} \oplus \gamma^4 \delta^{12} \oplus \gamma^5 \delta^{13} \oplus \gamma^6 \delta^{14} \oplus \gamma^8 \delta^{15} \oplus \gamma^{21} \delta^{+\infty}, \\ y_v^{max} &= H^{max} \otimes u_{OPERA} \\ &= \gamma^0 \delta^{16} \oplus \gamma^4 \delta^{18} \oplus \gamma^5 \delta^{19} \oplus \gamma^6 \delta^{20} \oplus \gamma^8 \delta^{21} \oplus \gamma^{21} \delta^{+\infty}. \end{aligned}$$

Enfin, la colonne StepOut du fichier LOTSPROJECTION est à nouveau utilisée afin d'obtenir la série $y_{OPERA} \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$:

$$y_{OPERA} = \gamma^0 \delta^{12} \oplus \gamma^4 \delta^{14} \oplus \gamma^5 \delta^{15} \oplus \gamma^7 \delta^{16} \oplus \gamma^{21} \delta^{+\infty}.$$

Cette sortie planifiée est représentée en noir sur la figure 8.7. On constate qu'elle est bien contenue dans l'intervalle des sorties y_v^{min} et y_v^{max} ce qui valide la modélisation GET d'une opération donnée dans la section précédente.

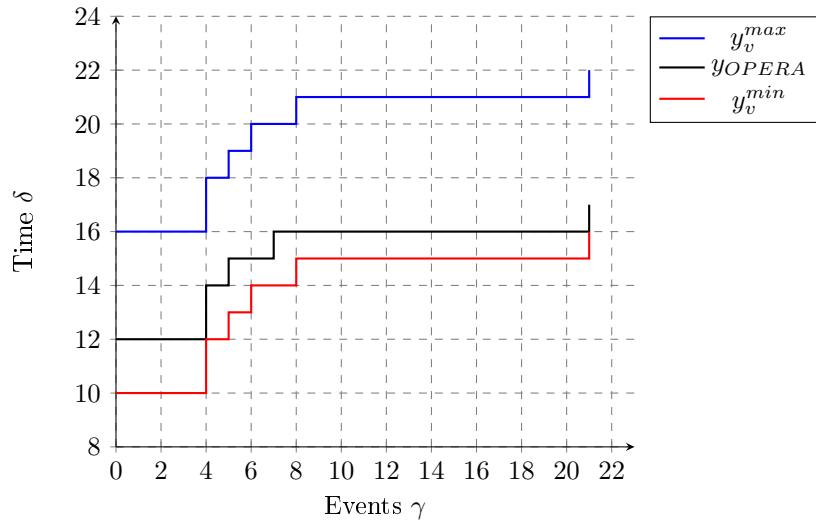


FIGURE 8.7 – Sortie y_{OPERA} de la planification encadrée par les sorties y_v^{min} et y_v^{max} des modèles sans faute d'une opération calculées avec u_{OPERA}

8.3.3 Calcul des sorties y_s des GET de l'opération avec l'entrée u

Les sorties y_s^{min} et y_s^{max} correspondant au comportement des GET 8.5 et 8.6 de notre modélisation sont maintenant calculées par rapport à l'entrée de l'historique u et connaissant H^{min} et

H^{max} (les fonctions de transfert des GET données en (8.1)) :

$$\begin{aligned} y_s^{min} &= H^{min} \otimes u \\ &= \gamma^0 \delta^3 \oplus \gamma^1 \delta^4 \oplus \gamma^3 \delta^6 \oplus \gamma^5 \delta^8 \oplus \gamma^7 \delta^{12} \oplus \gamma^9 \delta^{13} \oplus \gamma^{10} \delta^{14} \oplus \gamma^{11} \delta^{15} \oplus \\ &\quad \gamma^{12} \delta^{18} \oplus \gamma^{13} \delta^{20} \oplus \gamma^{14} \delta^{23} \oplus \gamma^{15} \delta^{24} \oplus \gamma^{19} \delta^{25} \oplus \gamma^{20} \delta^{26} \oplus \gamma^{21} \delta^{+\infty}, \end{aligned}$$

$$\begin{aligned} y_s^{max} &= H^{max} \otimes u \\ &= \gamma^0 \delta^9 \oplus \gamma^1 \delta^{10} \oplus \gamma^3 \delta^{12} \oplus \gamma^5 \delta^{14} \oplus \gamma^7 \delta^{18} \oplus \gamma^9 \delta^{19} \oplus \gamma^{10} \delta^{20} \oplus \gamma^{11} \delta^{21} \oplus \\ &\quad \gamma^{12} \delta^{24} \oplus \gamma^{13} \delta^{26} \oplus \gamma^{14} \delta^{29} \oplus \gamma^{15} \delta^{30} \oplus \gamma^{19} \delta^{31} \oplus \gamma^{20} \delta^{32} \oplus \gamma^{21} \delta^{+\infty}. \end{aligned}$$

La figure 8.8 représente ces sorties des modèles sans faute, rouge pour y_s^{min} et bleu pour y_s^{max} .

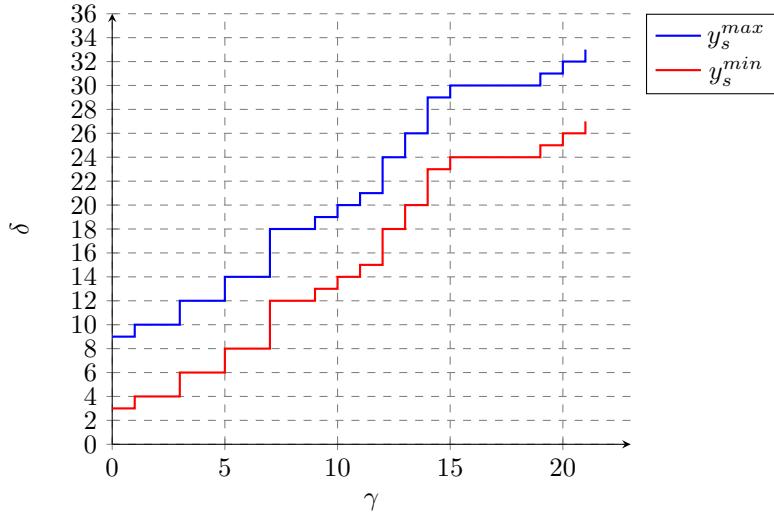


FIGURE 8.8 – Sorties y_s^{min} et y_s^{max} des modèles sans faute d'une opération

8.4 Diagnostic de dérives temporelles

Les sections précédentes viennent de définir les modèles GET utilisés pour représenter l'opération o , ainsi que la série de l'entrée u de ces GET issue de l'historique, la sortie mesurée y_o issue de l'historique ainsi que les sorties du modèle sans faute y_s^{min} et y_s^{max} . Cette section propose la mise en place de nos méthodes de diagnostic de dérives temporelles par intervalles du chapitre 7 et par observateur du chapitre 6.

8.4.1 Par intervalles

La première méthode de détection de dérives temporelles appliquée est celle utilisant les dioïdes et GET intervalles. Cette méthode requiert un GET intervalles modélisant des durées de tâches évoluant dans des intervalles dont on connaît les bornes. La sortie intervalle obtenue \mathbf{y}_s connaissant l'entrée u et la matrice de transfert du GET est comparée avec la sortie observée du système en fonctionnement y_o . Dans notre cas, nous n'avons pas obtenu un GET intervalles mais deux GET classiques représentant chacun un comportement extrême pour l'opération o étudiée. Pour l'entrée u issue de l'historique, la sortie calculée y_s^{min} du GET 8.5 choisi pour définir le comportement le plus rapide de l'opération est identifiée comme la borne minimale de la sortie intervalle $\mathbf{y}_s = [y_s, \bar{y}_s]$. La sortie calculée y_s^{max} du GET 8.6 choisi pour définir le comportement le plus lent de l'opération est identifiée comme sa borne maximale. Ainsi $\underline{y}_s = y_s^{min}$ et $\bar{y}_s = y_s^{max}$. Comme l'historique fournit une sortie y_o unique, nous utilisons la méthode de détection par indicateur avec sortie certaine de la section 7.4.1.

On calcule tout d'abord les distances temporelles du système (voir définition 7.9) entre $\underline{y}_s = y_s^{min}$ et $\bar{y}_s = y_s^{max}$. Les résiduations sur ces sorties donnent :

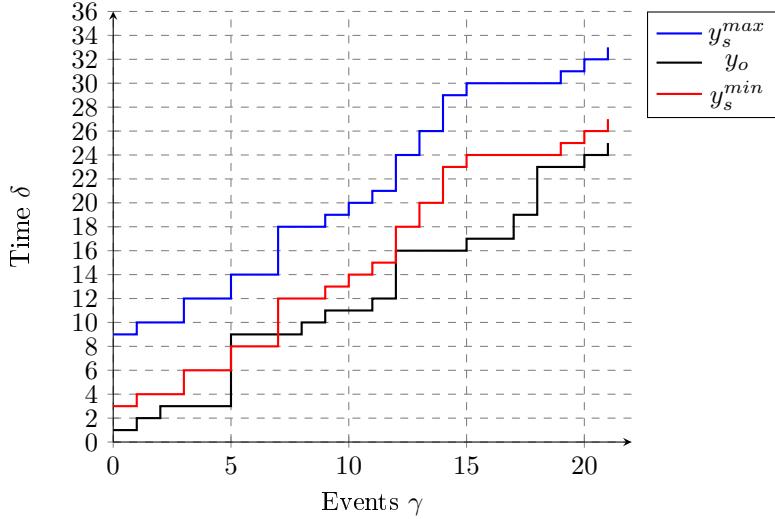


FIGURE 8.9 – Représentation de la sortie observée y_o non encadrée par les sorties y_s^{min} et y_s^{max} des modèles sans faute d'une opération

$$\begin{aligned}\bar{y}_s \neq y_s &= \gamma^0 \delta^6 \oplus \gamma^4 \delta^7 \oplus \gamma^5 \delta^8 \oplus \gamma^6 \delta^9 \oplus \gamma^7 \delta^{12} \oplus \gamma^8 \delta^{14} \oplus \gamma^9 \delta^{16} \oplus \gamma^{10} \delta^{17} \oplus \gamma^{11} \delta^{18} \oplus \gamma^{12} \delta^{19} \oplus \gamma^{13} \delta^{20} \oplus \gamma^{14} \delta^{23} \oplus \gamma^{15} \delta^{24} \oplus \gamma^{16} \delta^{25} \oplus \gamma^{17} \delta^{26} \oplus \gamma^{18} \delta^{27} \oplus \gamma^{19} \delta^{28} \oplus \gamma^{20} \delta^{29} \oplus \gamma^{21} \delta^{+\infty}, \\ \underline{y}_s \neq \bar{y}_s &= \gamma^0 \delta^{-6} \oplus \gamma^4 \delta^{-5} \oplus \gamma^5 \delta^{-4} \oplus \gamma^6 \delta^{-3} \oplus \gamma^7 \delta^0 \oplus \gamma^8 \delta^2 \oplus \gamma^9 \delta^4 \oplus \gamma^{10} \delta^5 \oplus \gamma^{11} \delta^6 \oplus \gamma^{12} \delta^7 \oplus \gamma^{13} \delta^8 \oplus \gamma^{14} \delta^{11} \oplus \gamma^{15} \delta^{12} \oplus \gamma^{16} \delta^{13} \oplus \gamma^{17} \delta^{14} \oplus \gamma^{18} \delta^{15} \oplus \gamma^{19} \delta^{16} \oplus \gamma^{20} \delta^{17} \oplus \gamma^{21} \delta^{+\infty}.\end{aligned}$$

Ainsi, les distances temporelles du système sont :

$$\begin{aligned}\Delta_{\bar{y}_s} &= [\mathcal{D}_{\bar{y}_s \neq \bar{y}_s}(0); -\mathcal{D}_{\bar{y}_s \neq \bar{y}_s}(0)] = [-6, -6] = [\underline{\Delta}_{\bar{y}}, \overline{\Delta}_{\bar{y}}], \\ \Delta_{\underline{y}_s} &= [\mathcal{D}_{\underline{y}_s \neq \underline{y}_s}(0); -\mathcal{D}_{\underline{y}_s \neq \underline{y}_s}(0)] = [6, 6] = [\underline{\Delta}_{\underline{y}}, \overline{\Delta}_{\underline{y}}].\end{aligned}$$

Les calculs des résiduations entre la sortie observée et les sorties des modèles sans faute donnent :

$$\begin{aligned}\bar{y}_s \neq y_o &= \gamma^0 \delta^5 \oplus \gamma^2 \delta^9 \oplus \gamma^4 \delta^{10} \oplus \gamma^5 \delta^{11} \oplus \gamma^6 \delta^{12} \oplus \gamma^7 \delta^{15} \oplus \gamma^8 \delta^{16} \oplus \gamma^9 \delta^{18} \oplus \gamma^{10} \delta^{19} \oplus \gamma^{11} \delta^{20} \oplus \gamma^{12} \delta^{21} \oplus \gamma^{14} \delta^{22} \oplus \gamma^{15} \delta^{23} \oplus \gamma^{16} \delta^{27} \oplus \gamma^{17} \delta^{28} \oplus \gamma^{18} \delta^{29} \oplus \gamma^{19} \delta^{30} \oplus \gamma^{20} \delta^{31} \oplus \gamma^{21} \delta^{+\infty}, \\ y_o \neq \underline{y}_s &= \gamma^0 \delta^{-7} \oplus \gamma^2 \delta^{-6} \oplus \gamma^3 \delta^{-4} \oplus \gamma^4 \delta^{-1} \oplus \gamma^5 \delta^0 \oplus \gamma^6 \delta^1 \oplus \gamma^7 \delta^4 \oplus \gamma^8 \delta^5 \oplus \gamma^{10} \delta^7 \oplus \gamma^{11} \delta^9 \oplus \gamma^{12} \delta^{11} \oplus \gamma^{14} \delta^{13} \oplus \gamma^{16} \delta^{14} \oplus \gamma^{17} \delta^{16} \oplus \gamma^{18} \delta^{19} \oplus \gamma^{19} \delta^{20} \oplus \gamma^{20} \delta^{21} \oplus \gamma^{21} \delta^{+\infty}.\end{aligned}$$

En reprenant l'indicateur de la définition 7.12, on obtient :

$$I(\mathbf{u}, y_o) = \text{vrai} \quad \text{car} \quad -\mathcal{D}_{\bar{y}_s \neq y_o}(0) = -5 \in [-6, 0] \quad \text{et} \quad \mathcal{D}_{y_o \neq \underline{y}_s}(0) = -7 \notin [0, 6].$$

La condition $\mathcal{D}_{\bar{y}_s \neq y_o}(0) = -5 \in [-6, 0]$ est vérifiée mais la condition $\mathcal{D}_{y_o \neq \underline{y}_s}(0) = -7 \notin [0, 6]$ ne l'est pas donc l'indicateur renvoie vrai, une dérive temporelle est détectée. La figure 8.9 représente les sorties $\underline{y}_s = y_s^{min}$ et $\bar{y}_s = y_s^{max}$, respectivement en rouge et en bleu, ainsi que la sortie y_o en noir.

Regardons maintenant les valeurs des conditions. La valeur -5 de $-\mathcal{D}_{\bar{y}_s \neq y_o}(0)$ indique que la série y_o est en-dessous de \bar{y}_s . Il n'y a donc pas de retard entre la sortie du modèle sans faute (\bar{y}_s) et l'historique (y_o). Tout est normal pour cette borne. Ensuite, la valeur -7 de $\mathcal{D}_{y_o \neq \underline{y}_s}(0)$ donne l'information que la série y_o est en-dessous de \underline{y}_s ce qui indique que la série y_o est en avance sur \underline{y}_s .

Avec ces informations, nous ne pouvons, a priori, pas mettre en place notre analyse de localisation par observateurs puisque les dérives de type avance temporelle ne sont pas caractérisées avec cette méthode. Cependant, l'observation de la figure 8.9 montre qu'il y a quelques lots en retard dans y_o par rapport à \underline{y}_s , comme par exemple le 5ème lot. Pour localiser cette dérive temporelle de type retard entre y_o et \underline{y}_s , on peut appliquer la méthode par observateur pour le modèle GETmin.

8.4.2 Par observateur

La section précédente a présenté l'utilisation de l'indicateur par intervalle qui a détecté une dérive temporelle dans l'opération o étudiée. Cette dérive est une avance de l'historique sur notre modèle sans faute mais l'analyse graphique des séries a montré que la sortie y_o était également en retard pour quelques lots par rapport à la sortie $\underline{y}_s = y_s^{\min}$. La méthode de localisation par observateur présentée dans le chapitre 6, méthode qui ne permet de manipuler que des dérives temporelles engendrant uniquement des retards dans le système, peut donc être appliquée sur le GET 8.5 qui décrit le fonctionnement considéré comme le plus rapide de l'opération et qui a engendré la sortie y_s^{\min} . L'entrée u utilisée dans cette méthode est à nouveau celle de l'historique décrite dans la section 8.2.1 et la fonction de transfert est H^{\min} .

La localisation de dérive par observateurs ($\max, +$) nécessite tout d'abord d'ajouter des entrées de perturbations sur tous les états, c'est-à-dire les transitions internes, du modèle. La matrice L de l'observateur est alors calculée :

$$L = \begin{pmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \gamma^0 \delta^{-133} \end{pmatrix}.$$

Elle donne comme information que seul l'état x_5 est estimé (seule ligne de la colonne différente de ε). En effet, d'après la proposition 6.1, les états x_1, x_2, x_3 et x_4 ne dépendent pas des perturbations. En continuant avec ce modèle et donc, cette matrice L , nous n'allons obtenir qu'un seul indicateur, celui de x_5 sur les 5 indicateurs possibles. La localisation ne sera donc pas très précise. Pour avoir des informations supplémentaires afin d'améliorer les estimations faites par l'observateur, il est possible d'ajouter deux nouvelles sorties à notre modèle GET : y_1 et y_2 . Ces sorties, reliées respectivement aux transitions x_1 et x_4 dans la figure 8.10, représentent l'information du JobPrep (affectation d'un lot sur un équipement) d'une étape issue des données de l'historique. Cette information n'est pas complètement sûre par rapport aux exigences de l'usine, c'est pourquoi elle n'était pas utilisée jusque là. En effet, les JobPrep sont des actions qui peuvent être annulées par les opérateurs, ce qui fait qu'un lot peut revenir dans sa situation d'avant ce JobPrep. Un second JobPrep peut donc être émis pour un même lot et ainsi, plusieurs événements JobPrep peuvent exister dans les fichiers de l'historique pour un lot identique. Cependant, nous avons pu constater que la présence de ces deux nouvelles sorties fournissent une matrice L plus complète et de ce fait, augmente le nombre d'états estimés. La localisation basée sur les indicateurs de ces états a donc plus de chance de donner une résultat intéressant, malgré l'incertitude apportée.

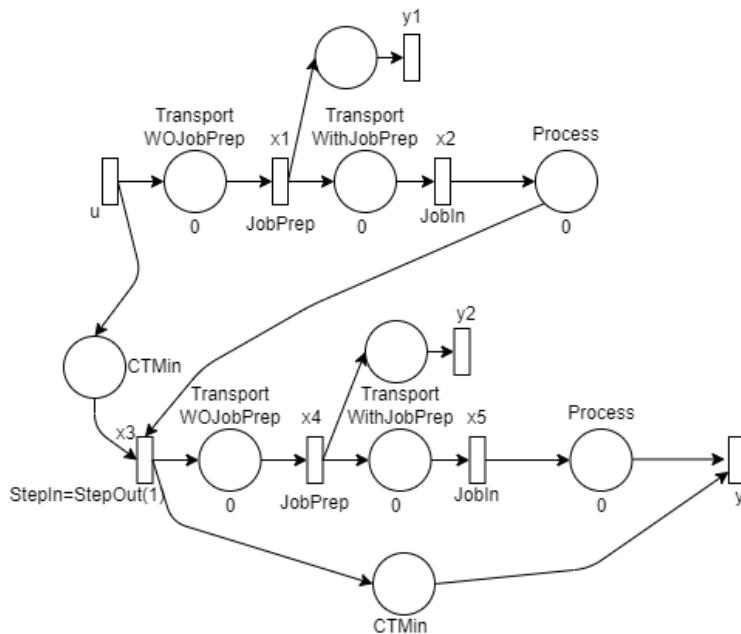


FIGURE 8.10 – GET de l'opération o avec des capteurs sur les transitions x_1, x_4

La colonne JobPrep du fichier JOB_PREP de l'historique contient les dates de cette action pour chaque lot. En appliquant à nouveau la méthode de transformation des données de l'usine présentée en section 8.2.1, les séries des sorties y_{o1} et y_{o2} sont calculées :

$$y_{o1} = \gamma^0\delta^0 \oplus \gamma^1\delta^1 \oplus \gamma^3\delta^3 \oplus \gamma^5\delta^5 \oplus \gamma^7\delta^9 \oplus \gamma^9\delta^{10} \oplus \gamma^{10}\delta^{11} \oplus \gamma^{11}\delta^{12} \oplus \gamma^{12}\delta^{13} \oplus \gamma^{13}\delta^{15} \oplus \gamma^{14}\delta^{17} \oplus \gamma^{15}\delta^{21} \oplus \gamma^{17}\delta^{23} \oplus \gamma^{18}\delta^{24} \oplus \gamma^{19}\delta^{25} \oplus \gamma^{21}\delta^{+\infty}$$

et

$$y_{o2} = \gamma^0\delta^0 \oplus \gamma^1\delta^1 \oplus \gamma^4\delta^8 \oplus \gamma^8\delta^9 \oplus \gamma^9\delta^{10} \oplus \gamma^{11}\delta^{14} \oplus \gamma^{12}\delta^{15} \oplus \gamma^{15}\delta^{18} \oplus \gamma^{16}\delta^{21} \oplus \gamma^{17}\delta^{22} \oplus \gamma^{18}\delta^{25} \oplus \gamma^{20}\delta^{26} \oplus \gamma^{21}\delta^{+\infty}.$$

L'observateur mis en place pour le GET 8.10, pour lequel tous les états sont perturbés, donne la matrice L suivante :

$$L = \begin{pmatrix} \gamma^0\delta^0 & \cdot & \cdot \\ \gamma^0\delta^0 & \cdot & \cdot \\ \gamma^0\delta^0 & \cdot & \cdot \\ \gamma^0\delta^0 & \gamma^0\delta^0 & \cdot \\ \gamma^0\delta^0 & \gamma^0\delta^0 & \gamma^0\delta^{-133} \end{pmatrix}.$$

D'après cette matrice, tous les états du modèle dépendent des perturbations. Le calcul de CA^*R est :

$$CA^*R = \begin{pmatrix} \gamma^0\delta^0 & \cdot & \cdot & \cdot & \cdot \\ \gamma^0\delta^0 & \gamma^0\delta^0 & \gamma^0\delta^0 & \gamma^0\delta^0 & \cdot \\ \gamma^0\delta^{133} & \gamma^0\delta^{133} & \gamma^0\delta^{133} & \gamma^0\delta^{133} & \gamma^0\delta^{133} \end{pmatrix},$$

ce qui permet d'obtenir le tableau 8.2 indiquant le ν_d de chaque état estimé.

Transition	N_i	ν_0^b	$\nu_0^{w_i}$	ν_d
x_1	0	0	0	0
x_2	0	0	0	0
x_3	0	0	0	0
x_4	0	0	0	0
x_5	0	0	0	0

TABLE 8.2 – Tableau récapitulatif des ν_d par états estimés

Avec l'entrée u de l'historique (même entrée que celle utilisée pour la détection par intervalles, elle est calculée dans la section 8.2.1), on obtient les états estimés SAP suivants :

$$x_{ed} = \begin{pmatrix} \gamma^0\delta^0 \oplus \gamma^1\delta^1 \oplus \gamma^3\delta^3 \oplus \gamma^5\delta^5 \oplus \gamma^7\delta^9 \oplus \gamma^9\delta^{10} \oplus \gamma^{10}\delta^{11} \oplus \gamma^{11}\delta^{12} \oplus \gamma^{12}\delta^{15} \oplus \gamma^{13}\delta^{17} \oplus \gamma^{14}\delta^{20} \oplus \gamma^{15}\delta^{21} \oplus \gamma^{19}\delta^{22} \oplus \gamma^{20}\delta^{23} \oplus \gamma^{21}\delta^{+\infty} \\ \gamma^0\delta^0 \oplus \gamma^1\delta^1 \oplus \gamma^3\delta^3 \oplus \gamma^5\delta^5 \oplus \gamma^7\delta^9 \oplus \gamma^9\delta^{10} \oplus \gamma^{10}\delta^{11} \oplus \gamma^{11}\delta^{12} \oplus \gamma^{12}\delta^{15} \oplus \gamma^{13}\delta^{17} \oplus \gamma^{14}\delta^{20} \oplus \gamma^{15}\delta^{21} \oplus \gamma^{19}\delta^{22} \oplus \gamma^{20}\delta^{23} \oplus \gamma^{21}\delta^{+\infty} \\ \gamma^0\delta^0 \oplus \gamma^1\delta^2 \oplus \gamma^3\delta^4 \oplus \gamma^5\delta^6 \oplus \gamma^7\delta^{10} \oplus \gamma^9\delta^{11} \oplus \gamma^{10}\delta^{12} \oplus \gamma^{11}\delta^{13} \oplus \gamma^{12}\delta^{16} \oplus \gamma^{13}\delta^{18} \oplus \gamma^{14}\delta^{21} \oplus \gamma^{15}\delta^{22} \oplus \gamma^{19}\delta^{23} \oplus \gamma^{20}\delta^{24} \oplus \gamma^{21}\delta^{+\infty} \\ \gamma^0\delta^0 \oplus \gamma^1\delta^2 \oplus \gamma^3\delta^4 \oplus \gamma^5\delta^6 \oplus \gamma^7\delta^{10} \oplus \gamma^9\delta^{11} \oplus \gamma^{10}\delta^{12} \oplus \gamma^{11}\delta^{13} \oplus \gamma^{12}\delta^{16} \oplus \gamma^{13}\delta^{18} \oplus \gamma^{14}\delta^{21} \oplus \gamma^{15}\delta^{22} \oplus \gamma^{19}\delta^{23} \oplus \gamma^{20}\delta^{24} \oplus \gamma^{21}\delta^{+\infty} \\ \gamma^0\delta^0 \oplus \gamma^1\delta^2 \oplus \gamma^3\delta^4 \oplus \gamma^5\delta^6 \oplus \gamma^7\delta^{10} \oplus \gamma^9\delta^{11} \oplus \gamma^{10}\delta^{12} \oplus \gamma^{11}\delta^{13} \oplus \gamma^{12}\delta^{16} \oplus \gamma^{13}\delta^{18} \oplus \gamma^{14}\delta^{21} \oplus \gamma^{15}\delta^{22} \oplus \gamma^{19}\delta^{23} \oplus \gamma^{20}\delta^{24} \oplus \gamma^{21}\delta^{+\infty} \end{pmatrix}.$$

Les états calculés à partir du GET 8.10 sont :

$$x_s = \left(\begin{array}{l} \gamma^0 \delta^0 \oplus \gamma^1 \delta^1 \oplus \gamma^3 \delta^3 \oplus \gamma^5 \delta^5 \oplus \gamma^7 \delta^9 \oplus \gamma^9 \delta^{10} \oplus \gamma^{10} \delta^{11} \oplus \gamma^{11} \delta^{12} \oplus \gamma^{12} \delta^{15} \oplus \gamma^{13} \delta^{17} \oplus \gamma^{14} \delta^{20} \oplus \gamma^{15} \delta^{21} \oplus \gamma^{17} \delta^{23} \oplus \gamma^{18} \delta^{24} \oplus \gamma^{19} \delta^{25} \oplus \gamma^{21} \delta^{+\infty} \\ \gamma^0 \delta^0 \oplus \gamma^1 \delta^1 \oplus \gamma^3 \delta^3 \oplus \gamma^5 \delta^5 \oplus \gamma^7 \delta^9 \oplus \gamma^9 \delta^{10} \oplus \gamma^{10} \delta^{11} \oplus \gamma^{11} \delta^{12} \oplus \gamma^{12} \delta^{15} \oplus \gamma^{13} \delta^{17} \oplus \gamma^{14} \delta^{20} \oplus \gamma^{15} \delta^{21} \oplus \gamma^{17} \delta^{23} \oplus \gamma^{18} \delta^{24} \oplus \gamma^{19} \delta^{25} \oplus \gamma^{21} \delta^{+\infty} \\ \gamma^0 \delta^0 \oplus \gamma^1 \delta^2 \oplus \gamma^3 \delta^4 \oplus \gamma^5 \delta^6 \oplus \gamma^7 \delta^{10} \oplus \gamma^9 \delta^{11} \oplus \gamma^{10} \delta^{12} \oplus \gamma^{11} \delta^{13} \oplus \gamma^{12} \delta^{16} \oplus \gamma^{13} \delta^{18} \oplus \gamma^{14} \delta^{21} \oplus \gamma^{15} \delta^{22} \oplus \gamma^{17} \delta^{23} \oplus \gamma^{18} \delta^{24} \oplus \gamma^{19} \delta^{25} \oplus \gamma^{21} \delta^{+\infty} \\ \gamma^0 \delta^0 \oplus \gamma^1 \delta^2 \oplus \gamma^3 \delta^4 \oplus \gamma^4 \delta^8 \oplus \gamma^7 \delta^{10} \oplus \gamma^9 \delta^{11} \oplus \gamma^{10} \delta^{12} \oplus \gamma^{11} \delta^{14} \oplus \gamma^{12} \delta^{16} \oplus \gamma^{13} \delta^{18} \oplus \gamma^{14} \delta^{21} \oplus \gamma^{15} \delta^{22} \oplus \gamma^{17} \delta^{23} \oplus \gamma^{18} \delta^{25} \oplus \gamma^{20} \delta^{26} \oplus \gamma^{21} \delta^{+\infty} \\ \gamma^0 \delta^0 \oplus \gamma^1 \delta^2 \oplus \gamma^3 \delta^4 \oplus \gamma^4 \delta^8 \oplus \gamma^7 \delta^{10} \oplus \gamma^9 \delta^{11} \oplus \gamma^{10} \delta^{12} \oplus \gamma^{11} \delta^{14} \oplus \gamma^{12} \delta^{16} \oplus \gamma^{13} \delta^{18} \oplus \gamma^{14} \delta^{21} \oplus \gamma^{15} \delta^{22} \oplus \gamma^{17} \delta^{23} \oplus \gamma^{18} \delta^{25} \oplus \gamma^{20} \delta^{26} \oplus \gamma^{21} \delta^{+\infty} \end{array} \right).$$

Quand on compare les états du modèle sans faute et les états estimés SAP, on obtient les indicateurs suivants :

$$\begin{array}{llll} I'_{x_1}(u, y_o) & = & \text{vrai} & \text{car } \Delta(x_{ed1}, x'_{s1}) = [0, 3], \\ I'_{x_2}(u, y_o) & = & \text{vrai} & \text{car } \Delta(x_{ed2}, x'_{s2}) = [0, 3], \\ I'_{x_3}(u, y_o) & = & \text{vrai} & \text{car } \Delta(x_{ed3}, x'_{s3}) = [0, 2], \\ I'_{x_4}(u, y_o) & = & \text{vrai} & \text{car } \Delta(x_{ed4}, x'_{s4}) = [0, 4], \\ I'_{x_5}(u, y_o) & = & \text{vrai} & \text{car } \Delta(x_{ed5}, x'_{s5}) = [0, 4]. \end{array}$$

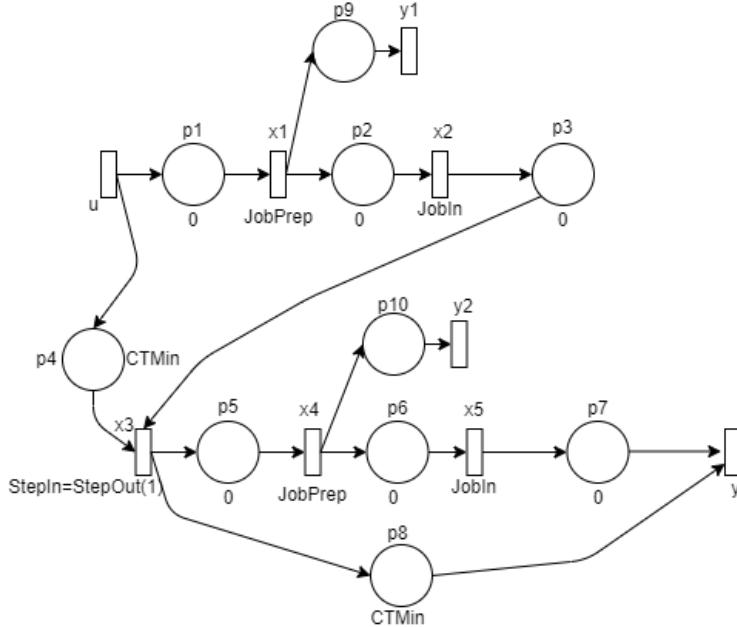


FIGURE 8.11 – GET du comportement le plus rapide de l'opération o avec une numérotation p_i des noms des places

Une dérive temporelle est donc détectée. La question est maintenant de savoir où se situe cette dérive. Pour cela, le GET de la figure 8.10 est repris figure 8.11 avec une numérotation p_i des noms des places. Le tableau 8.3 récapitule la décomposition de ce GET en structures élémentaires.

Rappelons tout d'abord que lors de notre décomposition, nous ne prenons pas en compte les transitions d'entrées ni celles de sorties car nos indicateurs sont calculés uniquement pour les états, donc les transitions internes. Le parallélisme issu de l'entrée u n'apparaît donc pas dans le tableau. De même pour la synchronisation entre l'entrée u et la transition x_1 , etc.

Ensuite, prenons par exemple la structure tandem entre les places p_2 et p_3 de la ligne 2. Les états impliqués dans cette structure sont les états x_2 et x_3 , or les valeurs de l'intervalle associé à

	Trans. et Pl.	Structure	Intervalles
1	x_1, x_2 p_1, p_2	Tandem	$\Delta(x_{ed1}, x_{s1}) = [0, 3]$ $\Delta(x_{ed2}, x_{s2}) = [0, 3]$
2	x_2, x_3 p_1, p_2, p_4, p_3	Tandem	$\Delta(x_{ed2}, x_{s2}) = [0, 3]$ $\Delta(x_{ed3}, x_{s3}) = [0, 2]$
3	x_3, x_4 $p_1, p_2, p_4, p_3,$	Tandem	$\Delta(x_{ed3}, x_{s3}) = [0, 2]$ $\Delta(x_{ed4}, x_{s4}) = [0, 4]$
4	x_4, x_5 $p_1, p_2, p_4, p_3, p_5, p_6$	Tandem	$\Delta(x_{ed4}, x_{s4}) = [0, 4]$ $\Delta(x_{ed5}, x_{s5}) = [0, 4]$

TABLE 8.3 – Résumé des différentes structures du GET 8.11

l'état x_3 sont différentes de celles de l'intervalle associé à x_2 (différence qui n'a pas lieu pour le tandem entre p_1 et p_2 de la ligne 1). Cela ne devrait pas arriver dans une structure tandem isolée où l'effet d'une dérive localisée en début de tandem se propage simplement (sans modification) sur les places et transitions en aval. Les résultats ne sont pas cohérents par rapport aux résultats par structure du chapitre 6. Cette différence pourrait s'expliquer par le fait que x_3 est également une transition de synchronisation entre l'entrée u et la transition x_2 , synchronisation que nous ne prenons pas encore en compte dans nos analyses de localisation. Une prise en compte plus complète des transitions d'entrée et de sortie dans l'analyse des intervalles pour la localisation pourrait peut-être apporter une réponse aux questions soulevées par ces résultats.

D'autre part, d'après le tableau, l'analyse des valeurs des intervalles pour chaque structure ainsi que le croisement des données pour le GET général montrent des incohérences dans les résultats qui expliquent que nous ne pouvons malheureusement pas conclure sur la localisation de la dérive. En effet, dans les résultats que nous obtenons **ligne 3**, les valeurs nous font penser que la dérive temporelle est sur la place p_5 du fait de la structure de tandem et des valeurs des intervalles associés à x_3 et x_4 (la borne supérieure de $\Delta(x_{ed3}, x_{s3})$ est plus grande que la borne supérieure de $\Delta(x_{ed4}, x_{s4})$). Cependant, un résultat d'intervalles avec des bornes de valeurs différentes (ce qui est le cas de tous les intervalles obtenus) n'est pas cohérent avec l'analyse que l'on effectue sur la localisation de dérive dans un tandem. Sans pouvoir expliquer pourquoi les intervalles ont tous 0 comme borne minimale, nous supposons que cela est du au fait que la plupart du temps la sortie y_o est en avance par rapport à la sortie du modèle sans faute y_s^{min} . Cette avance n'est pas détectée par notre observateur du fait de son fonctionnement. Au vu de la figure 8.9 on remarque qu'il y a alternance entre avance et retard entre y_o et y_s^{min} . Cela peut être la conséquence d'une dérive intermittente alors que les dérives que nous avons étudiées dans ce manuscrit sont des dérives permanentes.

8.5 Synthèse du chapitre

Ce chapitre présente la détection de dérives temporelles par intervalles et par observateur sur une opération o avec des données réelles de l'usine. Pour effectuer cela, on présente une méthode qui transforme les informations de l'historique de l'usine en série de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$: l'entrée u de l'historique qui va servir d'entrée pour nos modèles; la sortie observée y_o issue de l'historique. Après avoir montré comment obtenir les séries en entrée et en sortie, on définit le modèle que l'on utilise pour faire la comparaison entre la planification et l'historique.

Dans notre cas, on modélise une opération avec plusieurs étapes. Le chapitre vérifie que le modèle correspond bien à la modélisation de la planification. Puis quand la modélisation est vérifiée on compare les sorties de cette modélisation y_s avec la sortie historique y_o . Pour pouvoir utiliser notre méthode de détection par intervalle on définit non pas un GET par intervalle mais deux comportements de l'opération, l'un pour la borne maximale et l'autre pour la borne minimale. Après avoir défini nos sorties y_s on utilise la méthode de détection de dérives temporelles par intervalles pour détecter une dérive temporelle dans cette opération. Cette analyse peut se faire sur une route en générale en fonction de la capacité des opérations de la route pour isoler les opérations qui sont en retard.

Pour aller plus loin dans l'analyse de notre détection, on met en place la localisation des dérives temporelles. Pour cela, on définit un observateur de cette opération sur le comportement le plus rapide. La mise en place de notre observation détecte une dérive temporelle dans le modèle mais la

conclusion sur la localisation n'a pas abouti du fait de l'incohérence dans les valeurs des intervalles. Cependant des pistes pour améliorer cette localisation sont proposées.

Conclusions et perspectives

Conclusions

Ces travaux présentent le problème de détection des dérives temporelles pour le pilotage de flux de production dans l'industrie des semiconducteurs. La détection s'effectue dans les usines de production de type *front end*. Les chaînes de production dans ce type d'usine font partie des plus complexes dans le monde. La gestion des flux devient de plus en plus importante dans leur pilotage pour maximiser la production. Du fait de leur complexité et de leur taille toujours plus grande, la détection des dérives temporelles est un enjeu fondamental pour livrer les produits à temps et ainsi atteindre les objectifs de production de façon optimale. L'objectif est de détecter les dérives temporelles au plus tôt pour limiter l'impact des goulets d'étranglement qui limitent fortement la production et l'exploitation maximale de la capacité des équipements. Pour effectuer la gestion des flux, des méthodes sont utilisées dans l'usine mais sont souvent centrées sur les ateliers ou les technologies. Une méthode générale sur l'ensemble de l'usine n'est pas mise en place. Ces travaux proposent des moyens qui permettent d'effectuer une analyse plus générale des flux dans l'usine.

Dans le chapitre 2, nous avons dans un premier temps proposé une présentation de l'outil de planification OPERA et une analyse de la performance de cet outil suivant les différents paramètres du logiciel. L'analyse des performances de ces paramètres de la planification montre qu'il n'existe pas un paramétrage optimal dans l'utilisation de l'outil dans l'usine Crolles 300 du fait de la forte variabilité des entrées de OPERA (capacités équipements, position des en-cours,...). Dans un second temps, nous avons mis en évidence des problèmes liés à la variabilité dans les usines de semiconducteurs, et avons montré que la forte variabilité de l'usine Crolles 300 ne permettait pas de faire une analyse globale sur l'ensemble de l'usine ou sur des ateliers pour détecter des retards. Du fait de cette impossibilité, on met en place une comparaison systématique entre la planification et la production au niveau des opérations. Cette comparaison exploite une fonction cumulative qui permet de définir un retard entre la planification et l'historique. On propose alors un indicateur qui calcule l'impact du retard accumulé. Cet indicateur met également en évidence *quand* le retard n'est plus structurellement rattrapable. Ce type de retard est défini comme un retard *critique*.

La chapitre 4 présente notre méthode de diagnostic mise en place dans ces travaux. Les retards identifiés dans le chapitre 2 nous permettent de définir le type de défaillance que l'on va étudier. Nous définissons dans ce chapitre formellement le problème étudié. Une définition des dérives temporelles par rapport au retard entre la production et la planification est proposée, ainsi que la définition du modèle GET utilisé pour faire la modélisation. La deuxième partie du chapitre décrit le problème que l'on veut modéliser, c'est-à-dire les dérives temporelles au niveau des étapes de fabrication des semiconducteurs. On décrit les notions mathématiques et les notions temporelles utilisées pour caractériser notre problème. Avec cette description nous proposons une modélisation d'une étape de fabrication d'un produit dans la planification de OPERA en GET pour pouvoir mettre en place notre diagnostic en utilisant l'algèbre $(\max, +)$.

Afin résoudre le problème formel défini dans le chapitre 4, une méthode de détection et une analyse de localisation utilisant l'algèbre $(\max, +)$ sont proposées dans le chapitre 6. Cette méthode exploite les systèmes $(\max, +)$ -linéaires avec observateurs et s'appuie sur la théorie des dioïdes et en particulier le dioïde $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ qui est présenté dans le chapitre 5. Nous définissons une caractérisation des dérives temporelles comme des perturbations afin de mettre en place un observateur et d'effectuer la détection. Nous avons ainsi défini un premier indicateur des dérives temporelles basé sur les états estimés. Pour améliorer la détection des dérives temporelles sur ces états, on met ensuite en place un affinage des états pour ne prendre en compte que les perturbations. Avec les informations obtenues par les indicateurs affinés on propose une analyse de localisation d'une dérive temporelle s'appuyant sur les structures élémentaires dans un GET.

Dans le Chapitre 7 nous proposons une autre méthode de détection des dérives temporelles dans les systèmes (max, +)-linéaires à temps intervalles. Les systèmes sont représentés par des GET intervalles sur les places, cela définit les variations des durées dans les opérations. Pour effectuer la détection nous introduisons la notion de sorties acceptables caractérisant ces systèmes. Les sorties acceptables permettent de définir un indicateur de dérives temporelles pour une sortie certaine (observation exacte), et un indicateur pour une sortie incertaine (intervalle d'observations). L'indicateur de dérives temporelles pour une sortie certaine est utilisé pour la détection de dérives temporelles sur des problèmes modélisés dans le chapitre 4.

Dans le chapitre 8 on met en place nos différents modèles définis dans le chapitre 4 en utilisant des données réelles sur les produits (la durée de processus, temps d'attente et durée minimale d'une étape). Pour pouvoir appliquer notre méthode sur des données réelles, nous présentons un moyen systématique de transformer les fichiers de données de la planification en série $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ afin d'obtenir des entrées compatibles avec les modèles établis. La même transformation est effectuée sur les fichiers de données de l'historique en série $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ afin d'obtenir les sorties compatibles avec les indicateurs que nous avons développés. Ce chapitre illustre la méthode décrite dans le chapitre 7 en utilisant les deux fonctionnements d'une étape. Le fonctionnement avec le temps minimal d'une étape est pris comme comportement rapide et le fonctionnement avec le temps d'attente et le temps de processus comme comportement lent. Enfin, pour affiner l'analyse de la détection, la méthode de détection par observateur définie dans le chapitre 6 est utilisée afin de localiser les sources potentielles des dérives temporelles.

Perspectives

Ces travaux ont donné lieu à plusieurs contributions scientifiques et industrielles. Les résultats donnent à réfléchir à plusieurs perspectives.

Amélioration des approches (max, +) pour la détection et la localisation des dérives temporelles

Dans un premiers temps, pour les méthodes de détection utilisant les observateurs du chapitre 6 et les intervalles du chapitre 7 plusieurs améliorations peuvent être prises en compte. L'une des premières améliorations possibles concerne la localisation avec l'automatisation de la décomposition des GET généraux en structure élémentaire. Pour effectuer cette amélioration, une identification des structures élémentaires par rapport à la matrice A du GET général peut être envisagée. Une autre amélioration possible concerne la localisation de la dérive temporelle. Cette localisation s'effectue seulement sur une seule dérive dans le système. Pour améliorer la localisation on pourrait émettre les hypothèses qu'il y a plusieurs dérives dans le système pour indiquer les places mises en cause par les dérives.

Dans le cadre de cette thèse, nous nous sommes limités à l'étude de dérives temporelles qui sont caractérisées par des modifications d'exposant sur δ dans un monôme $\gamma^i \delta^j$. Une possibilité serait de mettre en place la méthode de détection mais au niveau des événements γ . On observerait alors un nombre de plaques produites pour un temps donné. Un tel indicateur par événement γ ne peut pas utiliser les observateurs de la même manière. En effet, pour trouver les états estimés SAP avec l'incertitude dans les événements il faut modifier le calcul du ν_d . Pour la détection par intervalles, l'indicateur doit aussi être modifié en définissant les sorties acceptables dans les systèmes par événements.

De plus, nous avons défini au chapitre 8 un indicateur par intervalle pour détecter les dérives temporelles sur une opération particulière d'une technologie. Il est tout à fait possible d'automatiser cette méthode sur l'indicateur en le calculant sur toute la route d'une technologie. Ainsi, l'indicateur sur la route renverrait seulement la valeur des indicateurs positifs ce qui conduirait à une première localisation au niveau des opérations et on mettrait en place notre méthode de localisation des dérives par observateur sur ces opérations pour un diagnostic plus fin.

Enfin, la méthode que nous proposons est hors-ligne. On pourrait envisager de mettre en place notre méthode de détection en-ligne. La correction des dérives temporelles serait ainsi plus rapide et en lien direct avec la chaîne de production pour alerter des retards au plus tôt et être plus réactif afin d'éviter l'accumulation des retards.

Vers une meilleure intégration des indicateurs dans la supervision de la chaîne de production

Nous avons défini au chapitre 2 des indicateurs pour mesurer les retards sur une opération particulière d'une technologie. Il est tout à fait possible d'automatiser cette méthode sur l'indicateur en le calculant sur toute la route d'une technologie. Ainsi, on renvoie seulement la valeur des indicateurs négatifs et on affiche la ou les opérations qui posent problème si besoin. Cette automatisation s'appuie sur l'algorithme 2 qui calcule l'indicateur sur les retards critiques sur chaque étape de la route et les compare pour retourner en particulier quelles sont les premières étapes en retard critique sur la route et quelles sont les étapes dont le retard critique est maximal.

Au niveau de la modélisation des opérations, on peut en automatiser la construction en utilisant les données d'entrées OPERA avec les temps associés aux étapes. Avec l'automatisation de la modélisation on pourrait plus facilement utiliser la méthode de diagnostic définie dans le chapitre 8.

Notre démarche a toujours consisté à étudier les dérives au niveau d'une technologie donnée. Mais le résultat de cette analyse, technologie par technologie, pourrait être exploité pour comprendre pourquoi une technologie est en avance et quelles autres technologies sont impactées par cette avance, pourquoi le retard accumulé sur une technologie dans l'usine deviendrait problématique et impliquerait un retard sur d'autres technologies dans la production.

Enfin, une autre idée est de mettre en place une méthode de détection par observateur sur des retards critiques introduits dans le chapitre 2. L'analyse de la localisation mise en place permet de connaître la valeur du retard maximal θ , alors qu'une analyse sur les événements γ permettrait d'avoir une information sur le nombre maximal de lots en retard. Avec ces deux informations le calcul du travail du retard de l'étape est possible.

Liste des publications

- Détection par GET intervalle

Detection of time shift failures in (max, +)-linear systems with time intervals for assembly line. *C. Paya, E. Le Corronc, Y. Pencolé.* IEEE Int Conf on Systems, Man, and Cybernetics (SMC). 2019

- Détection par observateur (max,+)

Observer-based detection of time shift failures in (max, +)-linear systems. *C. Paya, E. Le Corronc, Y. Pencolé, P. Vialletelle.* Int Workshop on Principles of Diagnosis (DX). 2020

Model-based diagnosis of time shift failures in DES : a (max, +) observer-based approach. *C. Paya, E. Le Corronc, Y. Pencolé, P. Vialletelle.* Int KES Conf on Intelligent Decision Technologies (KES-IDT). 2021

- Localisation par observateur (max,+)

Observer-based detection and localization of time shift failures in (max, +)-linear systems. *C. Paya, E. Le Corronc, Y. Pencolé, P. Vialletelle.* IEEE Int Conf on Automation Science and Engineering (CASE). 2021

Détection et localisation de dérives temporelles à base d'observateurs (max, +) dans les GET. *C. Paya, E. Le Corronc, Y. Pencolé, P. Vialletelle.* Colloque sur la Modélisation des Systèmes Réactifs (MSR). 2021

- Autres publications

Localization of time shift failures in (max, +)-linear systems. *E. Le Corronc, A. Sahuguède, Y. Pencolé, C. Paya.* Workshop on Discrete Event Systems (WODES). 2018

Failure detection and localization for timed event graph in (max, +)-algebra. *E. Le Corronc, Y. Pencolé, A. Sahuguède, C. Paya.* Journal of Discrete Event Dynamic Systems (JDEDS). 2021

Bibliographie

- [AD21] A-A. Ahmed and P. David. Fault diagnosis in labelled petri nets : A fourier–motzkin based approach. *Automatica*, 132 :109831, 2021. [46](#)
- [ALF78] J-M. Arnaudiès and J. Lelong-Ferrand. *Cours de mathématiques, tome 1 (algebre)*. Dunod, 1978. [65](#)
- [Ant65] R. N. Anthony. *Planning and control systems : A framework for analysis*. Division of Research, Graduate School of Business Administration, Harvard, 1965. [12](#)
- [BAEMB19] I. B. Brahim, S. A. Addouche, A. El Mhamedi, and Y. Boujelbene. Build a bayesian network from fmeca in the production of automotive parts : diagnosis and prediction. *World Congress of The International Federation of Automatic Control (IFAC'19)*, 52(13) :2572–2577, 2019. [48](#)
- [Bas14] F. Basile. Overview of fault diagnosis methods based on petri net models. In *European Control Conference (ECC'14)*, pages 2636–2642. IEEE, 2014. [46](#)
- [Bay63] T. Bayes. Lii. an essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfr s. *Philosophical transactions of the Royal Society of London*, (53) :370–418, 1763. [42](#)
- [BB19] O. Benatia and F. Belmajdoub. An algorithm to diagnose manufacturing hybrid systems. *International Journal of Automation and Control*, 13(3) :374–386, 2019. [51](#)
- [BCDT09] F. Basile, P. Chiacchio, and G. De Tommasi. An efficient approach for online diagnosis of discrete event systems. *IEEE Transactions on Automatic Control*, 54(4) :748–759, 2009. [46](#)
- [BCGM15] M. Bozzano, A. Cimatti, M. Gario, and A. Micheli. Smt-based validation of timed failure propagation graphs. *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'15)*, 29(1), Mar. 2015. [48](#)
- [BCOQ92] F. Baccelli, G. Cohen, G.J Olsder, and J.-P Quadrat. *Synchronization and linearity : an algebra for discrete event systems*. Wiley and sons, 1992. [54](#), [65](#), [66](#), [69](#), [76](#), [77](#), [78](#)
- [BCS14] F. Basile, M. P. Cabasino, and C. Seatzu. State estimation and fault diagnosis of labeled time petri net systems with unobservable transitions. *IEEE Transactions on Automatic Control*, 60(4) :997–1009, 2014. [47](#)
- [Bet12] B. Bettayeb. *Conception et évaluation des plans de surveillance basés sur le risque. Limitation des incertitudes qualité avec des ressources limitées de maîtrise*. PhD thesis, l’Université de Grenoble, 2012. [9](#)
- [BHJ⁺21] P. Bouyer, L. Henry, S. Jaziri, T. Jéron, and N. Markey. Diagnosing timed automata using timed markings. *International Journal on Software Tools for Technology Transfer (STTT)*, pages 1–25, 2021. [47](#)
- [BHRM13] T. Brunsch, L. Hardouin, J. Raisch, and C.A. Maia. Duality and interval analysis over idempotent semirings. *Linear algebra and Its Applications*, 2013. [113](#)

- [BS84] B. G. Buchanan and E. H. Shortliffe. *Rule-based expert systems : the MYCIN experiments of the Stanford Heuristic Programming Project*. CUMINCAD, 1984. 40
- [BW81] J. W. M. Bertrand and J. C. Wortmann. *Production control and information systems for component-manufacturing shops*. Citeseer, 1981. 16
- [CBYP18] A. Chouichi, J. Blue, C. Yugma, and F. Pasqualini. Heterogranular multivariate analytics for detecting and controlling the root causes of the mismatching machines in semiconductor manufacturing. In *Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC'18)*, pages 334–339. IEEE, 2018. 49
- [CD00] M-O. Cordier and C. Dousson. Alarm driven monitoring based on chronicles. *IFAC Proceedings Volumes*, 33(11) :291–296, 2000. IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'20). 47
- [CDPLV18] Q. Christ, S. Dauzère-Pérès, G. Lepelletier, and P. Vialletelle. A multi-purpose operational capacity and production planning tool. In *Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC'18)*, pages 40–44. IEEE, 2018. 19, 20
- [CGPS11] M. P. Cabasino, A. Giua, M. Poccia, and C. Seatzu. Discrete event diagnosis using labeled petri nets. an application to manufacturing systems. *Control Engineering Practice*, 19(9) :989–1001, 2011. 51
- [CGS12] M. P. Cabasino, A. Giua, and C. Seatzu. Diagnosis using labeled petri nets with silent or undistinguishable fault events. *IEEE Transactions on Systems, Man, and Cybernetics : Systems (SMC'12)*, 43(2) :345–355, 2012. 46
- [Chr20] Q. Christ. *Optimisation et aide à la décision pour la planification de production opérationnelle en fabrication de semi-conducteurs*. PhD thesis, Ecole des Mines de Saint-Etienne, 2020. 15
- [CL01] M-O. Cordier and C. Largouët. Using model-checking techniques for diagnosing discrete-event systems. In *International Workshop on Principles of Diagnosis (DX'01)*, pages 39–46. Citeseer, 2001. 47
- [CLHB00] B. Cottenceau, M. Lhommeau, L. Hardouin, and J.-L Boimond. Data processing tool for calculation in dioid. In *International Workshop on Discrete Event Systems (WODES'00)*, 2000. <http://www.istia.univ-angers.fr/~hardouin/outils.html>. 5, 63, 98
- [CLI21] M. Chastan, A. Lam, and F. Iutzeler. Unsupervised density based machine learning for abnormal leveling signatures detection. In *Metrology, Inspection, and Process Control for Semiconductor Manufacturing*. International Society for Optics and Photonics, 2021. 49
- [CMQV89] G. Cohen, P. Moller, J.-P. Quadrat, and M. Viot. Algebraic tools for the performance evaluation of discrete event systems. *Proceedings of the IEEE*, 77(1) :39–85, 1989. 78
- [Coh94] G. Cohen. Diods and discrete event systems. In *International Conference on Analysis and Optimization of Systems Discrete Event Systems (DES'94)*, pages 221–236. Springer, 1994. 65, 66
- [Cot99] B. Cottenceau. *Contribution à la commande de systèmes à événements discrets : synthèse de correcteurs pour les graphes d'événements temporisés dans les dioïdes*. PhD thesis, LISA - Université d'Angers, 1999. 66, 69, 76, 77
- [CRV19] E. Chanthery, P. Ribot, and A. Vignolles. Hymu : a software for hybrid systems health monitoring under uncertainty. In *International Workshop on Principles of Diagnosis (DX'19)*, 2019. 48
- [CWL20] J-C. Chien, M-T. Wu, and J-D. Lee. Inspection and classification of semiconductor wafer surface defects using cnn deep learning networks. *Applied Sciences*, 10(15) :5340, 2020. 49

- [Deq17] K. Dequeant. *Modélisation de la variabilité des flux de production en fabrication microélectronique*. PhD thesis, Université Grenoble Alpes (ComUE), 2017. 59
- [DGG93] C. Dousson, P. Gaborit, and M. Ghallab. Situation recognition : Representation and algorithms. *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'93)*, pages 166–172, 01 1993. 47
- [DMBDT07] M. A. Djeziri, R. Merzouki, B. O. Bouamama, and G. Dauphin-Tanguy. Robust fault diagnosis by using bond graph approach. *IEEE/ASME Transactions on Mechatronics (TMECH)*, 12(6) :599–611, 2007. 50
- [DSF06] D. Delp, J. Si, and J. W. Fowler. The development of the complete x-factor contribution measurement for improving cycle time and cycle time variability. *IEEE Transactions on Semiconductor Manufacturing*, 19(3) :352–362, 2006. 26
- [Dub90] B. Dubuisson. Diagnostic et reconnaissance des formes. Technical report, 1990. 41
- [DVLE16] K. Dequeant, P. Vialletelle, P. Lemaire, and M-L. Espinouse. A literature review on variability in semiconductor manufacturing : the next forward leap to industry 4.0. In *Winter Simulation Conference (WSC'16)*, pages 2598–2609. IEEE, 2016. 25, 26
- [EBC⁺00] T. F. Edgar, S. W. Butler, W. J. Campbell, C. Pfeiffer, C. Bode, S. B. Hwang, KS Balakrishnan, and J. Hahn. Automatic control in microelectronics manufacturing : Practices, challenges, and possibilities. *Automatica*, 36(11) :1567–1603, 2000. 16
- [EW70] F. A. Evans and J. C. WILCOX. Experimental strapdown redundant sensor inertial navigation system. *Journal of Spacecraft and Rockets*, 7(9) :1070–1074, 1970. 44
- [FCdPFT19] E. Fol, JM. Coello de Portugal, G. Franchetti, and R. Tomás. Application of machine learning to beam diagnostics. In *Conference Free Electron Laser (FEL'19)*, pages 311–317, 2019. 41, 42
- [Fra96] P. M. Frank. Analytical and qualitative model-based fault diagnosis—a survey and some new results. *European Journal of control*, 2(1) :6–28, 1996. 43
- [Gau92] S. Gaubert. *Théorie des systèmes linéaires dans les dioïdes*. PhD thesis, Paris, ENMP, 1992. 76
- [GDI02] E. P. Gatzke and F. J. Doyle III. Use of multiple models and qualitative knowledge for on-line moving horizon disturbance estimation and fault diagnosis. *Journal of Process Control*, 12(2) :339–352, 2002. 50
- [Ger98] J. J. Gertler. *Fault detection and diagnosis in engineering systems*. CRC press, 1998. 44
- [GHL18] A. Ghasemi, C. Heavey, and G. Laipple. A review of simulation-optimization methods with applications to semiconductor operational problems. In *Winter Simulation Conference (WSC'18)*, pages 3672–3683, 2018. 21
- [GR88] C. R. Glassey and M. GC. Resende. Closed-loop job release control for vlsi circuit manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, 1(1) :36–46, 1988. 16
- [GRFM06] J. N.D. Gupta, R. Ruiz, J.W. Fowler, and S.J. Mason. Operational planning and control of semiconductor wafer production. *Production Planning & Control*, 17(7) :639–647, 2006. 16
- [GSS96] C. R. Glassey, J. G. Shanthikumar, and S. Seshadri. Linear control rules for production control of semiconductor fabs. *IEEE Transactions on Semiconductor Manufacturing*, 9(4) :536–549, 1996. 16
- [GTY09] M. Ghazel, A. Toguyéni, and P. Yim. State observer for des under partial observation with time petri nets. *Journal of Discrete Event Dynamic Systems*, 19(2) :137–165, 2009. 47

- [Har04] L. Hardouin. *Sur la commande linéaire de systèmes à événements discrets dans l’algèbre (max, +)*. PhD thesis, École d’Ingénieur de l’Université d’Angers (ISTIA), 2004. 83, 84
- [HAYHAL06] N. B. Hadj-Alouane, M. Yeddes, A. B. Hadj-Alouane, and F. Lin. A mixed integer dynamic programming approach to a class of optimal control problems in hybrid systems. *Cybernetics and Systems*, 37(5) :481–504, 2006. 48
- [HL21] C-Y. Hsu and W-C. Liu. Multiple time-series convolutional neural network for fault detection and diagnosis and empirical study in semiconductor manufacturing. *Journal of Intelligent Manufacturing*, 32 :823–836, 2021. 49
- [HMCL10] L. Hardouin, C A. Maia, B. Cottenceau, and M. Lhommeau. Observer design for $(\max, +)$ linear systems. *IEEE Transactions on Automatic Control*, 55(2) :538–543, 2010. 83
- [HMCSM10] L. Hardouin, C.A. Maia, B. Cottenceau, and R. Santos-Mendes. Max-plus linear observer : application to manufacturing systems. In *International Workshop on Discrete Event Systems (WODES’10)*, 2010. 83, 84
- [Hop82] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8) :2554–2558, 1982. 41, 42
- [HY05] F-T. Hung, M-H. Cheng and S-C. Yeh. Development of a web-services-based e-diagnostics framework for semiconductor manufacturing industry. *IEEE Transactions on Semiconductor Manufacturing*, 18(1) :122–135, 2005. 50
- [Ign09] J. P. Ignizio. Cycle time reduction via machine-to-operation qualification. *International Journal of Production Research*, 47(24) :6899–6906, 2009. 26
- [IRA21] F. Pasqualini I. Rabhi, A. Roussy and C. Alegret. Out-of-control detection in semiconductor manufacturing using one-class support vector machines. 2021. 49
- [Ise84] R. Isermann. Process fault detection based on modeling and estimation methods—a survey. *Automatica*, 20(4) :387–404, 1984. 44
- [Ise05] R. Isermann. *Fault-diagnosis systems : an introduction from fault detection to fault tolerance*. Springer Science & Business Media, 2005. 44
- [JKHH10] S-I. Jeon, S-G. Kim, S. J. Hong, and S-S. Han. Endpoint detection of sio 2 plasma etching using expanded hidden markov model. In *International Symposium on Neural Networks (ISNN’10)*, pages 464–471. Springer, 2010. 50
- [JPP⁺19] H. Jeong, B. Park, S. Park, H. Min, and S. Lee. Fault detection and identification method using observer-based residuals. *Reliability Engineering & System Safety*, 184 :27–40, 2019. 50
- [KL15] C. Kim and T.E. Lee. Feedback control of cluster tools for regulating wafer delays. *IEEE Transactions on Automation Science and Engineering*, 13(2) :1189–1199, 2015. 51
- [LG88] C. Lozinski and C.R. Glassey. Bottleneck starvation indicators for shop floor control (semiconductor manufacturing process). *IEEE Transactions on Semiconductor Manufacturing*, 1(4) :147–153, 1988. 15
- [LHK13] B. Li, T. Han, and F. Kang. Fault diagnosis expert system of semiconductor manufacturing equipment using a bayesian network. *International Journal of Computer Integrated Manufacturing*, 26(12) :1161–1171, 2013. 48
- [Lho03] M. Lhommeau. *Étude de systèmes à événements discrets dans l’algèbre(max, +). Synthèse de correcteurs robustes dans un dioïde d’intervalles. Synthèse de correcteurs en présence de perturbations*. PhD thesis, École d’Ingénieur de l’Université d’Angers (ISTIA), 2003. 66, 69, 113, 114

- [LM00] J. Li and S. M. Meerkov. Production variability in manufacturing systems : Bernoulli reliability case. *Annals of Operations Research*, 93(1-4) :299–324, 2000. [26](#)
- [LS01] G.L. Litvinov and AN. Sobolevskii. Idempotent interval analysis and optimization problems. *Reliable Computing*, 7(5) :353–377, 2001. [113](#), [114](#)
- [Max91] MaxPlus. Second order theory of min-linear systems and its application to discrete event systems. In *IEEE Conference on Decision and Control (CDC'91)*, 1991. [65](#), [66](#), [78](#), [80](#)
- [MCDB10] A. M'halla, É. Craye, S. C. Dutilleul, and M. Benrejeb. Monitoring of a milk manufacturing workshop using chronicle and fault tree approaches. *Studies in Informatics and Control*, 19(4) :377–390, 2010. [51](#)
- [MFM12] L. Mönch, J. W. Fowler, and S. J Mason. *Production planning and control for semiconductor wafer fabrication facilities : modeling, analysis, and systems*, volume 52. Springer Science & Business Media, 2012. [7](#), [12](#)
- [MM94] J-F. Magni and P. Mouyon. On residual generation by observer and parity space approaches. *IEEE Transactions on Automatic Control*, 39(2) :441–447, 1994. [44](#)
- [MMJ⁺18] E. Mhiri, F. Mangione, M. Jacomino, P. Vialletelle, and G. Lepelletier. Heuristic algorithm for a wip projection problem at finite capacity in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, 31(1) :62–75, 2018. [19](#), [20](#)
- [Moo06] Gordon E Moore. Cramming more components onto integrated circuits, reprinted from electronics, volume 38, number 8, april 19, 1965, pp. 114 ff. *IEEE solid-state circuits society newsletter*, 11(3) :33–35, 2006. [7](#)
- [Mor02] Y. Mortureux. Arbres de défaillance, des causes et d'événement. 2002. [40](#)
- [MS93] Gary S May and Costas J Spanos. Automated malfunction diagnosis of semiconductor fabrication equipment : A plasma etch application. *IEEE Transactions on Semiconductor Manufacturing*, 6(1) :28–40, 1993. [50](#)
- [Mur89] T. Murata. Petri nets : properties, analysis and applications. In *IEEE Proceedings : Special issue on Discrete Event Systems*, volume 77, 1989. [54](#)
- [NAB⁺12] M. Najafi, D. M. Auslander, P. L. Bartlett, P. Haves, and M. D. Sohn. Application of machine learning in the fault diagnostics of air handling units. *Applied Energy*, 96 :347–358, 2012. [41](#), [42](#)
- [OKT⁺09] M. PL. Ooi, Y. C. Kuang, W. J. Tee, A. A. Mohanan, and C. Chan. Accurate defect cluster detection and localisation on fabricated semiconductor wafers using joint count statistics. In *Asia Symposium on Quality Electronic Design (ASQED'09)*, pages 225–232. IEEE, 2009. [49](#)
- [PC94] R. J. Patton and J. Chen. Review of parity space approaches to fault diagnosis for aerospace systems. *Journal of Guidance, Control, and Dynamics*, 17(2) :278–285, 1994. [50](#)
- [PK93] R. A. Perez and S. W. Koh. Integrating expert systems with a relational database in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, 6(3) :199–206, 1993. [48](#)
- [PMGR13] A. Philippot, P. Marangé, F. Gellot, and B. Riera. Decentralized diagnosis and diagnosability by model checking. *Universal Journal of Control and Automation*, 1(2) :28–33, 2013. [51](#)
- [PS09] Y. Pencolé and A. Subias. A chronicle-based diagnosability approach for discrete timed-event systems : Application to web-services. *Journal of Universal Computer Science*, 15(17) :3246–3272, nov 2009. [47](#)

- [PS17] Y. Pencolé and A. Subias. Diagnosis of supervision patterns on bounded labeled petri nets by model checking. In *International Workshop on Principles of Diagnosis (DX'17)*, page 9p, 2017. [46](#)
- [PSC21] Y. Pencolé, A. Subias, and C. Coquand. A model checking method to solve the event pattern diagnosis problem in safe labeled time petri nets. In *International Workshop on Principles of Diagnosis (DX'21)*, 2021. [47](#)
- [RFN03] J. Robinson, J. Fowler, and E. Neacy. Capacity loss factors in semiconductor manufacturing. *FabTime Inc. http://www.fabtime.com/abs_CapPlan.shtml*, 2003. [25](#), [26](#)
- [SCC06] M-H. Shu, C-H. Cheng, and J-R. Chang. Using intuitionistic fuzzy sets for fault-tree analysis on printed circuit board assembly. *Microelectronics Reliability*, 46(12):2139–2148, 2006. [48](#)
- [Sch99] A. K. Schoemig. On the corrupting influence of variability in semiconductor manufacturing. In *Proceedings of the conference on Winter Simulation Conference (WSC'99)*, pages 837–842, 1999. [26](#)
- [SDC99] K. J. Sullivan, J. B. Dugan, and D. Coppit. The galileo fault tree analysis tool. In *Annual International Symposium on Fault-Tolerant Computing(FTCS'99)*, pages 232–235. IEEE, 1999. [41](#)
- [SDZ07] J. G. Shanthikumar, S. Ding, and M. T. Zhang. Queueing theory for semiconductor manufacturing systems : A survey and open problems. *IEEE Transactions on Automation Science and Engineering*, 4(4):513–522, 2007. [25](#), [26](#)
- [SIA18] SIA. Global billings report history (3-month moving average). *Semiconductor Industry Association*, 2018. [8](#)
- [SLC118] A. Sahuguède, E. Le Corronc, and M-V. Le Lann. Chronicle discovery for diagnosis from raw data : a clustering approach. *IFAC-PapersOnLine*, 51(24):1–8, 2018. [47](#)
- [SLL93] C. J. Spanos, S. Leang, and S. Lee. A control & diagnosis scheme for semiconductor manufacturing. In *American Control Conference (ACC'93)*, pages 3008–3012. IEEE, 1993. [50](#)
- [SLPQ06] P. Supavatanakul, J. Lunze, V. Puig, and J. Quevedo. Diagnosis of timed automata : Theory and application to the damadics actuator benchmark problem. *Control Engineering Practice*, 14(6):609–619, 2006. [51](#)
- [SMB⁺06] A. K. Samantaray, K. Medjaher, B. O. Bouamama, M. Staroswiecki, and G. Dauphin-Tanguy. Diagnostic bond graphs for online fault detection and isolation. *Simulation Modelling Practice and Theory*, 14(3):237–262, 2006. [45](#)
- [SSL⁺95] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, 40(9):1555–1575, 1995. [45](#), [46](#), [47](#)
- [SSS19] J. Singh, S. Singh, and A. Singh. Distribution transformer failure modes, effects and criticality analysis (fmeca). *Engineering Failure Analysis*, 99:180–191, 2019. [40](#)
- [TCG90] A.K.A. Toguyeni, E. CRaye, and J.C. Gentina. A method of temporal analysis to perform on-line diagnosis. In *IEEE International Conference on Industrial Electronics, Control, and Instrumentation (IECON'90)*, pages 445–450, California, USA, 1990. [47](#)
- [TCG97] A.K.A. Toguyeni, E. CRaye, and J.C. Gentina. Time and reasoning for on-line Diagnosis of failures in Flexible Manufacturing Systems. In *World Congress on Scientific Computation, Modeling, and Applied Mathematics (IMACS'97)*, volume 6, pages 709–714, Berlin, Germany, 1997. [47](#)

- [TMP97] L. Trave-Massuyes and R. Pons. Causal ordering for multiple mode systems. In *International Workshop on Qualitative Reasoning (QR'97)*, pages 203–214. Citeseer, 1997. [45](#)
- [Tri02] S. Tripakis. Fault diagnosis for timed automata. In *International Symposium On Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT'02)*, 2002. [47](#)
- [TS10] A. C. Titman and L. D. Sharples. Model diagnostics for multi-state models. *Statistical Methods in Medical Research*, 19(6) :621–651, 2010. [41](#), [42](#)
- [ULMV94] R. Uzsoy, C-Y. Lee, and L. A. Martin-Vega. A review of production planning and scheduling models in the semiconductor industry part ii : Shop-floor control. *IIE transactions*, 26(5) :44–55, 1994. [16](#)
- [UOO98] T. Ushio, I. Onishi, and K. Okuda. Fault detection based on petri net models with faulty behaviors. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC'98)*, volume 1, pages 113–118. IEEE, 1998. [46](#)
- [VMN⁺21] E. Villain, G. M. Mattia, F. Nemmi, P. Péran, X. Franceris, and M. V. le Lann. Visual interpretation of cnn decision-making process using simulated brain mri. In *IEEE International Symposium on Computer-Based Medical Systems (CBMS'21)*, pages 515–520. IEEE, 2021. [42](#)
- [W⁺61] H. A. Watson et al. Launch control safety study. *Bell labs*, 1961. [41](#)
- [Wan98] J. Wang. *Timed Petri nets : Theory and application*, volume 9. Springer Science & Business Media, 1998. [54](#)
- [WGC⁺20] G. Wen, Z. Gao, Q. Cai, Y. Wang, and S. Mei. A novel method based on deep convolutional neural networks for wafer semiconductor surface defect inspection. *IEEE Transactions on Instrumentation and Measurement*, 69(12) :9668–9680, 2020. [49](#)
- [WMS15] X. Wang, C. Mahulea, and M. Silva. Diagnosis of time petri nets using fault diagnosis graph. *IEEE Transactions on Automatic Control*, 60(9) :2321–2335, 2015. [47](#)
- [YJXZ21] W. Yong, L. Jingjing, C. Xu, and W. Zhengying. Fault detection of sensor data in semiconductor processing using neural network with dynamic time wrapping loss. In *China Semiconductor Technology International Conference (CSTIC'21)*, pages 1–3. IEEE, 2021. [49](#)
- [YLLH06] C-C. Yang, W-T. Lin, M-Y. Lin, and J-T. Huang. A study on applying fmea to improving erp introduction : An example of semiconductor related industries in taiwan. *International Journal of Quality & Reliability Management*, 2006. [48](#)
- [Yov97] S. Yovine. Kronos : A verification tool for real-time systems. *International Journal on Software Tools for Technology Transfer*, 1(1-2) :123–133, 1997. [47](#)
- [YXZ20] W. Yong, C. Xu, and W. Zhengying. Fault detection of sensor data in semiconductor processing with variational autoencoder neural network. In *China Semiconductor Technology International Conference (CSTIC'21)*, pages 1–3. IEEE, 2020. [49](#)
- [ZL13] J. Zaytoon and S. Lafourture. Overview of fault diagnosis methods for discrete event systems. *Annual Reviews in Control*, 37(2) :308 – 320, 2013. [45](#)
- [Zwi95] G. Zwingelstein. *Diagnostic des défaillances : théorie et pratique pour les systèmes industriels*. Hermès, 1995. [39](#), [41](#)