

Modèles et calculs garantis pour les systèmes (min,+)-linéaires

Euriell Le Corronc

▶ **To cite this version:**

Euriell Le Corronc. Modèles et calculs garantis pour les systèmes (min,+)-linéaires. Automatique. Université d'Angers, 2011. Français. NNT : . tel-00665973

HAL Id: tel-00665973
https://theses.hal.science/tel-00665973

Submitted on 3 Feb 2012

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

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

université angers

Année : 2011
N° d'ordre : 1176

Modèles et calculs garantis pour les systèmes (min,+)-linéaires

Thèse de doctorat

Spécialité : Automatique et Génie Informatique

École doctorale : STIM (Sciences et Technologies de l'Information et Mathématiques)

Soutenue publiquement :

le : 4 novembre 2011

à : l'ISTIA, École d'Ingénieur de l'Université d'Angers

par : Euriell LE CORRONC

Devant le jury ci-dessous :

Rapporteurs :	Bruno GAUJAL	-	Directeur de Recherche INRIA (LIG)
	Jean-Jacques LESAGE	-	Professeur des Universités à l'ENS Cachan (LURPA - 61°)
Examinateurs :	Christian FRABOUL	-	Professeur des Universités à l'INP Toulouse (IRIT - 27°)
	Giovanni STEA	-	Université de Pise - Italie (Department of Information Engineering)
	Éric THIERRY	-	Maître de Conférences à l'ENS Lyon (LIP - 27°)
Directeur de thèse :	Laurent HARDOUIN	-	Professeur des Universités à l'Université d'Angers (LISA - 61°)
Co-encadrant :	Bertrand COTTENCEAU	-	Maître de Conférences à l'Université d'Angers (LISA - 61°)

Laboratoire : LISA (Laboratoire d'Ingénierie des Systèmes Automatisés)
EA4094 - Université d'Angers
62, avenue Notre Dame du Lac - 49000 ANGERS

ED (N°) : 503

Remerciements

Le travail présenté dans ce mémoire a été préparé au sein du Laboratoire d'Ingénierie des Systèmes Automatisés (LISA) de l'École d'Ingénieur ISTIA, à l'Université d'Angers.

Pour la réalisation de cette thèse, je tiens tout d'abord exprimer ma profonde gratitude à :

- Laurent Hardouin et Bertrand Cottenceau pour leur encadrement sans faille, leurs conseils scientifiques toujours avisés et les encouragements qu'ils m'ont prodigués quand j'en avais besoin,
- Jean-Louis Boimond, et son prédécesseur Jean-Louis Ferrier, pour m'avoir accueillie dans ce laboratoire dont l'ambiance de travail est si agréable,
- Bruno Gaujal, Jean-Jacques Lesage, Christian Fraboul, Giovanni Stea et Éric Thierry pour l'honneur qu'ils m'ont fait en acceptant de juger ce travail écrit et oral. Je tiens à remercier plus encore Bruno Gaujal et Éric Thierry pour m'avoir suivie tout au long de cette thèse en tant que membre de mon comité de suivi de thèse, et également Jean-Jacques Lesage pour avoir accepté d'être rapporteur de cette thèse.

Cette thèse n'aurait pas abouti sans un environnement scientifique stimulant et un cadre de travail optimal. C'est pourquoi je tiens à remercier :

- les membres du groupe WEED / WCTT (Anne Bouillard, Marc Boyer, Laurent Jouhet, ...) pour les réunions de travail toujours très instructives et intéressantes,
- les collègues du bureau E37 (Agnès, Jean-Luc, Julien, Abdelhak, Rémy, ...) pour les pauses cafés à toute heure de la journée, les astuces en LaTeX, les bout de codes en C++ ou encore les installations de Linux,
- les membres du LISA (Sébastien(s), Nicolas, Mehdi, Jean-Louis, ...) pour leur accueil et leurs conseils éclairés du monde de la recherche et de l'enseignement,
- sans oublier Marie-Françoise, Simone, Kristell et Claudine pour leur efficacité administrative,
- le personnel de l'ISTIA, lieu de mes 6 dernières années d'étude, pour leur constante disponibilité,

· les étudiants à qui j'ai donné des cours durant ces 3 années de thèse, certaines de leurs remarques tant orale qu'écrite m'ont fait beaucoup rire.

Merci également à :

· mes amis qui m'ont exprimé à la fois leur respect pour accomplir ce travail, et le sentiment que je suis quand même un peu folle de m'être lancée dans ce projet,

· mes parents pour leur soutien permanent et leur relecture de ce mémoire sans n'y rien comprendre,

· tous ceux qui, à un moment ou à un autre, m'ont souhaité bonne chance et bon courage pour la thèse, ces mots si simples m'ont souvent aidé à avancer.

Mon plus grand remerciement s'adresse bien sûr à Guillaume, sans qui je n'aurais tout simplement rien pu faire.

Table des matières

Liste des notations **vii**

Chapitre 1 – Introduction et état de l'art 1

1 Systèmes à Événements Discrets 1
1.1 Phénomènes de synchronisation et de délai 2
1.2 Modélisation des systèmes linéaires dans les dioïdes 2
1.3 Représentation graphique de ces systèmes 3
2 Domaines d'application 4
2.1 Applications aux systèmes de production 4
2.2 Applications aux réseaux informatiques 7
2.3 D'un point de vue logiciel 9
3 Contributions 11
3.1 Conteneur de systèmes (min,+)-linéaires 11
3.2 Commande de systèmes (min,+)-linéaires incertains 12
3.3 Plan du mémoire 12

Chapitre 2 – Préliminaires algébriques 15

1 Théorie des dioïdes 15
1.1 Définition et exemples 16
1.2 Relation d'ordre et théorie des treillis 17
1.3 Structure ordonnée d'un dioïde 19
2 Quelques exemples de dioïdes 21
2.1 Dioïde de matrices 22
2.2 Dioïde de fonctions 22
2.3 Dioïde de séries formelles 23
2.4 Dioïde quotient 25
3 Résolution d'équations dans un dioïde 29
3.1 Applications isotones et continues 29
3.2 Théorie de la résiduation 30
3.3 Equations aux points fixes 35
4 Conclusion 40

Chapitre 3 – Modélisation de systèmes 43

1 Modélisation des systèmes (min,+)-linéaires 43
1.1 Graphe d'Événements Temporisés 44
1.2 Modélisation analytique : équations aux compteurs 45
1.3 Représentation d'état et relation de transfert 47
1.4 Caractéristiques des systèmes (min,+)-linéaires 49
1.5 Caractéristiques des fonctions de $\mathcal{F}_{cp}$ 52
2 Du point de vue du Network Calculus 55
2.1 Contexte algébrique 56
2.2 Modélisation par des flux et des contraintes 58
2.3 Courbe d'arrivée 62
2.4 Courbe de service 64
2.5 Contrainte sur la sortie 66
3 Mesure des écarts : théorie du second ordre 67
3.1 Corrélation et autocorrélation 67
3.2 Application aux systèmes (min,+)-linéaires 70
3.3 Application au Network Calculus 72
4 Conclusion 76

Chapitre 4 – Conteneur de systèmes (min,+)-linéaires 79

1 Motivations 79
2 Définition du conteneur 81
2.1 Transformée de Legendre-Fenchel des fonctions de $\mathcal{F}_{cp}$ 81
2.2 Fonctions affines et ultimement affines 83
2.3 Opérateurs d'approximation 86
2.4 Définition du conteneur 89
3 Opérations sur les conteneurs 93
3.1 Approximations convexe $\mathcal{C}_{vx}$ et concave $\mathcal{C}_{cv}$ 94
3.2 Fonction d'inclusion de la somme : $[\oplus]$ 94
3.3 Fonction d'inclusion de l'inf-convolution : $[*]$ 96
3.4 Fonction d'inclusion de l'étoile de Kleene : $^{[\star]}$ 98
4 Conclusion 103

Chapitre 5 – Commande de systèmes (min,+)-linéaires incertains 105

1 Contrôleurs existants 106
1.1 Caractérisation de la commande 106
1.2 Précompensateur 108
1.3 Retour de sortie 110

1.4 Contexte de travail . . . 113
2 Retour de sortie : problème du Window Flow Control . . . 114
2.1 Objectif . . . 114
2.2 Calcul de la taille de fenêtre optimale $\hat{w}$ . . . 116
2.3 Exemple d'application . . . 117
3 Réduction de l'incertitude grâce à un précompensateur . . . 118
3.1 Objectif . . . 118
3.2 Calcul de l'incertitude avec un précompensateur . . . 119
3.3 Réduction de l'incertitude d'un système SISO . . . 121
3.4 Exemple d'application . . . 125
4 Précompensateur neutre optimal . . . 127
4.1 Objectif . . . 127
4.2 Dans le cas d'un système SISO . . . 128
4.3 Extension au cas MIMO . . . 133
4.4 Exemple d'application . . . 134
5 Conclusion . . . 134

Conclusion générale **139**

Annexe A – Dioïde de séries formelles $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ . . . 143

1 Modélisation dans le dioïde de séries formelles $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ . . . 144
1.1 Transformée en γ,δ . . . 144
1.2 Monotonie des trajectoires . . . 145
1.3 Relation de transfert . . . 146
1.4 Périodicité : réalisabilité, causalité et rationalité . . . 148
2 Librairie de calculs MinMaxGD . . . 151
2.1 Formes propre, simple et canonique . . . 151
2.2 Clôture des opérations dans MinMaxGD . . . 153
2.3 Opération produit : $\otimes$ . . . 153

Annexe B – Librairie de calculs ContainerMinMaxGD . . . 159

1 Présentation de la librairie . . . 159
1.1 Description de la librairie . . . 159
1.2 Quelques mot-clés et exemples de scripts . . . 161
2 Expérimentations du conteneur . . . 162
2.1 Première expérimentation . . . 162
2.2 Seconde expérimentation . . . 165

2.3 Complexité expérimentale 166

Annexe C – Détails des calculs du chapitre 5 169

1 Réduction de l'incertitude grâce à un précompensateur 169
1.1 Calcul de l'incertitude maximale 169
1.2 Suppression de l'incertitude d'un système SISO 170
2 Précompensateur neutre optimal 171

Bibliographie personnelle **175**

Bibliographie 177

Index 187

Liste des notations

Quelques dioïdes

$\overline{\mathbb{R}}_{min}$:	dioïde complet $(\mathbb{R} \cup \{+\infty\} \cup \{-\infty\}, \min, +)$, couramment appelé « algèbre (min,+) », cf. exemple 2.17
$(\overline{\mathbb{R}}_{min}^{\mathbb{R}}, \oplus, \otimes)$:	dioïde des fonctions $f : \mathbb{R} \mapsto \overline{\mathbb{R}}_{min}$ munies du minimum point à point comme somme $\oplus$ et de l'addition point à point comme produit $\otimes$, cf. exemple 2.23
$\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$:	dioïde des séries formelles croissantes en deux indéterminées commutatives γ et δ, à coefficients booléens dans $\mathbb{B}$ et à exposants dans $\mathbb{Z}$, cf. définition 2.37

Ensembles et dioïdes de fonctions

$f * g$:	opérateur d'inf-convolution avec $\forall t$, $(f * g)(t) = \bigoplus_{\tau \geq 0}\{f(\tau) \otimes g(t-\tau)\} = \min_{\tau \geq 0}\{f(\tau) + g(t-\tau)\}$, cf. définition 3.5
$\mathcal{F}_c$:	ensemble des fonctions croissantes et constantes par morceaux définies de $\mathbb{R}$ dans $\overline{\mathbb{R}}_{min}$, cf. définition 3.3
$(\mathcal{F}_c, \oplus, *)$:	dioïde des fonctions de $\mathcal{F}_c$ munies du minimum point à point comme somme $\oplus$ et de l'inf-convolution comme produit $*$, cf. définition 3.9
$\mathcal{F}_{cp}$:	ensemble des fonctions croissantes, constantes par morceaux et ultimement pseudo-périodiques définies de $\mathbb{R}$ dans $\overline{\mathbb{R}}_{min}$ ($\mathcal{F}_{cp} \subset \mathcal{F}_c$), cf. définition 3.14
$\sigma(f)$:	pente asymptotique d'une fonction $f \in \mathcal{F}_{cp}$, cf. définition 3.15
$(\mathcal{F}_{cp}, \oplus, *)$:	dioïde des fonctions de $\mathcal{F}_{cp}$ munies du minimum point à point comme somme $\oplus$ et de l'inf-convolution comme produit $*$, cf. définition 3.17
$(\mathcal{F}_{cp}, \oplus, *)^{\mathsf{caus}}$:	dioïde des fonctions causales de $\mathcal{F}_{cp}$, cf. définition 3.23
$\mathcal{F}$:	ensemble des fonctions croissantes définies de $\mathbb{R}$ dans $\overline{\mathbb{R}}_{min}$ avec $f(t) = 0$ lorsque $t < 0$, cf. définition 3.26

$(\mathcal{F}, \oplus, *)$: dioïde des fonctions de $\mathcal{F}$ munies du minimum point à point comme somme $\oplus$ et de l'inf-convolution comme produit $*$, cf. définition 3.28

Applications résiduées

$f^\sharp(b)$: application résiduée de f avec $f^\sharp(b) = \bigoplus\{x \in \mathcal{D} \mid f(x) \preccurlyeq b\}$ et $\mathcal{D}$ un dioïde complet, cf. définition 2.47

$L_a^\sharp = a \backslash x$: résiduée du produit à gauche avec $a \backslash b = \bigoplus\{x \mid ax \preccurlyeq b\}$, couramment appelé quotient à gauche, cf. équation (2.9)

$R_a^\sharp = x / a$: résiduée du produit à droite avec $b / a = \bigoplus\{x \mid xa \preccurlyeq b\}$, couramment appelé quotient à droite ou déconvolution avec $\forall t, (f / g)(t) = \bigwedge_{\tau \geq 0} \{f(\tau) / g(\tau - t)\} = \sup_{\tau \geq 0} \{f(\tau) - g(\tau - t)\}$, cf. équation (2.10) et proposition (3.29)

Points fixes d'application

$\mathcal{F}_f$: ensemble des points fixes d'application avec $\mathcal{F}_f = \{x \in \mathcal{D} \mid f(x) = x\}$, cf. définition 2.56

$\mathcal{P}_f$: ensemble des points post-fixes d'application avec $\mathcal{P}_f = \{x \in \mathcal{D} \mid f(x) \succcurlyeq x\}$ et $f(x) \succcurlyeq x \Leftrightarrow f(x) \wedge x = x$, cf. définition 2.56

$\mathcal{Q}_f$: ensemble des points pré-fixes d'application avec $\mathcal{Q}_f = \{x \in \mathcal{D} \mid f(x) \preccurlyeq x\}$ et $f(x) \preccurlyeq x \Leftrightarrow f(x) \oplus x = x$, cf. définition 2.56

$f^\star$: étoile de Kleene ou clôture sous-additive d'une fonction f avec $f^\star = \bigoplus_{i=0}^{+\infty} f^i$, $f^{i+1} = f \circ f^i$ et $f^0 = e$, cf. définition 2.64

$a^\star b$: plus petite solution de $x = ax \oplus b$, cf. théorème 2.68

Modélisation des systèmes (min,+)-linéaires

$x(t)$: fonction compteur avec $x \in \mathcal{F}_c$, cf. équation (3.1)

h : fonction de transfert avec $y = h * u$ (u est l'entrée du système et y sa sortie), cf. équation (3.8)

Δ_T^K : fonction élémentaire valant K lorsque $t \leq T$ et $+\infty$ sinon, cf. définition 3.18

Courbes de contrainte du Network Calculus

$\gamma_{r,b}$: fonction affine, cf. équation (3.16)

$\nu_{T,\tau}$: fonction en escalier, cf. équation (3.17)

δ_T : fonction délai garanti avec $\delta_T = \Delta^0_T$, cf. équation (3.18)

λ_R : fonction simple pente, cf. équation (3.19)

$\beta_{R,T}(t)$: fonction latence-pente avec $\beta_{R,T} = \lambda_R * \delta_T$, cf. équation (3.20)

α : courbe d'arrivée avec $u = \alpha^\star u$, cf. définition 3.40 et proposition 3.41

$[\ \underline{\beta}\ ,\ \overline{\beta}\ ]$: intervalle de courbes de service avec $\underline{\beta}u \preccurlyeq y \preccurlyeq \overline{\beta}u$, cf. équation (3.23)

α' : contrainte sur la sortie avec $\alpha' = (\alpha^\star \underline{\beta}) \not\phi \overline{\beta}$ et $y \succcurlyeq \alpha' y$, cf. définition 3.51 et proposition 3.52

Mesure des écarts de performances

$\Sigma_\tau(v_1, v_2)$: distance maximale en temps entre les fonctions v_1 et v_2 avec $\Sigma_\tau(v_1, v_2) = \inf_{\tau \geq 0}\{\tau \mid (v_2 \not\backslash v_1)(-\tau) \leq 0\}$, cf. théorème 3.54

$\Sigma_\nu(v_1, v_2)$: distance maximale en événement entre les fonctions v_1 et v_2 avec $\Sigma_\nu(v_1, v_2) = (v_2 \not\backslash v_1)(0)$, cf. théorème 3.54

$d(t)$: délai de bout-en-bout des paquets avec $d(t) \leq \Sigma_\tau(u, y) \leq \Sigma_\tau(\alpha^\star, \overline{\beta})$, cf. proposition 3.64

$b(t)$: backlog à l'instant t avec $b(t) \leq \Sigma_\nu(u, y) \leq \Sigma_\nu(\alpha^\star, \overline{\beta})$, cf. proposition 3.64

Conteneur de systèmes (min,+)-linéaires

$\mathcal{L}$: transformée de Legendre-Fenchel d'une fonction $f \in \mathcal{F}_{cp}$ avec $\mathcal{L}(f)(s) = \sup_t\{s.t - f(t)\}$, cf. définition 4.3

$\mathcal{F}_{cp/\mathcal{L}}$: dioïde $\mathcal{F}_{cp}$ quotienté par la transformée $\mathcal{L}$, cf. définition 4.5

$\mathcal{F}_{aa}$: ensemble des fonctions croissantes, affines par morceaux et ultimement affines définies de $\mathbb{R}$ dans $\overline{\mathbb{R}}_{min}$, cf. définition 4.6

$\sigma(f)$: pente asymptotique d'une fonction $f \in \mathcal{F}_{aa}$, cf. définition 4.7

$\mathcal{F}_{acx}$: ensemble des fonctions convexes avec $\mathcal{F}_{acx} \subset \mathcal{F}_{aa}$, cf. définition 4.9

$\mathcal{F}_{acv}$: ensemble des fonctions concaves sur $]\tau_f, +\infty[$ avec $\mathcal{F}_{acv} \subset \mathcal{F}_{aa}$, cf. définition 4.10

Γ_f : partie concave d'une fonction $f \in \mathcal{F}_{acv}$ avec $f = \Delta_{\tau_f}^{\kappa_f} * \Gamma_f$, cf. équation (4.1)

$\mathcal{C}_{vx}$: approximation convexe (d'une fonction de $\mathcal{F}_{cp}$ ou de $\mathcal{F}_{aa}$), cf. définition 4.15

$\mathcal{C}_{cv}$: approximation concave (d'une fonction de $\mathcal{F}_{cp}$ ou de $\mathcal{F}_{aa}$), cf. définition 4.19

$\mathbf{f}$: conteneur de $\mathbf{F}$ avec $\mathbf{f} = [\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}} = [\ \underline{f}\ ,\ \overline{f}\ ] \cap [\ \overline{f}\ ]_{\mathcal{L}} = \{\ f \mid \underline{f} \preccurlyeq f \preccurlyeq \overline{f}\ ,\ [f]_{\mathcal{L}} = [\ \overline{f}\ ]_{\mathcal{L}}\}$, cf. définition 4.21

$\Omega(\overline{f})$: minorant de la classe d'équivalence $[\ \overline{f}\ ]_{\mathcal{L}}$, cf. définition 4.22

$[\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}}$: forme canonique d'un conteneur de $\mathbf{F}$ avec $[\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}} = [\ \mathcal{C}_{cv}\big(\underline{f} \oplus \Omega(\overline{f})\big)\ ,\ \overline{f}\ ]_{\mathcal{L}}$, cf. définition 4.24

$\Theta_{\mathbf{f}}$: incertitude maximale (en temps et en événement) d'un conteneur $\mathbf{f} \in \mathbf{F}$, cf. définition 4.26

$\{[\oplus], [*], {}^{[\star]}\}$: fonctions d'inclusion des conteneurs de $\mathbf{F}$, cf. définition 4.27, et propositions 4.30, 4.34 et 4.36

Chapitre 1

Introduction et état de l'art

Sommaire

1 Systèmes à Événements Discrets 1
1.1 Phénomènes de synchronisation et de délai 2
1.2 Modélisation des systèmes linéaires dans les dioïdes 2
1.3 Représentation graphique de ces systèmes 3
2 Domaines d'application 4
2.1 Applications aux systèmes de production 4
2.2 Applications aux réseaux informatiques 7
2.3 D'un point de vue logiciel 9
3 Contributions 11
3.1 Conteneur de systèmes (min,+)-linéaires 11
3.2 Commande de systèmes (min,+)-linéaires incertains 12
3.3 Plan du mémoire 12

1 Systèmes à Événements Discrets

Sous l'appellation Systèmes (Dynamiques) à Événements Discrets (SED) sont regroupés des systèmes généralement conçus par l'homme et dont les transformations ont lieu à des instants discrets, en réponse à des événements ponctuels. A l'instar des systèmes dits « naturels » qui peuvent être modélisés par des équations différentielles dépendantes du temps, les systèmes à événements discrets doivent bénéficier de modèles aptes à prendre en compte leurs caractéristiques dynamiques suscitées par divers phénomènes (tâches séquentielles ou simultanées, temporisées ou non, synchronisées ou concurrentes, etc.).

L'étude des SED s'appuie sur des modèles graphiques et mathématiques variés. Parmi tous les modèles étudiés, les réseaux de Petri constituent un modèle important puisqu'ils permettent de décrire une grande variété des phénomènes

dynamiques à savoir le délai temporel, la précédence, le parallélisme, la synchronisation, la concurrence, la notion de ressource partagée, etc. En revanche, sur le plan de l'analyse, il est difficile d'appréhender la grande variété des comportements dynamiques que peut engendrer un réseau de Petri général.

C'est pourquoi les SED sont également étudiés à travers d'autres modèles, parfois moins généraux, mais pour lesquels les phénomènes dynamiques sont mieux délimités et donc mieux compris. Par exemple, les systèmes non temporisés déterministes peuvent être décrits par des automates à états finis, et disposent ainsi d'une théorie permettant d'aborder des problèmes de contrôle (théorie de Ramadge et Wonham, cf. [Wonham 1987]).

1.1 Phénomènes de synchronisation et de délai

La classe des SED présentant essentiellement des phénomènes de synchronisation et de délai est amplement étudiée depuis le début des années 1980. Il est en effet possible d'obtenir un modèle mathématique de ces systèmes sous la forme d'équations de récurrence qui utilisent des opérateurs max, min et la somme +. Sous cette forme, les opérations ne sont pas linéaires, mais traduites dans une structure algébrique particulière appelée *dioïde*, ou encore *semi-anneau idempotent*, la linéarité apparaît.

Les recherches autour de cette méthode de modélisation ont été regroupées de manière générale sous divers noms : théorie des systèmes (max,+)-linéaires / (min,+)-linéaires[1] ou encore théorie des systèmes linéaires dans les dioïdes. Un ouvrage de référence pour cette théorie est [Baccelli 1992], et d'autres travaux plus récents ont également vu le jour dans les livres [Chang 2000], [Le Boudec 2001] et [Heidergott 2006].

1.2 Modélisation des systèmes linéaires dans les dioïdes

Les systèmes étudiés dans ce mémoire ont en commun l'algèbre des dioïdes comme cadre algébrique ainsi que des modèles dynamiques semblables. Par la suite, un système est dit (min,+)-linéaire s'il s'agit d'un SED pouvant être décrit au moyen

1. La différence entre les appellations (max,+) et (min,+) est liée au point de vue adopté lors de l'étude.

de la représentation d'état suivante :

$$\begin{cases} x(t) = A \otimes x(t-1) \oplus B \otimes u(t), \\ y(t) = C \otimes x(t). \end{cases} \tag{1.1}$$

Les opérations somme $\oplus$ et produit $\otimes$ sont celles de l'algèbre (min,+) étendues aux matrices, c'est-à-dire que $\oplus = \min$ et $\otimes = +$ (la somme classique). Les fonctions considérées $x(t)$, $u(t)$ et $y(t)$ sont quant à elles des compteurs d'événements. Le modèle d'état ainsi obtenu relie l'évolution de l'entrée $u(t)$ à celle du vecteur d'état $x(t)$, puis à la sortie $y(t)$. Il est clair que cette représentation d'état présente une grande analogie de forme avec celle des systèmes linéaires continus, notamment lorsqu'il s'agit des modèles à temps discret.

De même qu'en théorie des systèmes, il est également possible d'obtenir une relation liant l'entrée à la sortie via une convolution, en l'occurrence une inf-convolution :

$$y(t) = (h * u)(t). \tag{1.2}$$

La fonction $h(t)$ est alors la sortie $y(t)$ du système en réponse à un signal d'entrée $u(t)$ spécifique qui joue le rôle d'une impulsion. Cette fonction h peut donc être considérée comme analogue à la réponse impulsionnelle des systèmes linéaires conventionnels.

Les premiers travaux sur ces représentations sont proposés dans [Cohen 1989]. Les trajectoires $x(t)$, $u(t)$ et $y(t)$ sont alors représentées par des séries formelles en deux indéterminées γ et δ, et à coefficients booléens. Cette représentation offre l'avantage d'être en deux dimensions, l'indéterminée γ représente les événements tandis que l'indéterminée δ représente le temps.

1.3 Représentation graphique de ces systèmes

Un outil de modélisation graphique décrivant la dynamique des SED présentée dans (1.1) et (1.2) est le Graphe d'Événements Temporisés (GET), sous-classe des réseaux de Petri pour laquelle toute place ne possède qu'une transition en amont et une en aval.

Dans un GET, la synchronisation est représentée dans la consommation de jetons (lorsque plusieurs arcs convergent sur une même transition, cf. figure 1.1a)

ou la production de jetons (lorsque plusieurs arcs divergent depuis une même transition, cf. figure 1.1b). En raison de cette particularité, les GET sont souvent appelés des modèles déterministes. Enfin, l'introduction du temps dans un GET se fait par les temps de séjour associés aux places des graphes.



(a) Consommation de jetons

(b) Production de jetons

Figure 1.1 Synchronisation dans un Graphe d'Événements Temporisés.

2 Domaines d'application

Parmi les domaines d'applications de la théorie des systèmes (min,+)-linéaires, on peut citer les systèmes de production, les réseaux de transport ou encore les réseaux informatiques. Chaque domaine utilise à sa propre manière les éléments de la théorie et à des fins parfois différentes.

2.1 Applications aux systèmes de production

Dans le cadre d'une application de cette théorie au domaine des systèmes de production, l'objectif est double. D'une part, à l'instar de l'automatique classique, on souhaite pouvoir aborder des problèmes de commande et ainsi répondre à cette question : quand et en quelle quantité faut-il lancer les ordres de fabrication pour optimiser les performances du système de production ? D'autre part, des indicateurs de performance tels que des temps de cycle, des taux de production, une taille de stock ou encore des temps d'attente, sont également recherchés.

Lorsqu'un système manufacturier de production peut être décrit par un modèle tel que (1.1), le problème de pilotage de la production est alors exprimé de la façon

suivante : la fonction $u(t)$ décrit les dates et quantités des lancements de produits bruts ou semi-finis en production, et la fonction $y(t)$ décrit les dates et quantités de produits finis (ou semi-finis) disponibles en sortie du système.

2.1.1 Commande des systèmes (min,+)-linéaires

Lors de la conception d'une commande d'un système, il est nécessaire de définir un critère à optimiser. Ainsi, lorsque les exigences sur la disponibilité de produits finis en sortie du système sont fixées, le critère choisi et connu sous le nom critère du *juste-à-temps*[2], permet de trouver la trajectoire $u(t)$ décrivant le déclenchement le plus tardif des ordres de fabrication en entrée du système. La quantité juste est ainsi disponible au plus tard avant la date fixée par le client.

Par analogie avec l'automatique conventionnelle, une première structure permettant de réaliser une commande selon ce critère du juste-à-temps, est une commande optimale en boucle ouverte dont l'objectif est la poursuite d'une trajectoire connue *a priori*. Ce problème de pilotage apparaît alors comme une inversion de la dynamique du système (1.1), ou de manière équivalente comme une pseudo-inverse du produit de convolution de (1.2). Il a été posé et résolu la première fois dans [Cohen 1989].

Une autre démarche de commande en boucle ouverte correspond à la recherche d'un précompensateur optimal selon le critère du juste-à-temps. Il s'agit d'un contrôleur placé en amont du système afin de compenser la dynamique du système représenté par sa fonction de transfert h de (1.2), et de s'approcher au mieux d'un modèle de référence décrivant le comportement entrée/sortie désiré. Ce problème est par exemple traité dans [Gruet 1995], [Libeaut 1996], [Hardouin 1997] ou encore [Cottenceau 1999b].

Toujours inspiré de l'automatique classique, un rebouclage de la sortie (ou de l'état interne) sur l'entrée permet de tenir compte directement ou indirectement de l'évolution du système pour gérer au mieux l'entrée, à nouveau au sens du juste-à-temps. Une telle structure de pilotage étudiée peut être décrite par le modèle d'état

2. Traduction littérale du terme anglais « just-in-time ».

suivant :

$$\begin{cases} x(t) = A \otimes x(t-1) \oplus B \otimes u(t) \oplus F \otimes y(t), \\ y(t) = C \otimes x(t), \end{cases}$$

où la matrice F correspond au retour de la sortie. Ce problème est par exemple traité dans [Cottenceau 2001], [Luders 2002] et [Maia 2003a]. Un tel feedback F peut ainsi être vu comme une extension du système de gestion de production appelé Kanban (systèmes de gestion dits à *flux tirés*).

De nombreux travaux cherchent à développer plus encore les analogies entre la théorie des systèmes (min,+)-linéaires et celle de l'automatique classique en tentant de transposer, voire de ré-interpréter au besoin, les résultats existants. On peut notamment citer :

- le problème d'identification : déduction d'un modèle à partir d'une observation des signaux $u(t)$ et $y(t)$ d'un système (cf. [Gallot 1997] et [Maia 2003b]),
- le problème de rejet de perturbation : la commande doit tenir compte de certains aléas comme par exemple des pannes de machines (cf. [Lhommeau 2003b], [Ouerghi 2006a, Ouerghi 2006b] et [Shang 2011]),
- le problème de robustesse : mesure du domaine de variation des paramètres du système (1.1) qui ne remet pas en cause l'optimalité de la commande (cf. [Lhommeau 2004b]),
- la synthèse d'observateur : reconstruction de l'état du système à partir du modèle, ainsi que des entrées et sorties mesurées du système (cf. [Hardouin 2004, Hardouin 2010a, Hardouin 2010b]).

Remarque 1.1 (Applications aux réseaux de transports). Un grand nombre de travaux utilise les modèles (min,+) pour la planification de moyens de transports (ferroviaires, routiers, etc.). On peut notamment citer le livre [Heidergott 2006], mais également [Houssin 2004], [Katz 2007] ou encore plus récemment [Maia 2011].

2.1.2 Extension aux systèmes incertains

Il est parfois difficile d'approcher le comportement d'un système réel, qu'il soit manufacturier, informatique ou de transport par un système tel que (1.1). Certains phénomènes de choix peuvent apparaître (problème de routage), des paramètres peuvent être incertains ou variables, les délais ne sont alors pas constants. C'est

pourquoi, notamment à des fins d'application, beaucoup de travaux ont tenté d'élargir la classe des systèmes pour lesquels la théorie puisse s'appliquer. Une méthode consiste à décrire les délais et ressources par des intervalles. Ces extensions se sont notamment inspirées de l'analyse par intervalles dont les premiers résultats sont apparus dans [Moore 1979]. Un ouvrage de référence plus récent est [Jaulin 2001].

Dans le cadre des semi-anneaux idempotents, l'analyse par intervalles a été initialement proposée par [Litvinov 2001], tandis que des applications sont proposées dans [Di Loreto 2009] et [Hardouin 2009]. Les systèmes modélisés le sont grâce à l'utilisation de dioïdes d'intervalles qui contiennent leurs comportements et dont les bornes représentent les extrêmes.

On peut également citer comme exemple d'application un résultat de commande robuste proposé dans [Lhommeau 2004a] et permettant de manipuler des systèmes dont les matrices A, B et C de la représentation d'état (1.1) comportent une ou plusieurs valeurs incertaines décrites par des intervalles. Les délais du système peuvent être variables mais restent bornés.

Enfin, un autre exemple consiste à encadrer le comportement de certains systèmes présentant des conflits (systèmes a priori mal conditionnés), par des comportements extrêmes représentés grâce à un dioïde d'intervalles. Les bornes de ces intervalles sont décrites par des systèmes (min,+)-linéaires et correspondent alors au comportement le plus rapide et à celui le plus lent. Certains problèmes de partages de ressources peuvent ainsi être traités, au prix d'une politique particulière d'accès à la ressource (cf. [Boutin 2008]). De même, certains problèmes de routage dans des systèmes parallèles peuvent être encadrés par des comportements extrêmes (min,+)-linéaires (cf. [Boutin 2009]).

2.2 Applications aux réseaux informatiques

L'application de la théorie des systèmes (min,+)-linéaires aux réseaux informatiques est principalement connue sous le nom *Network Calculus*. Le cadre simplifié de cette théorie est alors de caractériser les flux de données et les comportements dynamiques des équipements d'un réseau informatique par des courbes. L'analyse des performances du réseau, via des indicateurs tels que les délais de transmission des paquets ou le nombre de paquets en attente de service, s'appuie alors sur les

opérations et propriétés relevant de l'algèbre (min,+).

Cette approche de l'évaluation de la performance des réseaux a été initiée dans [Cruz 1991a, Cruz 1991b] par l'introduction de la notion de *courbe de service*, et appliquée en exemple sur un cas de multiplexeur particulier [3] dans [Parekh 1993, Parekh 1994]. Ces travaux ont par la suite été repris et développés dans plusieurs ouvrages, dont les deux livres de référence suivants : [Chang 2000] et [Le Boudec 2001]. Un rappel des travaux récents de cette théorie est donné dans [Fidler 2010].

2.2.1 Modélisation dans l'algèbre (min,+)

Dans la théorie du Network Calculus, les flux de données sont décrits au moyen de fonctions compteurs, contenant par exemple le nombre cumulé de paquets ayant transités par un lien du réseau jusqu'à la date t. Puis, des courbes de service permettent de modéliser les différents équipements tels qu'un serveur, un routeur, un switch ou encore un lien physique. Ces courbes représentent alors le service que l'élément peut fournir dans un intervalle de temps donné. Le modèle présenté par l'équation (1.2) est utilisé pour décrire la dynamique élémentaire entrée/sortie d'un équipement du réseau, et ainsi le service fourni par l'équipement s'exprime grâce à une convolution dans l'algèbre (min,+).

Moyennant certaines hypothèses sur les flux de données, les modèles du Network Calculus permettent donc d'une part de calculer le service résultant de la combinaison de différents équipements, mais aussi de déduire les temps de transports dans le pire cas de fonctionnement ainsi que le nombre maximal de paquets en transit sur un lien.

2.2.2 Applications et extensions

Une variété d'applications du Network Calculus, et donc de recherche d'indicateurs de performance lors des pire cas de fonctionnement, a émergée au cours des dernières années. On peut par exemple citer :

- le problème du contrôle de flux dans des réseaux en boucle ouverte ou en boucle fermée (cf. [Agrawal 1999]),

3. Appelé « Generalized Processor Sharing » en anglais.

- les réseaux de capteurs sans fil : analyse du pire cas de traffic (cf. [Schmitt 2005]),
- les réseaux embarqués critiques de type AFDX [4] : recherche de bornes sur les temps de latence et la taille des files d'attente (cf. [Frances 2006] et [Boyer 2008]),
- les réseaux de multiplexeurs de type FIFO mis en série : recherche de bornes sur les délais pire-cas (cf. [Lenzini 2008]).

Enfin, cette théorie a également connu une extension à travers le Real Time Calculus qui correspond à l'application des modèles et outils du Network Calculus dans le cadre des systèmes temps réels (cf. [Chakraborty 2003, Chakraborty 2006]). A la différence du Network Calculus, les flux d'entrée d'un équipement réseau sont contraints par un intervalle de courbes dont les bornes peuvent être interprétées comme les quantités minimum et maximum d'événements en entrée de l'équipement. De la même manière, le service fourni par un élément de réseau est donné à travers un intervalle de service dont les bornes représentent des capacités de ressources minimum et maximum.

2.3 D'un point de vue logiciel

2.3.1 Librairies existantes

Les premiers éléments manipulés dans des librairies de calculs sont les séries formelles ultimement périodiques du dioïde appelé $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ (cf. définition 2.37 et annexe A), et qui peuvent être présentées dans le contexte de cette thèse comme des fonctions définies de $\mathbb{Z}$ dans $\mathbb{Z}$, *constantes par morceaux* et *ultimement pseudo-périodiques*. Deux outils ont été développés dans ce sens.

- Durant sa thèse [Gaubert 1992], Stéphane Gaubert a développé une bibliothèque de calculs pour manipuler les séries formelles périodiques. Ce premier outil a été diffusé sous le nom **MaxPlus** et créé pour le logiciel Mapple.
- Les algorithmes ont ensuite été repris et étendus par Laurent Hardouin à travers plusieurs travaux de master et de thèses (cf. [Gruet 1995], [Cottenceau 1999a], [Lhommeau 2003a]), dans une bibliothèque C++ appelée **MinMaxGD** et possédant une interface avec le logiciel Scilab (cf. [Cottenceau 2000] et annexe A.2). Les éléments manipulés dans MinMaxGD sont alors des matrices de séries formelles périodiques dont il est possible de

4. Pour « Avionics Full-Duplex Switched Ethernet ».

calculer la *somme*, le *produit*, l'*étoile de Kleene* (également appelée *clôture sous-additive*) et enfin la *résiduation* (soit le *quotient*). Les complexités de ces opérations sont données dans [Cottenceau 2006] et la méthode de calcul du produit avec sa complexité algorithmique est reprise dans le théorème A.28 de l'annexe A.

Pour le Network Calculus, ce mode de représentation n'est pas réellement adapté aux différentes fonctions de la littérature. En effet, les fonctions manipulées relèvent pour la plupart des fonctions croissantes, *affines par morceaux* et *ultimement pseudo-périodiques* ou *ultimement affines*. C'est pourquoi, des équipes ont également développé des outils spécifiques permettant de manipuler ces objets, dont voici une liste non exhaustive.

- La librairie **DISCO** écrite en Java (cf. [Schmitt 2006]), est un calculateur pour le Network Calculus manipulant des fonctions affines par morceaux et ultimement affines.
- La librairie **COINC** (COmputational Issues in Network Calculus), écrite en C++ (cf. [Bouillard 2009]). Les algorithmes de calculs ainsi que les complexités qui leurs sont associées sont proposés dans [Bouillard 2008b].
- La librairie **CyNC** (Cyclic Network Calculus) conçue pour l'environnement Matlab/Simulink (cf. [Schioler 2007]) afin notamment de traiter des problèmes de dépendances cycliques.
- La librairie **RTC** (Real Time Calculus Toolbox) écrite en Java et interfacée avec Matlab (cf. [Wandeler 2006]) permet la manipulation de fonctions affines par morceaux et ultimement pseudo-périodiques. Le calcul de la clôture sous-additive n'est cependant pas disponible dans cet outil.

2.3.2 Complexités spatiales des opérations

L'inconvénient principal de toutes ces librairies est que les calculs peuvent engendrer des résultats pour lesquels le comportement ultimement pseudo-périodique, ou ultimement affine, s'installe après un comportement transitoire arbitrairement long. L'efficacité des algorithmes est alors pénalisée par l'occupation spatiale des résultats.

Par exemple, réalisons le calcul d'un produit de deux fonctions ultimement pseudo-périodiques dont la représentation informatique ne nécessite que peu d'information. Une fonction f est dite ultimement pseudo-périodique lorsqu'à partir d'un

certain temps T_p elle est définie comme suit :

$$\exists K, T \in \mathbb{Z} \quad \text{tels que} \quad \forall t \geq T_p,\ f(t) = K + f(t - T).$$

Soient f et f' deux fonctions croissantes et ultimement pseudo-périodiques pour lesquelles $f(t) = f(T_p)$ et $f'(t) = f'(T_p)$ lorsque $t < T_p$. Les valeurs de T_p, T et K de ces fonctions sont données ci-dessous :

- pour f : $T_p = 6$, $f(T_p) = 5$, $T = 23$ et $K = 22$,
- pour f' : $T_p = 7$, $f'(T_p) = 6$, $T = 24$ et $K = 23$.

Le produit de ces deux fonctions $g = f * f'$ est réalisé grâce à la librairie MinMaxGD et fournit le résultat suivant proposé par couple de coordonnées $(t, g(t))$:

$$(13, 11)\ ;\ (36, 33)\ ;\ (37, 34)\ ;\ \ldots\ ; (496, 473)\ ;\ (497, 474)\ ;\ (498, 475)\ ;\ \ldots\ .$$

La périodicité du produit de ces deux fonctions commence à $T_p = 496$, avec $g(T_p) = 473$. Le nombre de points donnés avant ce régime périodique est 253 tandis que le motif périodique nécessite 22 points afin d'être représenté.

Ces explosions de comportements transitoires se produisent en particulier lorsque les périodicités des fonctions sont proches, *i.e.* les rapports K/T. En effet, le développement du régime transitoire du résultat dépend notamment de la valeur à partir de laquelle une des deux fonctions se retrouve définitivement « au-dessus » de l'autre fonction. Cette valeur ne peut être connue qu'en réalisant ce développement coûteux en espace mémoire et subséquemment en temps de calcul.

3 Contributions

3.1 Conteneur de systèmes (min,+)-linéaires

Le premier objectif du travail présenté dans ce mémoire est l'élaboration d'un outil de calculs efficace pour les systèmes (min,+)-linéaires. A l'image du calcul par intervalles (cf. [Moore 1979, Jaulin 2001]), l'idée consiste à réaliser des calculs approchés qui encadrent de manière garantie les résultats.

La sortie des algorithmes est un conteneur, c'est-à-dire un ensemble inclus dans un intervalle dont la représentation informatique est volontairement réduite et dont les bornes sont des fonctions affines et ultimement affines. Nous proposons des

algorithmes de complexité linéaire et quasi-linéaire afin de réaliser les approximations de la somme (qui correspond au min point à point), de l'inf-convolution, et de l'étoile de Kleene (soit la clôture sous-additive).

En outre, lorsqu'on applique ces algorithmes au calcul de la fonction de transfert d'un système défini par une représentation d'état telle que (1.1), la borne supérieure du conteneur obtenu est dans la même classe d'équivalence que la fonction de transfert exacte du système, modulo la transformée de Legendre-Fenchel (cf. [Rockafellar 1997, Fidler 2006]). Nous conservons donc grâce à cette borne certaines informations du système réel. Par exemple, la pente asymptotique du système encadré est conservée tout au long des calculs.

3.2 Commande de systèmes (min,+)-linéaires incertains

Dans ce contexte de modèle incertain, nous proposons dans un second temps différents problèmes de commande pour lesquels le comportement entrée/sortie du système n'est connu qu'à travers un intervalle, soit par exemple lorsque la dynamique des systèmes considérés contient des paramètres incertains. Ces résultats qui nous semblent originaux dans ce contexte incertain, sont applicables à la fois pour des systèmes de production, mais également dans le cadre du Network Calculus.

Dans cette étude, nous nous limiterons aux cas pour lesquels les hypothèses de fonctionnement sont données de type First In First Out (FIFO) et dont les systèmes étudiés sont décrits comme étant des systèmes causals (l'état actuel ne dépend pas du futur) et déterministes (dont on peut prévoir exactement l'évolution dans le temps).

3.3 Plan du mémoire

Afin de présenter cette théorie des systèmes (min,+)-linéaires ainsi que les contributions que nous souhaitons apporter, ce mémoire de thèse est organisé comme suit.

Tout d'abord, le chapitre 2 présente les notions algébriques nécessaires à l'étude des systèmes (min,+)-linéaires. Les dioïdes, ou semi-anneau idempotents, sont présentés en lien avec la théorie des treillis. Des résolutions d'équations définies sur cette structure sont également proposées grâce à l'application de la théorie de la

résiduation et de la théorie des points fixes d'applications.

Ensuite, le chapitre 3 propose deux exemples de modélisations des systèmes (min,+)-linéaires. La première section définit les relations entre les entrées, les état internes et les sorties du système modélisé à travers sa représentation d'état donnée en (1.1). Puis, nous nous intéressons à la fonction de transfert du système, similaire à une réponse impulsionnelle, et issue de la relation entrée/sortie donnée en (1.2). La seconde section est tournée vers la théorie du Network Calculus avec des contraintes appliquées aux flux d'entrée et de sortie d'un réseau. La relation (1.2) est à nouveau proposée avec cette fois des bornes inférieures et supérieures sur les dynamiques des éléments de réseaux représentés, soit le service qu'ils fournissent. Enfin, la dernière section de ce chapitre permet l'utilisation d'indicateurs de performance tels que des temps de séjour dans un système, des délais de bout-en-bout d'un réseau, des stocks de produits, ou encore des quantités de paquets en attente de service.

Le conteneur présenté dans le chapitre 4 (et l'annexe B) permet d'encadrer les systèmes (min,+)-linéaires pour les raisons invoquées précédemment. Nous proposons ainsi de disposer d'un outil de calculs lorsque les systèmes modélisés ne sont pas connus exactement, ou lorsqu'il est nécessaire d'approcher les résultats afin de diminuer la complexité des opérations. Les bornes de ce conteneur ont des propriétés de convexité qui facilitent leurs manipulations. En outre, la borne supérieure est cohérente par rapport au système encadré puisqu'elle est l'élément maximal de la classe d'équivalence de ce système réel, modulo la transformée de Legendre-Fenchel (cf. [Le Corronc 2009a]). Les opérations sur ces conteneurs sont la somme, le produit d'inf-convolution et l'étoile de Kleene. A l'instar du calcul par intervalles, elles sont proposées à travers des fonctions d'inclusion dont les algorithmes sont de complexités linéaire ou quasi-linéaire. Une première publication relative à ce travail est [Le Corronc 2009c], une autre [Le Corronc 2011a] est actuellement en cours de soumission.

Finalement, dans le chapitre 5, nous reprenons différentes structures de contrôle (précompensateur, retour de sortie, problème du Window Flow Control) afin de les appliquer à des systèmes (min,+)-linéaires incertains. Ces résultats existent déjà dans un contexte de systèmes exacts, mais suite aux calculs réalisés sur des conteneurs, nous montrons qu'il est toujours possible de contrôler les systèmes devenus incertains. Nous proposons notamment de limiter les entrées dans un réseau selon un critère d'une quantité totale de données en transit à ne pas

dépasser (cf. [Le Corronc 2010b, Le Corronc 2011b]). Ensuite, nous proposons une loi de commande élaborée afin de diminuer l'incertitude sur la sortie d'un système connu uniquement à travers un intervalle (cf. [Le Corronc 2010a]). Pour finir, nous posons le problème de ralentir les entrées d'un système incertain sans en altérer la dynamique (cf. [Le Corronc 2009b]).

Chapitre 2

Préliminaires algébriques

Sommaire

1 Théorie des dioïdes 15
1.1 Définition et exemples 16
1.2 Relation d'ordre et théorie des treillis 17
1.3 Structure ordonnée d'un dioïde 19
2 Quelques exemples de dioïdes 21
2.1 Dioïde de matrices 22
2.2 Dioïde de fonctions 22
2.3 Dioïde de séries formelles 23
2.4 Dioïde quotient 25
3 Résolution d'équations dans un dioïde 29
3.1 Applications isotones et continues 29
3.2 Théorie de la résiduation 30
3.3 Equations aux points fixes 35
4 Conclusion 40

1 Théorie des dioïdes

Un dioïde est un semi-anneau idempotent dont le plus populaire est sans doute l'algèbre (max,+). Il fut largement utilisé pour modéliser les Systèmes à Événements Discrets au cours des années 80 (voir notamment [Cuninghame-Green 1979] et [Cohen 1989]). Le nom *dioïde* fut introduit par Gondran et Minoux en référence aux deux monoïdes sous-jacents (cf. [Gondran 1984]). L'idempotence de la loi additive induit une relation d'ordre sur cette structure algébrique qui est donc également un treillis (la théorie des treillis est par exemple détaillée dans [Blyth 2005]).

De nombreuses références présentent les éléments fondamentaux de cette théorie des dioïdes. On peut notamment citer : [Cohen 1989], [Baccelli 1992], [Gaubert 1992] et [Heidergott 2006].

1.1 Définition et exemples

Définition 2.1 (Dioïde). Un dioïde (ou semi-anneau idempotent) est un ensemble $\mathcal{D}$ muni de deux lois notées $\oplus$ et $\otimes$ telles que $\forall a, b, c \in \mathcal{D}$:

- la loi $\oplus$ est
  - associative : $(a \oplus b) \oplus c = a \oplus (b \oplus c) = a \oplus b \oplus c$,
  - commutative : $a \oplus b = b \oplus a$,
  - idempotente : $a \oplus a = a$,
  - d'élément neutre noté ε : $a \oplus \varepsilon = a$,
- la loi $\otimes$ est
  - associative : $(a \otimes b) \otimes c = a \otimes (b \otimes c) = a \otimes b \otimes c$,
  - distributive à gauche et à droite par rapport à l'addition : $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$ et $(b \oplus c) \otimes a = (b \otimes a) \oplus (c \otimes a)$,
  - d'élément neutre noté e : $a \otimes e = e \otimes a = a$,
  - d'élément absorbant ε : $a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$.

De plus, un semi-anneau idempotent, couramment noté $(\mathcal{D}, \oplus, \otimes)$, est commutatif lorsque $\otimes$ est commutatif, soit lorsque $a \otimes b = b \otimes a$.

Remarque 2.2 Par analogie avec l'algèbre usuelle et lorsqu'il n'y aura aucune ambiguïté, le signe $\otimes$ pourra être omis. Ainsi $a \otimes b = ab$.

Exemple 2.3 (Dioïde $\mathbb{R}_{max}$). L'ensemble $(\mathbb{R} \cup \{-\infty\})$ muni de l'opération max notée $\oplus$ et de l'addition usuelle notée $\otimes$, correspond au semi-anneau idempotent $\mathbb{R}_{max}$ pour lequel $\varepsilon = -\infty$ et $e = 0$. Dans la littérature, ce dioïde est couramment appelé « algèbre (max,+) ».

Exemple 2.4 (Dioïde $\mathbb{R}_{min}$). L'ensemble $(\mathbb{R} \cup \{+\infty\})$ muni de l'opération min notée $\oplus$ et de l'addition usuelle notée $\otimes$, correspond au semi-anneau idempotent $\mathbb{R}_{min}$ pour lequel $\varepsilon = +\infty$ et $e = 0$. Ce dioïde est couramment appelé « algèbre (min,+) » ou encore « algèbre tropicale ».

Définition 2.5 (Sous-dioïde). Soit $(\mathcal{D}, \oplus, \otimes)$ un dioïde. Le sous-ensemble $\mathcal{C} \subset \mathcal{D}$ est un sous-dioïde de $(\mathcal{D}, \oplus, \otimes)$ si, et seulement 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 2.6 (Dioïdes $\mathbb{Z}_{max}$ et $\mathbb{Z}_{min}$). Les ensembles $\mathbb{Z}_{max} = (\mathbb{Z} \cup \{-\infty\}, \max, +)$ et $\mathbb{Z}_{min} = (\mathbb{Z} \cup \{-\infty\}, \min, +)$ sont respectivement des sous-dioïdes de $\mathbb{R}_{max}$ et $\mathbb{R}_{min}$.

1.2 Relation d'ordre et théorie des treillis

Afin de définir certaines propriétés des dioïdes et d'en exposer d'autres exemples, il est nécessaire de présenter quelques notions d'ordre et de treillis sur lesquelles repose amplement l'algèbre des dioïdes.

Définition 2.7 (Relation d'ordre). Une relation $\mathcal{R}$ sur un ensemble $\mathcal{S}$ est une relation d'ordre si les trois axiomes suivants sont vérifiés $\forall x, y, z \in \mathcal{S}$:

- $\mathcal{R}$ est réflexive : $x\mathcal{R}x$,
- $\mathcal{R}$ est transitive : si $x\mathcal{R}y$ et $y\mathcal{R}z$, alors $x\mathcal{R}z$,
- $\mathcal{R}$ est antisymétrique : si $x\mathcal{R}y$ et $y\mathcal{R}x$, alors $x = y$.

Définition 2.8 (Ensemble ordonné). Un ensemble ordonné est un ensemble $\mathcal{S}$ muni d'une relation d'ordre notée $\preccurlyeq$. Sur $\mathcal{S}$, l'ordre $\preccurlyeq$ est dit total si deux éléments quelconques de l'ensemble sont toujours comparables, il est dit partiel dans le cas contraire. En outre, les notions suivantes sont définies pour $\mathcal{S}$ et $\mathcal{U} \subset \mathcal{S}$:

majorant : $M \in \mathcal{S}$ est un majorant de $\mathcal{U}$ (mais n'appartient pas nécessairement à $\mathcal{U}$) si $\forall u \in \mathcal{U}, u \preccurlyeq M$,

minorant : $m \in \mathcal{S}$ est un minorant de $\mathcal{U}$ (mais n'appartient pas nécessairement à $\mathcal{U}$) si $\forall u \in \mathcal{U}, m \preccurlyeq u$,

borne supérieure : plus petit majorant noté $\bigvee \mathcal{U}$ ($= \sup\ \mathcal{U}$),

borne inférieure : plus grand minorant noté $\bigwedge \mathcal{U}$ ($= \inf\ \mathcal{U}$),

plus grand élément : $\top \in \mathcal{S}$ est le plus grand élément de $\mathcal{S}$ si $\forall x \in \mathcal{S}$, $x \preccurlyeq \top$. Cet élément est connu en anglais sous le nom « Top » et est nécessairement unique,

plus petit élément : $\bot \in \mathcal{S}$ est le plus petit élément de $\mathcal{S}$ si $\forall x \in \mathcal{S}$, $\bot \preccurlyeq x$. Cet élément est connu en anglais sous le nom « Bottom » et est nécessairement unique.

La théorie des treillis permet d'établir le lien entre les ensembles ordonnés définis ci-dessus et les dioïdes.

Définition 2.9 (Théorie des treillis). Parmi les ensembles ordonnés, on distingue :

sup-demi treillis : ensemble ordonné $\mathcal{P}$ dans lequel tout couple d'éléments (x, y) admet une borne supérieure, soit un plus petit majorant, notée $x \vee y$,

inf-demi treillis : ensemble ordonné $\mathcal{Q}$ dans lequel tout couple d'éléments (x, y) admet une borne inférieure, soit un plus grand minorant, notée $x \wedge y$,

treillis : ensemble ordonné $\mathcal{S}$ à la fois sup-demi treillis et inf-demi treillis [1],

sup-demi treillis complet : un sup-demi treillis $\mathcal{P}$ est complet s'il existe une borne supérieure $\bigvee \mathcal{U}$ pour tout sous-ensemble fini ou infini $\mathcal{U} \subset \mathcal{P}$,

inf-demi treillis complet : un inf-demi treillis $\mathcal{Q}$ est complet s'il existe une borne inférieure $\bigwedge \mathcal{V}$ pour tout sous-ensemble fini ou infini $\mathcal{V} \subset \mathcal{Q}$,

treillis complet : ensemble ordonné $\mathcal{S}$ à la fois sup-demi treillis complet et inf-demi treillis complet,

application isotone : application $\Pi : \mathcal{S} \mapsto \mathcal{T}$ définie sur des ensembles ordonnés telle qu'elle conserve l'ordre, *i.e.* $\forall a, b \in \mathcal{S}$, $a \preccurlyeq b \Rightarrow \Pi(a) \preccurlyeq \Pi(b)$.

Remarque 2.10 Un sup-demi treillis complet $\mathcal{P}$ a nécessairement un plus grand élément noté $\top = \bigvee_{x \in \mathcal{P}} x$. De même, un inf-demi treillis complet $\mathcal{Q}$ a toujours un plus petit élément noté $\bot = \bigwedge_{x \in \mathcal{Q}} x$.

Théorème 2.11 ([Baccelli 1992, Th. 4.27]). Un sup-demi treillis complet est un treillis complet si, et seulement si, il a un plus petit élément $\bot$. De même qu'un inf-demi treillis est complet si, et seulement si, il a un plus grand élément $\top$.

La théorie des treillis présentée ici a été introduite de façon ensembliste, en faisant uniquement intervenir les propriétés de la relation d'ordre $\preccurlyeq$ définie sur l'ensemble $\mathcal{S}$. Néanmoins, il est intéressant de faire le lien entre ce point de vue ensembliste et le point du vue algébrique comme le montre le théorème suivant.

Théorème 2.12 ([Blyth 2005, Th. 2.3]). Un ensemble ordonné $\mathcal{S}$ peut-être muni d'une structure de treillis si, et seulement si, il peut être doté de deux lois de composition internes $\vee$ et $\wedge$ telles que :

(i) les lois $\vee$ et $\wedge$ sont associatives, commutatives et idempotentes,

1. Autrement dit, tout ensemble fini de $\mathcal{S}$ admet une borne supérieure et inférieure.

(ii) $\forall x, y \in \mathcal{S},\ x \wedge (x \vee y) = x = x \vee (x \wedge y)$ (propriété d'absorption).

Ainsi, les lois $\vee$ et $\wedge$ d'un treillis $\mathcal{S}$ sont dites isotones selon l'ordre $\preccurlyeq$ soit $\forall x, y, z \in \mathcal{S}$:

$$x \preccurlyeq y \quad \Rightarrow \quad \begin{cases} x \vee z \preccurlyeq y \vee z, \\ x \wedge z \preccurlyeq y \wedge z, \end{cases}$$

et la notion d'ordre $x \preccurlyeq y$ peut être remplacée par les expressions $x = x \wedge y \Leftrightarrow x \vee y = y$.

1.3 Structure ordonnée d'un dioïde

Ces définitions d'ordre et de treillis ont un sens dans l'algèbre des dioïdes grâce à l'idempotence de la loi $\oplus$ qui leur confère naturellement une structure ordonnée.

Définition 2.13 (Relation d'ordre d'un dioïde). Un dioïde $(\mathcal{D}, \oplus, \otimes)$ est naturellement muni d'une relation d'ordre notée $\preccurlyeq$ et définie comme suit :

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

Cette relation d'ordre est compatible avec les lois $\oplus$ et $\otimes$ et détermine les propriétés d'isotonie suivantes $\forall a, b, c \in \mathcal{D}$:

$$a \preccurlyeq b \quad \Rightarrow \quad \begin{cases} a \oplus c \preccurlyeq b \oplus c, \\ a \otimes c \preccurlyeq b \otimes c \quad \text{et} \quad c \otimes a \preccurlyeq c \otimes b. \end{cases}$$

Remarque 2.14 (Ordre associé aux dioïdes $\mathbb{R}_{max}$, $\mathbb{Z}_{max}$, $\mathbb{R}_{min}$ et $\mathbb{Z}_{min}$). L'ordre $\preccurlyeq$ associé aux dioïdes $\mathbb{R}_{max}$ et $\mathbb{Z}_{max}$ coïncide avec l'ordre naturel $\leq$:

$$2 \preccurlyeq 3 \quad \Leftrightarrow \quad 2 \leq 3 \quad \Leftrightarrow \quad \max(2, 3) = 3,$$

alors qu'il correspond à l'ordre naturel inverse $\geq$ pour les dioïdes $\mathbb{R}_{min}$ et $\mathbb{Z}_{min}$:

$$3 \preccurlyeq 2 \quad \Leftrightarrow \quad 3 \geq 2 \quad \Leftrightarrow \quad \min(3, 2) = 2.$$

L'idempotence de la somme dans un dioïde induit une structure de sup-demi treillis pour lequel la borne supérieure est représentée par la loi $\oplus$ soit :

$$\forall a, b \in \mathcal{D}, \quad a \oplus b = a \vee b.$$

Ainsi, $a \oplus b$ est le plus petit majorant de a et b selon l'ordre $\preccurlyeq$ donné par la définition 2.13.

Définition 2.15 (Dioïde complet). Un dioïde est dit complet s'il est fermé pour les sommes infinies (*i.e.* les sommes infinies sont définies) et si la loi $\otimes$ distribue, à gauche comme à droite, sur ces sommes infinies. Soit, $\forall a \in \mathcal{D}$ et pour tout sous-ensemble $\mathcal{C} \subset \mathcal{D}$:

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

Il en résulte que $\forall \mathcal{A}, \mathcal{B} \subset \mathcal{D}$:

$$(\bigoplus_{a \in \mathcal{A}} a) \otimes (\bigoplus_{b \in \mathcal{B}} b) = \bigoplus_{(a,b) \in \mathcal{A} \times \mathcal{B}} (a \otimes b).$$

Puisqu'un dioïde a une structure de sup-demi treillis, il admet, s'il est complet, un plus grand élément noté $\top$ (cf. remarque 2.10) et vérifiant $\forall a \in \mathcal{D}$ et $a \neq \varepsilon$, $\top \oplus a = \top$. Soit par extension :

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

L'élément $\top$ correspond donc à la somme de tous les éléments du dioïde et respecte l'absorption de ε pour le produit : $\top \otimes \varepsilon = \varepsilon \otimes \top = \varepsilon$. En outre, d'après le théorème 2.11, l'existence d'un plus petit élément ε ($= \bot$ en référence aux notations des treillis) dans un dioïde complet [2] permet de dire qu'il possède une structure de treillis complet. Un dioïde complet est donc un treillis complet qui admet ainsi une borne inférieure représentée par la loi $\wedge$ associative, commutative, idempotente et isotone (cf théorème 2.12) et qui vérifie par conséquent :

$$\forall a, b \in \mathcal{D}, \quad a \preccurlyeq b \quad \Leftrightarrow \quad a \oplus b = b \quad \Leftrightarrow \quad a \wedge b = a. \tag{2.1}$$

L'élément $\top$ agit alors comme élément neutre sur cette loi : $\forall a \in \mathcal{D}, \top \wedge a = a$.

Remarque 2.16 La relation (2.1) peut faire penser que les lois $\oplus$ et $\wedge$ jouent un rôle symétrique dans un dioïde complet. Néanmoins, la loi $\otimes$ n'est pas nécessairement distributive par rapport à $\wedge$ et seule cette sous-distributivité est exacte grâce à

2. Soit un sup-demi treillis complet.

l'isotonie des lois $\otimes$ et $\wedge$, soit $\forall a \in \mathcal{D}$:

$$\begin{aligned} a \otimes (\bigwedge_{x\in\mathcal{C}} x) &\preccurlyeq \bigwedge_{x\in\mathcal{C}} (a \otimes x), \\ (\bigwedge_{x\in\mathcal{C}} x) \otimes a &\preccurlyeq \bigwedge_{x\in\mathcal{C}} (x \otimes a). \end{aligned}$$

Exemple 2.17 (Dioïdes $\overline{\mathbb{R}}_{max}$, $\overline{\mathbb{Z}}_{max}$, $\overline{\mathbb{R}}_{min}$ et $\overline{\mathbb{Z}}_{min}$). En ajoutant le plus grand élément $\top = +\infty$ aux dioïdes $\mathbb{R}_{max}$ et $\mathbb{Z}_{max}$, les dioïdes complets notés $\overline{\mathbb{R}}_{max}$ et $\overline{\mathbb{Z}}_{max}$ sont obtenus. De même qu'en ajoutant le plus grand élément $\top = -\infty$ aux dioïdes $\mathbb{R}_{min}$ et $\mathbb{Z}_{min}$ les dioïdes complets notés $\overline{\mathbb{R}}_{min}$ et $\overline{\mathbb{Z}}_{min}$ sont formés. Ainsi, pour la loi $\otimes$:

- sur $\overline{\mathbb{R}}_{max}$ et $\overline{\mathbb{Z}}_{max}$: $\top \otimes \varepsilon = +\infty \otimes -\infty = -\infty = \varepsilon$,
- sur $\overline{\mathbb{R}}_{min}$ et $\overline{\mathbb{Z}}_{min}$: $\top \otimes \varepsilon = -\infty \otimes +\infty = +\infty = \varepsilon$.

De par sa définition, la borne inférieure $\wedge$ correspond alors à l'opérateur min dans $\overline{\mathbb{R}}_{max}$ et $\overline{\mathbb{Z}}_{max}$, et à l'opérateur max dans $\overline{\mathbb{R}}_{min}$ et $\overline{\mathbb{Z}}_{min}$.

Définition 2.18 (Dioïde distributif). Un dioïde $(\mathcal{D}, \oplus, \otimes)$ est dit distributif s'il est complet et si $\forall a \in \mathcal{D}$ et pour tout sous-ensemble $\mathcal{C} \subset \mathcal{D}$:

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

Remarque 2.19 Si le dioïde n'est pas distributif, les inégalités suivantes sont néanmoins vérifiées grâce à l'isotonie des lois $\oplus$ et $\wedge$:

$$\begin{aligned} (\bigoplus_{x\in\mathcal{C}} x) \wedge a &\succcurlyeq \bigoplus_{x\in\mathcal{C}} (a \wedge x), \\ (\bigwedge_{x\in\mathcal{C}} x) \oplus a &\preccurlyeq \bigwedge_{x\in\mathcal{C}} (x \oplus a). \end{aligned}$$

2 Quelques exemples de dioïdes

D'autres formes de semi-anneau idempotents existent. Ceux utilisés dans ce mémoire et décrits ci-après sont les dioïdes de matrices, de fonctions, de séries formelles et les semi-anneaux quotientés par une relation d'équivalence.

2.1 Dioïde de matrices

Définition 2.20 (Dioïde de matrices). Soit $(\mathcal{D}, \oplus, \otimes)$ un dioïde. L'ensemble des matrices carrées de dimension n et à coefficients dans $\mathcal{D}$ forme le dioïde de matrices noté $\mathcal{D}^{n\times n}$ et dont les lois $\oplus$ et $\otimes$ sont définies comme suit avec $A, B \in \mathcal{D}^{n\times n}$ et $\forall i, j = [1, n]$:

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

L'élément neutre pour la loi $\oplus$ est noté ε et correspond à la matrice dont tous les coefficients sont égaux à ε. L'élément neutre pour la loi $\otimes$ (la matrice identité) est quant à lui noté e (ou E lorsqu'une ambiguïté est possible avec le scalaire e). Il correspond à la matrice dont tous les coefficients sont égaux à ε, à l'exception de ceux de la diagonale qui sont égaux à e.

Si $\mathcal{D}$ est complet et/ou distributif, alors $\mathcal{D}^{n\times n}$ l'est également. Tandis que si $\mathcal{D}$ est commutatif, ce n'est pas obligatoirement vrai pour $\mathcal{D}^{n\times n}$.

Remarque 2.21 La somme et le produit de deux matrices compatibles mais non nécessairement carrées sont également possibles. Ces opérations peuvent être définies de la manière suivante avec $A, B, C \in \mathcal{D}^{n\times p}$ et $D \in \mathcal{D}^{p\times q}$:

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

Afin de manipuler ces matrices en tant qu'éléments d'un dioïde de matrices, il faut considérer qu'elles appartiennent au dioïde des matrices carrées de dimension $\max(n, p, q) \times \max(n, p, q)$, en les complétant si nécessaire par des lignes et/ou colonnes constituées de l'élément ε.

2.2 Dioïde de fonctions

Définition 2.22 (Dioïde de fonctions). L'ensemble des fonctions f définies de $\mathbb{R}$ dans un dioïde $\mathcal{D}$ et munies des lois $\oplus$ et $\otimes$ telles que $\forall a \in \mathbb{R}$:

$$\begin{aligned} f \oplus g: \quad (f \oplus g)(a) &\triangleq f(a) \oplus g(a), \\ f \otimes g: \quad (f \otimes g)(a) &\triangleq f(a) \otimes g(a), \end{aligned}$$

forme le dioïde de fonctions noté $(\mathcal{D}^{\mathbb{R}}, \oplus, \otimes)$.

Si $\mathcal{D}$ est complet, alors $\mathcal{D}^{\mathbb{R}}$ l'est également. En outre, les lois d'un dioïde de fonctions sont isotones, et conservent les axiomes donnés dans la définition 2.1.

Exemple 2.23 (Dioïde de fonctions $(\overline{\mathbb{R}}_{min}^{\mathbb{R}}, \oplus, \otimes)$). L'ensemble des fonctions $f : \mathbb{R} \mapsto \overline{\mathbb{R}}_{min}$ munies du minimum point à point comme somme $\oplus$:

$$\forall a \in \mathbb{R},\ (f \oplus g)(a) = \min\{f(a), g(a)\},$$

et de l'addition point à point comme produit $\otimes$:

$$\forall a \in \mathbb{R},\ (f \otimes g)(a) = f(a) + g(a),$$

forme le dioïde de fonctions noté $(\overline{\mathbb{R}}_{min}^{\mathbb{R}}, \oplus, \otimes)$.

2.3 Dioïde de séries formelles

Définition 2.24 (Série formelle). Soit $(\mathcal{D}, \oplus, \otimes)$ un dioïde. Une série formelle en l'indéterminée notée z et à coefficients dans $\mathcal{D}$, est l'application s définie de $\mathbb{Z}$ dans $\mathcal{D}$ et 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 :

$$s = \bigoplus_{k \in \mathbb{Z}} s(k) z^k = s(0) \oplus s(1) z \oplus s(2) z^2 \oplus \ldots \oplus s(k) z^k \oplus \ldots \ .$$

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

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

Remarque 2.25 (Série formelle en plusieurs indéterminées). Si la série formelle s a p indéterminées notées z_1 à z_p (nécessairement commutatives soit $z_1 z_2 = z_2 z_1$), elle est alors représentée par l'expression suivante :

$$s = \bigoplus_{k \in \mathbb{Z}^p} s(k_1, \ldots, k_p) z_1^{k_1} \ldots z_p^{k_p},$$

avec :

$$\mathsf{Supp}(s) = \{(k_1, \ldots, k_p) \in \mathbb{Z}^p \mid s(k_1, \ldots, k_p) \neq \varepsilon\}.$$

Remarque 2.26 (Monôme et polynôme). Une série formelle à support fini (c'est-à-dire une série avec un nombre fini de coefficients non nuls) est appelée

polynôme alors qu'une série formelle dont le support est un singleton est appelée monôme.

Définition 2.27 (Dioïde de séries formelles). L'ensemble des séries formelles à une indéterminée z, à coefficients dans $\mathcal{D}$, et munies des opérations suivantes :

$$\begin{array}{lrcl} s_a \oplus s_b : & (s_a \oplus s_b)(k) & \triangleq & s_a(k) \oplus s_b(k), \\ s_a \otimes s_b : & (s_a \otimes s_b)(k) & \triangleq & \bigoplus\limits_{k=k_a+k_b} (s_a(k_a) \otimes s_b(k_b)), \end{array} \tag{2.2}$$

forme le dioïde de séries formelles noté $\mathcal{D}[\![z]\!]$.

Exemple 2.28 (Dioïde $\overline{\mathbb{R}}_{min}[\![\delta]\!]$). Le dioïde $\overline{\mathbb{R}}_{min}[\![\delta]\!]$ est le dioïde des séries formelles en l'indéterminée δ, à coefficients dans $\overline{\mathbb{R}}_{min}$ et à exposants dans $\mathbb{R}$. Une série s appartenant à $\overline{\mathbb{R}}_{min}[\![\delta]\!]$ s'écrit :

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

Les lois $\oplus$ et $\otimes$ de $\overline{\mathbb{R}}_{min}[\![\delta]\!]$ sont alors définies par :

$$\begin{aligned} s_a \oplus s_b &= \bigoplus_{t\in\mathbb{R}} \min(s_a(t), s_b(t))\delta^t, \\ s_a \otimes s_b &= \bigotimes_{t=t_a+t_b} (s_a(t_a) + s_b(t_b))\delta^t. \end{aligned}$$

Remarque 2.29 (Dioïde de séries formelles en plusieurs indéterminées). L'ensemble des séries formelles à p indéterminées commutatives z_1 à z_p, à coefficients dans $\mathcal{D}$ et munies des opérations données par (2.2), forme le dioïde de séries formelles noté $\mathcal{D}[\![z_1, \ldots, z_p]\!]$.

Exemple 2.30 (Dioïde $\mathbb{B}[\![\gamma, \delta]\!]$). Le dioïde de séries formelles en deux indéterminées commutatives [3] γ et δ, à coefficients booléens dans $\mathbb{B}$ et à exposants dans $\mathbb{Z}$ est noté $\mathbb{B}[\![\gamma, \delta]\!]$. La série $s \in \mathbb{B}[\![\gamma, \delta]\!]$ s'écrit de manière unique :

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

Le dioïde $\mathbb{B}[\![\gamma, \delta]\!]$ est un dioïde complet, commutatif et distributif. Il est muni des lois usuelles sur les séries formelles ainsi que des éléments neutres suivants :

3. Soit $\gamma^2\delta^1 = \gamma^1\delta^1\gamma^1 = \delta^1\gamma^2$.

· pour la loi $\oplus$: $\varepsilon(\gamma,\delta) = \bigoplus_{k,t\in\mathbb{Z}} \varepsilon\gamma^k\delta^t$ (série nulle),

· pour la loi $\otimes$: $e(\gamma,\delta) = \gamma^0\delta^0$.

Graphiquement, une série s de $\mathbb{B}[\![\gamma,\delta]\!]$ peut être représentée dans le plan $\mathbb{Z}^2$, avec pour axe horizontal l'indéterminée γ et pour axe vertical l'indéterminée δ, par la collection des points (n,t) appartenant à son support $\mathsf{Supp}(s)$. En effet, les coefficients booléens peuvent être traduits comme synonymes de présence ($s(n,t) = e$) ou d'absence ($s(n,t) = \varepsilon$) d'un point.

Exemple 2.31 Les points de coordonnées $(2,3)$, $(3,4)$, $(5,7)$ et $(6,5)$ de $\mathbb{Z}^2$ permettent la représentation de la série $s = \gamma^2\delta^3 \oplus \gamma^3\delta^4 \oplus \gamma^5\delta^7 \oplus \gamma^6\delta^5 \in \mathbb{B}[\![\gamma,\delta]\!]$ (cf. figure 2.1). Cette série ayant un nombre fini de monômes $\gamma^n\delta^t$, elle correspond à un polynôme de $\mathbb{B}[\![\gamma,\delta]\!]$.

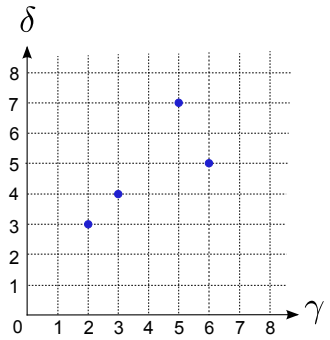


Figure 2.1 Représentation graphique de la série $s = \gamma^2\delta^3 \oplus \gamma^3\delta^4 \oplus \gamma^5\delta^7 \oplus \gamma^6\delta^5 \in \mathbb{B}[\![\gamma,\delta]\!]$.

2.4 Dioïde quotient

Définition 2.32 (Homomorphisme, isomorphisme). Une application Π d'un dioïde $\mathcal{D}$ dans un dioïde $\mathcal{C}$ est un homomorphisme si $\forall a,b \in \mathcal{D}$:

$$\Pi(a\oplus b) = \Pi(a)\oplus\Pi(b) \quad \text{et} \quad \Pi(\varepsilon) = \varepsilon, \tag{2.3}$$

$$\Pi(a\otimes b) = \Pi(a)\otimes\Pi(b) \quad \text{et} \quad \Pi(e) = e. \tag{2.4}$$

Une application vérifiant seulement la propriété (2.3) est dite $\oplus$-morphisme alors qu'une application vérifiant uniquement (2.4) est dite $\otimes$-morphisme. Enfin, un

homomorphisme bijectif[4] est appelé isomorphisme.

Définition 2.33 (Congruence). Dans un dioïde $\mathcal{D}$, une congruence est une relation d'équivalence[5] notée $\equiv$ et compatible avec les lois $\oplus$ et $\otimes$, soit $\forall a, b, c \in \mathcal{D}$:

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

Définition 2.34 (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 $\equiv$:

$$[a]_{\equiv} \triangleq \{x \in \mathcal{D} \mid x \equiv a\}.$$

Lemme 2.35 (Congruence induite par un homomorphisme [Baccelli 1992, Cor. 4.26]). Si l'application $\Pi : \mathcal{D} \mapsto \mathcal{C}$ est un homomorphisme, alors la relation $\overset{\Pi}{\equiv}$ définie ci-dessous $\forall a, b \in \mathcal{D}$ est une congruence :

$$\Pi(a) = \Pi(b) \quad \Leftrightarrow \quad a \overset{\Pi}{\equiv} b.$$

Lemme 2.36 (Dioïde quotient [Baccelli 1992, Lem. 4.24]). Le quotient d'un dioïde $\mathcal{D}$ par une congruence $\equiv$ est un dioïde noté $\mathcal{D}_{/\equiv}$ et muni des lois $\oplus$ et $\otimes$ définies comme suit :

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

Un exemple de dioïde quotient est le dioïde $\mathbb{B}[\![\gamma, \delta]\!]_{/\gamma^\star(\delta^{-1})^\star}$, noté plus simplement $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ et présenté ci-dessous. La modélisation des systèmes (min,+)-linéaires réalisée grâce à ce dioïde est décrite dans l'annexe A. La congruence spécifique $\gamma^\star(\delta^{-1})^\star$ est utilisée en réponse à la nécessité de prendre en

4. Tout élément de $\mathcal{C}$ admet un et un seul antécédent dans $\mathcal{D}$.

5. Une relation d'équivalence est une relation $\equiv$ réflexive, transitive et symétrique (*i.e.* si $x \equiv y$ alors $y \equiv x$).

compte le caractère monotone croissant des trajectoires de ces systèmes.

Définition 2.37 (Dioïde $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$). Le quotient du dioïde $\mathbb{B}[\![\gamma,\delta]\!]$ par la congruence $\equiv$ définie comme suit $\forall a,b \in \mathbb{B}[\![\gamma,\delta]\!]$:

$$a \equiv b \quad \Leftrightarrow \quad \gamma^\star(\delta^{-1})^\star a = \gamma^\star(\delta^{-1})^\star b,$$

est le dioïde noté $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!] = \mathbb{B}[\![\gamma,\delta]\!]_{/\gamma^\star(\delta^{-1})^\star}$ et pour lequel l'opérateur *, appelé étoile de Kleene, est défini comme suit [6] :

$$a^\star = \bigoplus_{i=0}^{+\infty} a^i \quad \text{avec} \quad a^{i+1} = a \otimes a^i \quad \text{et} \quad a^0 = e.$$

Ainsi, $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ est un dioïde complet, commutatif et distributif. Il est muni des lois usuelles sur les séries formelles, ainsi que des éléments neutres suivants :

- pour la loi $\oplus$: $\varepsilon = \varepsilon(\gamma,\delta)$ (élément neutre de $\mathbb{B}[\![\gamma,\delta]\!]$),
- pour la loi [7] $\otimes$: $e = \gamma^\star(\delta^{-1})^\star = \gamma^\star = (\delta^{-1})^\star = \gamma^0 = \delta^0 = \gamma^0\delta^0$.

L'élément $\top$ est donné par : $\top = (\gamma^{-1}\delta)^\star$. Un élément du dioïde quotient $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ est donc une classe d'équivalence notée $[a]_{\gamma^\star(\delta^{-1})^\star}$ regroupant tous les éléments de $\mathbb{B}[\![\gamma,\delta]\!]$ équivalents modulo $\gamma^\star(\delta^{-1})^\star$.

Remarque 2.38 (Simplification d'écriture). Un abus d'écriture peut éventuellement être fait quant aux plus petit et plus grand éléments de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ par les élément suivants : $\varepsilon = \gamma^{+\infty}\delta^{-\infty}$ et $\top = \gamma^{-\infty}\delta^{+\infty}$. De même, 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 $[\cdot]_{\gamma^\star(\delta^{-1})^\star}$. Les éléments manipulés sont donc tacitement des classes d'équivalence, bien que les écritures employées ne fassent figurer que des représentants de ces classes.

Graphiquement, le produit d'un monôme $\gamma^n\delta^t$ de $\mathbb{B}[\![\gamma,\delta]\!]$ par $\gamma^\star(\delta^{-1})^\star$ revient à considérer cet élément non plus comme un point (n,t) de $\mathbb{Z}^2$ mais comme un cône Sud-Est de sommet de coordonnées (n,t). En effet, la présence de $\gamma^\star$ permet au monôme de se reproduire depuis son origine jusqu'à $\gamma^{+\infty}$ sur l'axe horizontal :

$$\gamma^\star = e \oplus \gamma^1 \oplus \gamma^2 \oplus \gamma^3 \oplus \dots ,$$

alors que $(\delta^{-1})^\star$ induit une reproduction de ces points vers $\delta^{-\infty}$ sur l'axe vertical :

$$(\delta^{-1})^\star = e \oplus \delta^{-1} \oplus \delta^{-2} \oplus \delta^{-3} \oplus \dots .$$

6. Cet opérateur est présenté plus en détail dans la section 3.3.2 de ce chapitre.

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

Exemple 2.39 Soient s_1 et s_2, deux polynômes de $\mathbb{B}[\![\gamma,\delta]\!]$ définis par :

$$\begin{aligned} s_1 &= \gamma^2\delta^3 \oplus \gamma^3\delta^2 \oplus \gamma^5\delta^6, \\ s_2 &= \gamma^2\delta^3 \oplus \gamma^5\delta^6. \end{aligned}$$

Les produits $\gamma^\star(\delta^{-1})^\star s_1$ et $\gamma^\star(\delta^{-1})^\star s_2$ couvrent la même surface dans le plan $\mathbb{Z}^2$ (cf. figure 2.2). Le monôme $\gamma^3\delta^2$ est absorbé par le cône Sud-Est du monôme $\gamma^2\delta^3$, cône issu du produit de $\gamma^2\delta^3$ avec $\gamma^\star(\delta^{-1})^\star$. Ces séries s_1 et s_2 de $\mathbb{B}[\![\gamma,\delta]\!]$ appartiennent donc à la même classe d'équivalence dans $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$:

$$[s_1]_{\gamma^\star(\delta^{-1})^\star} = [s_2]_{\gamma^\star(\delta^{-1})^\star}.$$

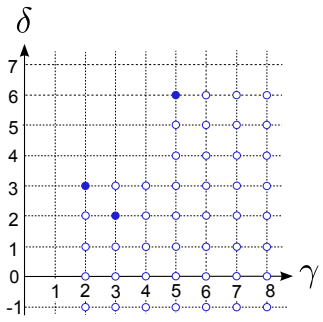


Figure 2.2 Représentation graphique des séries $\gamma^\star(\delta^{-1})^\star s_1 = \gamma^\star(\delta^{-1})^\star s_2$.

Propriété 2.40 (Représentant maximal). Dans chaque classe d'équivalence de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$, il existe un élément maximal, somme de tous les éléments de cette classe. Ce représentant maximal appartient à $\mathbb{B}[\![\gamma,\delta]\!]$ et est obtenu en multipliant un représentant quelconque de la classe par $\gamma^\star(\delta^{-1})^\star$. Graphiquement, dans $\mathbb{Z}^2$, l'élément maximal correspond à tous les points de $\mathbb{Z}^2$ situés dans les cônes sud-est engendrés par les monômes $\gamma^{n_i}\delta^{t_i}$. Dans l'exemple 2.39, le représentant maximal s_{max} de $s_1 = \gamma^2\delta^3 \oplus \gamma^3\delta^2 \oplus \gamma^5\delta^6$ est donné par :

$$s_{max} = ((\gamma^\star(\delta^{-1})^\star)(\gamma^2\delta^3 \oplus \gamma^3\delta^2 \oplus \gamma^5\delta^6).$$

Propriété 2.41 (Représentant minimal). Certains éléments de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ tels que les polynômes, admettent également un représentant minimal dans $\mathbb{B}[\![\gamma,\delta]\!]$. Graphiquement, ce représentant minimal est obtenu en ne codant que les sommets du

représentant maximal de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$. Dans l'exemple 2.39, le représentant minimal s_{min} de $s_1 = \gamma^2\delta^3 \oplus \gamma^3\delta^2 \oplus \gamma^5\delta^6$ est donné par :

$$s_{min} = \gamma^2\delta^3 \oplus \gamma^5\delta^6.$$

Le représentant minimal d'une classe de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ étant simple à écrire, il est toujours choisi pour présenter les résultats.

3 Résolution d'équations dans un dioïde

Dans la suite de ce mémoire, plusieurs types de fonctions définies sur les dioïdes sont considérés, fonctions qui généralement ne sont pas inversibles. Cependant, grâce à la théorie de la résiduation, il est possible de considérer des « pseudo-inverses » d'applications isotones et ainsi de résoudre des équations du type $f(x) = b$. Par ailleurs, l'étude des points fixes d'applications définies sur des treillis a également fait l'objet de plusieurs recherches qui bénéficient à la théorie des dioïdes et permettent notamment la résolution d'équations implicites du type $f(x) = x$. Ces résolutions d'équations nécessitent au préalable la définition des notions d'isotonie et de continuité.

3.1 Applications isotones et continues

Définition 2.42 (Isotonie, antitonie, monotonie). Soient $\mathcal{D}$ et $\mathcal{C}$ deux dioïdes complets. Une application $\Pi : \mathcal{D} \mapsto \mathcal{C}$ est dite isotone [8] si $\forall a, b \in \mathcal{D}$:

$$a \preccurlyeq b \quad \Rightarrow \quad \Pi(a) \preccurlyeq \Pi(b),$$

et antitone si :

$$a \preccurlyeq b \quad \Rightarrow \quad \Pi(a) \succcurlyeq \Pi(b).$$

En outre, une application est dite monotone si elle est isotone ou antitone.

Remarque 2.43 La composition de deux applications isotones demeure isotone tandis que la composition d'applications antitones est isotone.

8. Plus communément, une application isotone conserve l'ordre. Elle sert alors à caractériser les fonctions croissantes ou décroissantes.

Remarque 2.44 Si l'application $\Pi : \mathcal{D} \mapsto \mathcal{C}$ est un homomorphisme, alors Π est isotone. En effet :

$$a \preccurlyeq b \quad \Leftrightarrow \quad a \oplus b = b \quad \Rightarrow \quad \Pi(a \oplus b) = \Pi(a) \oplus \Pi(b) = \Pi(b) \quad \Leftrightarrow \quad \Pi(a) \preccurlyeq \Pi(b).$$

En revanche, une application isotone n'est pas nécessairement $\oplus$-morphisme et vérifie seulement $\forall a, b \in \mathcal{D}$ que $\Pi(a \oplus b) \succcurlyeq \Pi(a) \oplus \Pi(b)$ (en considérant que $a \oplus b \succcurlyeq a$ et $a \oplus b \succcurlyeq b$, puisque Π est isotone : $\Pi(a) \oplus \Pi(b) \succcurlyeq \Pi(a)$ et $\Pi(a) \oplus \Pi(b) \succcurlyeq \Pi(b)$).

Ensuite, la notion de continuité caractérise les applications pour lesquelles la préservation de l'ordre dans leur image est effective.

Définition 2.45 (Continuité). Soient $\mathcal{D}$ et $\mathcal{C}$ deux dioïdes complets. Une application $\Pi : \mathcal{D} \mapsto \mathcal{C}$ est dite semi-continue inférieurement (s.c.i. en abrégé) si $\forall \mathcal{B} \subset \mathcal{D}$:

$$\Pi(\bigoplus_{x \in \mathcal{B}} x) \quad = \quad \bigoplus_{x \in \mathcal{B}} \Pi(x),$$

et semi-continue supérieurement (s.c.s.) si :

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

L'application Π est dite continue si elle est à la fois s.c.i et s.c.s.

Remarque 2.46 Une application s.c.i. ou s.c.s est nécessairement isotone puisque :

$$a \preccurlyeq b \;\Leftrightarrow \begin{cases} a \oplus b = b \quad \Rightarrow \quad \Pi(a \oplus b) = \Pi(a) \oplus \Pi(b) = \Pi(b) \quad \Leftrightarrow \quad \Pi(a) \preccurlyeq \Pi(b), \\ a \wedge b = a \quad \Rightarrow \quad \Pi(a \wedge b) = \Pi(a) \wedge \Pi(b) = \Pi(a) \quad \Leftrightarrow \quad \Pi(a) \preccurlyeq \Pi(b). \end{cases}$$

En outre, une application s.c.i. est un $\oplus$-morphisme.

3.2 Théorie de la résiduation

La théorie de la résiduation, initialement destinée aux treillis (cf. [Blyth 2005]), peut être étendue aux applications f semi-continues inférieures définies sur des dioïdes. Elle montre alors que si l'ensemble des solutions de l'inéquation $f(x) \preccurlyeq b$ est non vide, la borne supérieure de cet ensemble est considérée comme la plus

grande solution [9]. Il faut pour cela vérifier que cette borne est elle-même solution, ce qui impose la semi-continuité inférieure de f. Une recherche similaire peut être appliquée pour la détermination de la plus petite solution, lorsqu'elle existe, de l'inéquation $f(x) \succcurlyeq b$.

Définition 2.47 (Application résiduable et sa résiduée). Une application isotone $f : \mathcal{D} \mapsto \mathcal{C}$ définie sur des dioïdes complets est dite résiduable si $\forall b \in C$, l'équation $f(x) \preccurlyeq b$ admet une plus grande solution dans $\mathcal{D}$. L'application qui associe à b la plus grande solution de $f(x) \preccurlyeq b$ est notée $f^\sharp$ et est appelée application résiduée de f. Ainsi :

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

Remarque 2.48 La théorie de la résiduation permet également de considérer les équations du type $f(x) \succcurlyeq b$ définies de $\mathcal{D}$ dans $\mathcal{C}$. Dans ce cas, l'application f est dite dualement résiduable si $\forall b \in \mathcal{C}$, $f(x) \succcurlyeq b$ admet une plus petite solution dans $\mathcal{D}$. L'application qui associe à b la plus petite solution de $f(x) \succcurlyeq b$ est notée $f^\flat$ et est dite application résiduée duale de f. Ces applications ne sont pas considérées dans ce mémoire, le lecteur intéressé pourra consulter [Blyth 2005].

Le théorème suivant fournit une caractérisation de ces applications résiduables.

Théorème 2.49 ([Baccelli 1992, Th. 4.50]). Soit une application isotone $f : \mathcal{D} \mapsto \mathcal{C}$ définie sur des dioïdes complets. Les points suivants sont équivalents :

(i) f est résiduable,

(ii) f est semi-continue inférieurement (s.c.i) et $f(\varepsilon) = \varepsilon$,

(iii) il existe une application isotone et semi-continue supérieurement $f^\sharp : \mathcal{C} \mapsto \mathcal{D}$ telle que :

$$\begin{aligned} f \circ f^\sharp &\preccurlyeq \mathsf{Id}_{\mathcal{C}}, \\ f^\sharp \circ f &\succcurlyeq \mathsf{Id}_{\mathcal{D}}, \end{aligned}$$

où $\mathsf{Id}_{\mathcal{C}}$ (respectivement $\mathsf{Id}_{\mathcal{D}}$) est l'application identité de $\mathcal{C}$ (respectivement de $\mathcal{D}$). Par conséquent, $f^\sharp$ est unique.

9. Afin de garantir l'existence d'une telle borne, les dioïdes manipulés sont supposés complets. Cette plus grande solution correspond alors à la borne supérieure des « sous-solutions » de l'équation $f(x) = b$.

3.2.1 Résolution de $ax \preccurlyeq b$ et $xa \preccurlyeq b$

On se propose maintenant d'étudier le caractère résiduable de certaines applications de référence définies sur des dioïdes complets.

Soient L_a et R_a, également appelée *produit à gauche* et *produit à droite*, les applications suivantes définies sur un dioïde complet $\mathcal{D}$:

$$L_a \quad : \quad x \mapsto a \otimes x, \tag{2.5}$$

$$R_a \quad : \quad x \mapsto x \otimes a. \tag{2.6}$$

D'après la définition 2.15 d'un dioïde complet, le produit distribue à gauche comme à droite sur les sommes infinies. Les applications L_a et R_a sont donc semi-continues inférieurement. De plus, ε est absorbant pour le produit, ainsi $L_a(\varepsilon) = a \otimes \varepsilon = \varepsilon$ et $R_a(\varepsilon) = \varepsilon \otimes a = \varepsilon$. En appliquant le théorème 2.49, L_a et R_a sont donc résiduables, et leurs applications résiduées sont notées :

$$L_a^\sharp \quad : \quad x \mapsto a \backslash x = \frac{x}{a}, \tag{2.7}$$

$$R_a^\sharp \quad : \quad x \mapsto x / a = \frac{x}{a}, \tag{2.8}$$

et respectivement appelées *quotient à gauche* et *quotient à droite*. Ainsi, $L_a^\sharp(b) = a \backslash b$ et $R_a^\sharp(b) = b / a$ sont les plus grandes solutions des inégalités [10] $ax \preccurlyeq b$ et $xa \preccurlyeq b$, *i.e.* :

$$a \backslash b \quad = \quad \bigoplus \{x \mid ax \preccurlyeq b\}, \tag{2.9}$$

$$b / a \quad = \quad \bigoplus \{x \mid xa \preccurlyeq b\}. \tag{2.10}$$

Donc :

$$ax \preccurlyeq b \quad \Leftrightarrow \quad x \preccurlyeq a \backslash b, \tag{2.11}$$

$$xa \preccurlyeq b \quad \Leftrightarrow \quad x \preccurlyeq b / a. \tag{2.12}$$

Remarque 2.50 Lorsque $\mathcal{D}$ est commutatif, $L_a = R_a$. Cela implique donc également que $L_a^\sharp = R_a^\sharp$.

De nombreuses propriétés sont associées à ces quotients $L_a^\sharp$ et $R_a^\sharp$. Elles sont rappelées dans le théorème suivant.

Théorème 2.51 Les applications $L_a^\sharp$ et $R_a^\sharp$ vérifient les propriétés suivantes $\forall x, y, a, b \in \mathcal{D}$ avec $\mathcal{D}$ un dioïde complet. Tout d'abord, pour le quotient à gauche

10. Rappelons que le signe $\otimes$ peut être omis.

$\backslash$:

$$x \preccurlyeq y \quad \Rightarrow \quad \begin{cases} a \backslash x \preccurlyeq a \backslash y & (x \mapsto a \backslash x \text{ est isotone}), \\ x \backslash a \succcurlyeq y \backslash a & (x \mapsto x \backslash a \text{ est antitone}). \end{cases} \tag{2.13}$$

De même pour le quotient à droite $/$:

$$x \preccurlyeq y \quad \Rightarrow \quad \begin{cases} x / a \preccurlyeq y / a & (x \mapsto x / a \text{ est isotone}), \\ a / x \succcurlyeq a / y & (x \mapsto a / x \text{ est antitone}). \end{cases} \tag{2.14}$$

Enfin :

$$a \backslash \top = \top \qquad \top / a = \top, \tag{2.15}$$
$$a(a \backslash x) \preccurlyeq x \qquad (x/a)a \preccurlyeq x, \tag{2.16}$$
$$a \backslash (ax) \succcurlyeq x \qquad (xa)/a \succcurlyeq x, \tag{2.17}$$
$$a(a \backslash (ax)) = ax \qquad ((xa)/a)a = xa, \tag{2.18}$$
$$a \backslash (a(a \backslash x)) = a \backslash x \qquad ((x/a)a)/a = x/a, \tag{2.19}$$
$$a \backslash (x \wedge y) = a \backslash x \wedge a \backslash y \qquad (x \wedge y)/a = x/a \wedge y/a, \tag{2.20}$$
$$a \backslash (x \oplus y) \succcurlyeq (a \backslash x) \oplus (a \backslash y) \qquad (x \oplus y)/a \succcurlyeq (x/a) \oplus (y/a), \tag{2.21}$$
$$(a \wedge b) \backslash x \succcurlyeq (a \backslash x) \oplus (b \backslash x) \qquad x/(a \wedge b) \succcurlyeq (x/a) \oplus (x/b), \tag{2.22}$$
$$(a \oplus b) \backslash x = a \backslash x \wedge b \backslash x \qquad x/(a \oplus b) = x/a \wedge x/b, \tag{2.23}$$
$$(ab) \backslash x = b \backslash (a \backslash x) \qquad x/(ba) = (x/a)/b, \tag{2.24}$$
$$(a \backslash x)b \preccurlyeq a \backslash (xb) \qquad b(x/a) \preccurlyeq (bx)/a, \tag{2.25}$$
$$b(a \backslash x) \preccurlyeq (a/b) \backslash x \qquad (x/a)b \preccurlyeq x/(b \backslash a). \tag{2.26}$$

Preuve. Les preuves de ces propriétés sont données dans la littérature. Se référer à [MaxPlus 1991] pour les équations (2.13) et (2.14), et à [Baccelli 1992, §4.4.4], [Gaubert 1992, §5.3] et [Cottenceau 1999a, §1.3.3, Th. 1.95] pour les équations (2.15) à (2.26). □

3.2.2 Extension aux dioïdes de matrices : $AX \preccurlyeq B$ et $XA' \preccurlyeq B'$

Concernant le cas matriciel, définissons comme suit les applications produit à gauche $L_A : \mathcal{D}^{n\times p} \mapsto \mathcal{D}^{m\times p}$ et produit à droite $R_{A'} : \mathcal{D}^{p\times n} \mapsto \mathcal{D}^{p\times m}$ sur les dioïdes de matrices à coefficients dans le dioïde complet $\mathcal{D}$:

$$L_A \quad : \quad X \mapsto A \otimes X, \tag{2.27}$$
$$R_{A'} \quad : \quad X \mapsto X \otimes A'. \tag{2.28}$$

Dans ces applications, $A \in \mathcal{D}^{m\times n}$ et $A' \in \mathcal{D}^{n\times m}$.

Les plus grandes solutions des inéquations $AX \preccurlyeq B$ et $XA' \preccurlyeq B'$, pour lesquelles $B \in \mathcal{D}^{m\times p}$ et $B' \in \mathcal{D}^{p\times m}$, sont respectivement la résiduée du produit à gauche notée $L^\sharp_A(B) = A\backslash B$ et la résiduée du produit à droite notée $R^\sharp_{A'}(B') = B'/A'$. Les valeurs de ces matrices sont alors données comme suit :

$$L^\sharp_A(B) = A\backslash B \quad : \quad (A\backslash B)_{ij} = \bigwedge_{k=1}^{m} A_{ki}\backslash B_{kj}, \tag{2.29}$$

$$R^\sharp_{A'}(B') = B'/A' \quad : \quad (B'/A')_{ij} = \bigwedge_{k=1}^{m} B'_{ik}/A'_{jk}. \tag{2.30}$$

Remarque 2.52 En toute rigueur, seul le produit de matrices carrées constitue une application résiduable. Cependant, il est toujours possible de réécrire les applications L_A et $R_{A'}$ dans le dioïde de matrices $\mathcal{D}^{q\times q}$ pour lequel $q = \max(m, n, p)$. Les lignes et colonnes doivent alors être complétées par l'élément ε, ce qui peut compliquer les écritures.

3.2.3 Résiduation de l'application injection canonique

Selon la théorie de la résiduation, l'application injection canonique définie ci-dessous est résiduable et sa résiduée est un projecteur d'un dioïde dans un autre.

Définition 2.53 (Injection canonique). Soit $\mathcal{D}_{sub}$ un sous-dioïde du dioïde $\mathcal{D}$, avec $\mathcal{D}_{sub}$ et $\mathcal{D}$ complets. L'injection canonique de $\mathcal{D}_{sub}$ dans $\mathcal{D}$ est l'application notée $\mathsf{I}_{\mathcal{D}_{sub}}$ et définie comme suit pour tout $x \in \mathcal{D}_{sub}$:

$$\mathsf{I}_{\mathcal{D}_{sub}}(x) = x.$$

Théorème 2.54 (Lemme de projection [Gaubert 1992]). L'injection canonique $\mathsf{I}_{\mathcal{D}_{sub}} : \mathcal{D}_{sub} \mapsto \mathcal{D}$, avec $\mathcal{D}_{sub}$ et $\mathcal{D}$ complets, est résiduable. Sa résiduée $\mathsf{I}^\sharp_{\mathcal{D}_{sub}}$ est notée plus simplement $\mathsf{Pr}_{\mathcal{D}_{sub}}$ et vérifie :

(i) $\mathsf{Pr}_{\mathcal{D}_{sub}} \circ \mathsf{Pr}_{\mathcal{D}_{sub}} = \mathsf{Pr}_{\mathcal{D}_{sub}}$,

(ii) $\mathsf{Pr}_{\mathcal{D}_{sub}} \preccurlyeq \mathsf{Id}_{\mathcal{D}}$,

(iii) $x \in \mathcal{D}_{sub} \Leftrightarrow \mathsf{Pr}_{\mathcal{D}_{sub}}(x) = x$.

L'application $\mathsf{Pr}_{\mathcal{D}_{sub}}$ est appelée projecteur de $\mathcal{D}$ dans $\mathcal{D}_{sub}$.

Remarque 2.55 (Projecteur). Il est également possible de définir des projecteurs d'applications d'un dioïde dans lui-même. En effet, soit Π une application de $\mathcal{D}$

dans $\mathcal{D}$, lorsque Π vérifie $\Pi = \Pi \circ \Pi$, l'application est appelée projecteur.

3.3 Equations aux points fixes

Grâce à leur structure de treillis, il est possible d'appliquer aux dioïdes les résultats concernant les points fixes d'applications isotones définies sur des treillis complets. Ainsi, des équations du type $f(x) = x$ peuvent être résolues.

Définition 2.56 (Ensemble des points fixes d'application). Soit f une application isotone définie sur un dioïde complet $\mathcal{D}$. Les ensembles des points fixes, post-fixes et pré-fixes d'applications sont respectivement définis comme suit :

$$\begin{aligned} \mathcal{F}_f &= \{x \in \mathcal{D} \mid f(x) = x\}, \\ \mathcal{P}_f &= \{x \in \mathcal{D} \mid f(x) \succcurlyeq x\}, \\ \mathcal{Q}_f &= \{x \in \mathcal{D} \mid f(x) \preccurlyeq x\}. \end{aligned}$$

Notons que les ensembles des points post-fixes $\mathcal{P}_f$ et pré-fixes $\mathcal{Q}_f$ peuvent être interprétés dans $\mathcal{F}_f$ selon les équivalences suivantes :

$$\begin{aligned} f(x) \succcurlyeq x &\Leftrightarrow f(x) \wedge x = x, \\ f(x) \preccurlyeq x &\Leftrightarrow f(x) \oplus x = x. \end{aligned}$$

Théorème 2.57 ([Tarski 1955]). Soit f une application isotone définie sur un dioïde complet $\mathcal{D}$. L'ensemble $\mathcal{F}_f$ est un treillis complet non-vide. Ainsi, f admet un plus petit et un plus grand point fixe.

Le théorème suivant montre qu'il est possible de substituer la recherche de la borne supérieure de $\mathcal{F}_f$ à celle de $\mathcal{P}_f$. Il en est de même pour les bornes inférieures de $\mathcal{F}_f$ et de $\mathcal{Q}_f$.

Théorème 2.58 ([Baccelli 1992, Th. 4.70 et 4.72]). Soit f une application isotone définie sur un dioïde complet $\mathcal{D}$. La borne supérieure de $\mathcal{F}_f$ coïncide avec celle de $\mathcal{P}_f$, soit :

$$\bigoplus \mathcal{F}_f = \bigoplus \mathcal{P}_f \quad \text{et} \quad \bigoplus \mathcal{P}_f \in \mathcal{F}_f.$$

De même, la borne inférieure de $\mathcal{F}_f$ coïncide avec celle de $\mathcal{Q}_f$:

$$\bigwedge \mathcal{F}_f = \bigwedge \mathcal{Q}_f \quad \text{et} \quad \bigwedge \mathcal{Q}_f \in \mathcal{F}_f.$$

3.3.1 Plus grands points fixes d'applications

Dans un premier temps, considérons les plus grands points fixes d'applications, soient les bornes supérieures des ensembles $\mathcal{F}_f$ et $\mathcal{P}_f$. Deux méthodes d'obtention de ces bornes sont données : l'une est analytique, l'autre est algorithmique.

Une première solution permettant de calculer cette borne supérieure est donnée par le théorème suivant, en ajoutant la condition que f est une application semi-continue supérieurement.

Théorème 2.59 ([Baccelli 1992, Th. 4.72]). Soit f une application semi-continue supérieurement définie sur un dioïde complet $\mathcal{D}$. Le plus grand point fixe de f est noté $\hat{y}$ et est donné par :

$$\hat{y} = \bigoplus_{y \in \mathcal{P}_f} y = f_\star(\bigoplus_{x \in \mathcal{D}} x) = f_\star(\top),$$

avec $f_\star = \bigwedge_{i=0}^{+\infty} f^i$, $f^{i+1} = f \circ f^i$ et $f^0 = \mathsf{Id}_{\mathcal{D}}$.

Une autre manière d'obtenir ce résultat consiste à exploiter l'égalité donnée dans le théorème 2.58 grâce à l'algorithme suivant qui, s'il converge, aboutit au plus grand point fixe.

Théorème 2.60 (Théorème de Kleene). Soit f une application isotone définie sur un dioïde complet $\mathcal{D}$. Si le schéma itératif suivant :

soit $x_0 = \top$,
faire $x_{n+1} = f(x_n)$,
jusqu'à $x_{m+1} = x_m$ pour $m \in \mathbb{N}$,

converge en un nombre fini m d'itérations, alors $\hat{y} = x_m$ est le plus grand point fixe de f.

Preuve. Tout d'abord, puisque $x_m = f(x_m)$, alors $x_m \preccurlyeq f(x_m)$, et ainsi x_m appartient à $\mathcal{P}_f$. Ensuite, il est nécessaire de montrer que x_m est le plus grand point fixe de f, soit $x_m = \bigoplus \mathcal{P}_f$. Cette solution sera également le plus grand point fixe de $\mathcal{F}_f$ grâce au théorème 2.58. Soit $x' \in \mathcal{P}_f$, puisque $x_0 = \top$, $x_0 \succcurlyeq x'$. Puis, si $x_n \succcurlyeq x'$, alors $x_{n+1} \succcurlyeq x'$ car grâce à l'isotonie de f et sachant que $x' \in \mathcal{P}_f$: $x_{n+1} = f(x_n) \succcurlyeq f(x') \succcurlyeq x'$.

□

Remarque 2.61 De par l'isotonie de f, la suite définie dans le théorème 2.60 est décroissante.

Il est également possible d'utiliser une variante de cet algorithme afin de trouver le plus grand point fixe, plus petit qu'une valeur de départ donnée dans $\mathcal{D}$.

Corollaire 2.62 Soient h une application isotone définie sur un dioïde complet $\mathcal{D}$ et w un élément de $\mathcal{D}$. Soit $f : \mathcal{D} \mapsto \mathcal{D}$ une application définie par $f(x) = h(x) \wedge w$. Si le schéma itératif suivant :

soit $x_0 = w$,
faire $x_{n+1} = h(x_n) \wedge w$,
jusqu'à $x_{m+1} = x_m$ pour $m \in \mathbb{N}$,

converge en un nombre fini m d'itérations, alors $\hat{y} = x_m$ est le plus grand point fixe de f. Bien sûr, ce plus grand point fixe est plus petit que w : $\hat{y} \preccurlyeq w$.

3.3.2 Plus petits points fixes d'application

Il est également possible d'appliquer ces résultats de manière duale aux plus petits points fixes d'applications, soient les bornes inférieures de $\mathcal{F}_f$ et $\mathcal{Q}_f$. Dans ce mémoire, seul le résultat donné par le théorème 2.59 est repris et décrit ci-dessous.

Théorème 2.63 ([Baccelli 1992, Th. 4.70]). Soit f une application semi-continue inférieurement définie sur un dioïde complet $\mathcal{D}$. Le plus petit point fixe de f est noté $\check{y}$ et est donné par :

$$\check{y} = \bigwedge_{y \in \mathcal{Q}_f} y = f^\star(\bigwedge_{x \in \mathcal{D}} x) = f^\star(\varepsilon),$$

avec $f^\star = \bigoplus_{i=0}^{+\infty} f^i$, $f^{i+1} = f \circ f^i$ et $f^0 = \mathsf{Id}_{\mathcal{D}}$.

Définition 2.64 (Etoile de Kleene - Clôture sous-additive). L'application notée * et appelée étoile de Kleene ou clôture sous-additive, est définie comme suit sur un dioïde complet $\mathcal{D}$ et $\forall a \in \mathcal{D}$:

$$a^\star \triangleq \bigoplus_{i=0}^{+\infty} a^i \quad \text{avec} \quad a^{i+1} = a \otimes a^i \quad \text{et} \quad a^0 = e.$$

Remarque 2.65 (Extension au cas matriciel). D'après [Baccelli 1992, Lem. 4.101],

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 $A^\star$ s'écrit alors :

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

Cette application étoile de Kleene a déjà été utilisée lors de la construction du dioïde quotient $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ (cf. définition 2.37). Quelques-unes de ses nombreuses propriétés sont données dans le théorème suivant.

Théorème 2.66 Pour commencer, la relation d'ordre du dioïde complet $\mathcal{D}$ détermine la propriété d'isotonie suivante $\forall a, b \in \mathcal{D}$:

$$a \preccurlyeq b \quad \Rightarrow \quad a^\star \preccurlyeq b^\star.$$

Les équations suivantes sont également données :

$$a \preccurlyeq a^\star, \tag{2.31}$$

$$(a^\star)^\star = a^\star, \tag{2.32}$$

$$a(ba)^\star = (ab)^\star a, \tag{2.33}$$

$$(a\oplus b)^\star = (a^\star b)^\star a^\star = a^\star(ba^\star)^\star = b^\star(ab^\star)^\star = (b^\star a)^\star b^\star, \tag{2.34}$$

$$a^\star a^\star = a^\star, \tag{2.35}$$

$$(ab^\star)^\star = e \oplus a(a\oplus b)^\star, \tag{2.36}$$

$$(a\oplus b)^\star = (a^\star \oplus b)^\star = (a\oplus b^\star)^\star = (a^\star b^\star)^\star = (a^\star \oplus b^\star)^\star. \tag{2.37}$$

Lorsque $\mathcal{D}$ est commutatif :

$$(a\oplus b)^\star = a^\star b^\star. \tag{2.38}$$

En lien avec les quotients à gauche $\backslash$ et à droite $/$ (cf. équations (2.7) et (2.8)) :

$$a = a^\star \Leftrightarrow a = a\backslash a \quad \text{et} \quad a = a^\star \Leftrightarrow a = a/a, \tag{2.39}$$

$$a^*\backslash x = a^*\backslash(a^*\backslash x) \quad \text{et} \quad x/a^* = (x/a^*)/a^*, \tag{2.40}$$

$$a^*x = a^*\backslash(a^*x) \quad \text{et} \quad xa^* = (xa^*)/a^*, \tag{2.41}$$

$$a^*\backslash x = a^*(a^*\backslash x) \quad \text{et} \quad x/a^* = (x/a^*)a^*. \tag{2.42}$$

Pour finir :

$$a^\star = \bigoplus\{x \mid x^\star \preccurlyeq a^\star\}, \tag{2.43}$$

$$a^\star = \bigwedge\{x \mid x = x^\star,\ x \succcurlyeq a\}. \tag{2.44}$$

Preuve. Les preuves des équations (2.32) à (2.38) sont données dans les thèses [Gaubert 1992, §4.1, Prop. 4.1.6] et [Cottenceau 1999a, §1.3.2, Th. 1.88], tandis que l'équivalence (2.39) est donnée dans [Cottenceau 1999a, §1.3.4, Th. 1.106] et les équations (2.40) à (2.42) dans [Cottenceau 1999a, §1.3.3, Th. 1.101]. Concernant l'équation (2.43), il suffit de vérifier que $x = a^\star$ est solution de $x^\star \preccurlyeq a^\star$, soit : $x^\star \preccurlyeq a^\star \Leftrightarrow (a^\star)^\star \preccurlyeq a^\star \Leftrightarrow a^\star \preccurlyeq a^\star$ (cf. équation (2.32)). Il s'agit alors de la plus grande solution. Finalement, pour l'équation (2.44), vérifions que $x = a^\star$ est solution, soit :

$$\begin{cases} x = x^\star \\ x \succcurlyeq a \end{cases} \Leftrightarrow \begin{cases} a^\star = (a^\star)^\star = a^\star & \text{cf. équation (2.32)}, \\ a^\star \succcurlyeq a & \text{cf. équation (2.31)}. \end{cases}$$

Il s'agit de la solution la plus petite car tout autre x est plus grand que $a^\star$, soit : $x \succcurlyeq a \Leftrightarrow x^\star \succcurlyeq a^\star$ (isotonie de *) $\Leftrightarrow (a^\star)^\star \preccurlyeq a^\star \Leftrightarrow a^\star \preccurlyeq a^\star$ (cf. équation (2.32)) $\Leftrightarrow x \preccurlyeq a^\star$.

□

Remarque 2.67 Les propriétés (2.43) et (2.44) peuvent être considérées comme des problèmes de résiduation de l'application $x \mapsto x^\star$. La relation indique que $x^\star \preccurlyeq a$ admet une solution optimale lorsque $a = a^\star$. Ces aspects sont étudiés dans [Cottenceau 1999a] et [Cottenceau 2001].

3.3.3 Résolution de l'équation $x = ax \oplus b$

Le cas particulier de l'application implicite $f(x) = ax \oplus b$ est donné ci-dessous.

Théorème 2.68 ([Baccelli 1992, Th. 4.75]). Soit $\mathcal{D}$ un dioïde complet et a, b deux éléments de $\mathcal{D}$. L'équation implicite suivante :

$$x = ax \oplus b, \tag{2.45}$$

admet $a^\star b$ comme plus petite solution.

Preuve. Les applications $L_a : x \mapsto a \otimes x$ (cf. équation (2.5)) et $x \mapsto x \oplus b$ étant semi-continues inférieurement, $ax \oplus b$ l'est également. Soit $f(x) = ax \oplus b$, le calcul de $f^\star(x)$ est donné comme suit :

$$\begin{aligned} f^2(x) &= a(ax \oplus b) \oplus b = a^2x \oplus ab \oplus b, \\ f^3(x) &= a^2(ax \oplus b) \oplus ab \oplus b = a^3x \oplus a^2b \oplus ab \oplus b, \\ &\vdots \\ f^i(x) &= a^ix \oplus a^{i-1}b \oplus ab \oplus b. \end{aligned}$$

Soit :

$$\begin{aligned} f^\star(x) &= x \oplus ax \oplus a^2x \oplus \ldots \oplus b \oplus ab \oplus a^2b \oplus \ldots, \\ &= x(e \oplus a \oplus a^2 \oplus \ldots) \oplus b(e \oplus a \oplus a^2 \oplus \ldots), \\ &= a^\star(x \oplus b). \end{aligned}$$

Ainsi, d'après le théorème 2.63, $f^\star(\varepsilon) = a^\star b$ est le plus petit point fixe de l'équation (2.45).

□

4 Conclusion

Ce chapitre a permis de présenter la théorie des dioïdes, nom donné aux semi-anneaux idempotents. Cette structure algébrique est au cœur de la modélisation des systèmes (min,+)-linéaires présentée dans le chapitre suivant.

La section 1 a rappelé un ensemble de définitions et théorèmes relatifs à l'utilisation des dioïdes, et de nombreux exemples de dioïdes sont proposés dans la section 2. Une caractéristique importante est la relation d'ordre associée à la loi additive grâce à son idempotence.

Puisqu'un dioïde peut être considéré comme une structure ordonnée, des résultats issus de la théorie des treillis s'appliquent naturellement aux dioïdes dans la section 3. La théorie de la résiduation fournit tout d'abord une réponse alternative au problème d'inversion d'applications en solutionnant des inéquations du type $f(x) \preccurlyeq b$ (cf. définition 2.47). Les équivalences suivantes sont notamment obtenues :

$$\begin{aligned} ax \preccurlyeq b &\Leftrightarrow x \preccurlyeq a \backslash b, \\ xa \preccurlyeq b &\Leftrightarrow x \preccurlyeq b / a. \end{aligned}$$

Ensuite, les recherches de points fixes d'applications isotones sont proposées à travers des équations du type $f(x) = x$. L'application étoile de Kleene * (également appelée clôture sous-additive) est ainsi introduite et permet de proposer $a^\star b$ comme plus petite solution de l'équation $x = ax \oplus b$ (cf. théorème 2.68).

Chapitre 3

Modélisation de systèmes

Sommaire

1 Modélisation des systèmes (min,+)-linéaires 43
1.1 Graphe d'Événements Temporisés 44
1.2 Modélisation analytique : équations aux compteurs 45
1.3 Représentation d'état et relation de transfert 47
1.4 Caractéristiques des systèmes (min,+)-linéaires 49
1.5 Caractéristiques des fonctions de $\mathcal{F}_{cp}$ 52
2 Du point de vue du Network Calculus 55
2.1 Contexte algébrique 56
2.2 Modélisation par des flux et des contraintes 58
2.3 Courbe d'arrivée 62
2.4 Courbe de service 64
2.5 Contrainte sur la sortie 66
3 Mesure des écarts : théorie du second ordre 67
3.1 Corrélation et autocorrélation 67
3.2 Application aux systèmes (min,+)-linéaires 70
3.3 Application au Network Calculus 72
4 Conclusion 76

1 Modélisation des systèmes (min,+)-linéaires

La classe des SED mettant en jeu des phénomènes de synchronisation et de délai peut être modélisée de façon linéaire grâce à l'algèbre des dioïdes. Avec les outils algébriques introduits dans le chapitre 2, il est maintenant possible de développer cette modélisation.

Dans un premier temps, une représentation graphique de ces systèmes est proposée grâce à l'utilisation des Graphes d'Événements Temporisés. Dans un

second temps, des fonctions compteurs d'événements sont définies [1] sur le dioïde complet $\overline{\mathbb{R}}_{min}$ (cf. exemple 2.17). Les systèmes étudiés sont alors appelés des systèmes (min,+)-linéaires [2].

1.1 Graphe d'Événements Temporisés

Définition 3.1 (Graphe d'Événements Temporisés). Les Graphes d'Événements Temporisés constituent une sous-classe des réseaux de Petri pour laquelle toute place ne possède qu'une transition en amont et une en aval. Une transition d'un GET correspond alors à un événement (soit par exemple le déclenchement d'une action, le début ou la fin d'une tâche, la réservation ou la restitution d'une ressource), et le tir de celle-ci représente l'occurrence de cet événement.

Exemple 3.2 Le GET de la figure 3.1 représente un modèle simplifié d'un système d'assemblage manufacturier constitué de trois machines M_1, M_2 et M_3, et dans lequel les pièces issues des machines M_1 et M_2 sont assemblées sur une machine M_3.

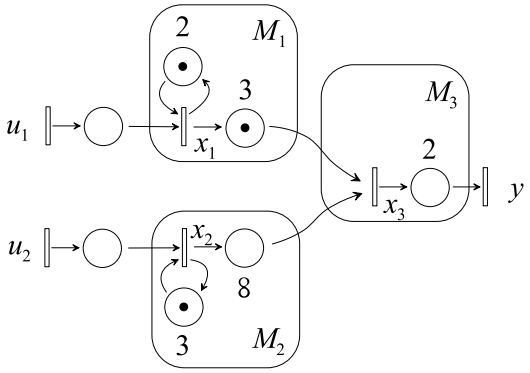


Figure 3.1 Graphe d'Événements Temporisés.

Les places temporisées (cercles blancs) induisent un temps minimum de séjour aux jetons (points noirs) avant leurs contributions au franchissement des transitions

1. De nombreux travaux existent déjà sur des fonctions dites dateurs d'événements, nous laisserons donc cet aspect de côté dans ce mémoire.

2. Dans le cadre des fonctions dateurs, la modélisation se fait par exemple à travers le dioïde $\overline{\mathbb{R}}_{max}$ et les systèmes sont ainsi appelés systèmes (max,+)-linéaires.

situées en aval. Ils ont pour rôle de traduire une tâche qui possède une durée : la transition x_3 exprime la synchronisation des événements x_1 et x_2 retardés respectivement de 3 et 8 unités de temps. Les jetons initialement présents dans le système signifient que les ressources sont disponibles. Ils provoquent un décalage dans les événements : le tir numéro k de la transition x_3 sera engendré par le tir numéro $k-1$ de la transition x_1 et l'activation numéro k de x_2.

Enfin, le comportement d'un GET est considéré ici soumis à la règle de franchissement *au plus tôt* des transitions, c'est-à-dire que toute transition franchissable est immédiatement franchie.

1.2 Modélisation analytique : équations aux compteurs

L'étape suivante de la modélisation est la mise en équations du modèle graphique, donnée ici dans l'algèbre (min,+). Cette modélisation peut être faite sans obligatoirement réaliser le GET correspondant.

Tout d'abord, les événements d'entrée (événements pour lesquels il est possible de contrôler les occurrences), les événements internes et les événements de sortie d'un GET sont représentés par des vecteurs appelés u, x et y avec pour tailles respectives p, n et q.

Ensuite, les synchronisations d'un système (min,+)-linéaire sont traduites comme suit : le nombre d'occurrences d'un événement dépend du nombre minimum d'occurrences des événements précédents. Elles sont donc représentées par l'opérateur min et interprétés dans le dioïde $\overline{\mathbb{R}}_{min}$ par l'opérateur $\oplus$. Quant aux délais passés avant les occurrences des événements, ils s'additionnent et sont donc représentés par l'opérateur $\otimes$.

Sur $\overline{\mathbb{R}}_{min}$, la modélisation linéaire est réalisée à travers des fonctions appelées *fonctions compteurs*. Plus précisément, pour un événement noté x, la fonction[3] $x(t) : \mathbb{R} \mapsto \overline{\mathbb{R}}_{min}$ fait le compte des événements x qui se sont déroulés jusqu'au temps t. Il s'agit alors de fonctions croissantes (le nombre d'événements est cumulé) et constantes par morceaux (un événement est considéré comme un entier)

3. Une fonction compteur $x(t)$ est également appelée la *trajectoire* de l'événement x et correspond donc à l'historique de ses occurrences.

appartenant à l'ensemble $\mathcal{F}_c$ défini ci-dessous.

Définition 3.3 (Ensemble $\mathcal{F}_c$). Une fonction $f : \mathbb{R} \mapsto \overline{\mathbb{R}}_{min}$ appartient à l'ensemble noté $\mathcal{F}_c$ si elle est croissante, *i.e.* $\forall t_1 > t_2, f(t_1) \geq f(t_2)$, et constante par morceaux, *i.e.* qui correspond à la réunion d'un nombre fini ou infini d'intervalles dans lesquels la fonction coïncide avec une fonction constante (cf. figure 3.2).

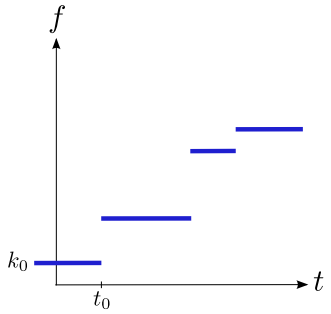


Figure 3.2 Une fonction constante par morceaux $f \in \mathcal{F}_c$.

Ainsi, en reprenant le GET de la figure 3.1, les équations présentant les synchronisations et les délais sont les suivantes, avec u, x et $y \in \mathcal{F}_c$ les fonctions compteurs du système :

$$\begin{cases} x_1(t) = \min(u_1(t), 1 + x_1(t-2)), \\ x_2(t) = \min(u_2(t), 1 + x_2(t-3)), \\ x_3(t) = \min(1 + x_1(t-3), x_2(t-8)), \\ y(t) = x_3(t-2). \end{cases}$$

Grâce au dioïde $\overline{\mathbb{R}}_{min}$, ce système d'équations devient :

$$\begin{cases} x_1(t) = u_1(t) \oplus 1 \otimes x_1(t-2), \\ x_2(t) = u_2(t) \oplus 1 \otimes x_2(t-3), \\ x_3(t) = 1 \otimes x_1(t-3) \oplus x_2(t-8), \\ y(t) = x_3(t-2). \end{cases} \tag{3.1}$$

1.3 Représentation d'état et relation de transfert

Il est possible d'écrire ces interactions entre les fonctions d'entrée u, d'état x et de sortie y selon une forme matricielle récurrente [4] appelée *représentation d'état* ou *équation d'état*. Pour ce faire, les trois matrices $A \in {\overline{\mathbb{R}}_{min}}^{n\times n}$, $B \in {\overline{\mathbb{R}}_{min}}^{n\times p}$ et $C \in {\overline{\mathbb{R}}_{min}}^{q\times n}$ sont introduites avec, rappelons-le, n la taille du vecteur d'état x, p la taille du vecteur d'entrée u et q la taille du vecteur de sortie y. L'équation d'état provenant du système d'équations (3.1) est alors la suivante :

$$\begin{cases} x(t) = A \otimes x(t-1) \oplus B \otimes u(t), \\ y(t) = C \otimes x(t). \end{cases} \tag{3.2}$$

Remarque 3.4 Le système (3.1) ne se ramène pas immédiatement à la forme standard (3.2). Cette transformation est néanmoins toujours possible au prix d'un agrandissement du vecteur d'état et donc également des matrices A, B et C.

La dynamique des systèmes (min,+)-linéaires exposée sous forme d'équations récurrentes dans la représentation d'état précédente, peut également être décrite selon une vision entrée/sortie. En effet, dans le cas d'un système système mono-entrée mono-sortie (SISO : Single-Input Single-Output), soit un système où $p = 1$ et $q = 1$, le développement des récurrences de (3.2) montre que la fonction compteur $y(t)$ associée à la sortie du système peut s'exprimer comme suit :

$$\begin{aligned} y(t) &= Cx(t), \\ &= CAx(t-1) \oplus CBu(t), \\ &= CA^2Bu(t-2) \oplus CABu(t-1) \oplus CBu(t), \\ &= \dots, \\ &= \bigoplus_{\tau\geq 0} CA^\tau Bu(t-\tau). \end{aligned} \tag{3.3}$$

La définition suivante introduit un opérateur qui va simplifier cette expression. Le dioïde d'étude sera alors le dioïde de fonctions noté $(\mathcal{F}_c, \oplus, *)$.

Définition 3.5 (Produit d'inf-convolution). Soient $f(t)$ et $g(t)$ deux fonctions compteurs appartenant à $\mathcal{F}_c$. Le produit de convolution appelé (min,+)-convolution ou encore inf-convolution de f par g est la fonction compteur $f * g \in \mathcal{F}_c$ définie

4. Forme proche de celle utilisée lors de la description d'un système en automatique classique.

comme suit :

$$f * g : \quad \forall t,\ (f*g)(t) \triangleq \bigoplus_{\tau \geq 0} \{f(\tau) \otimes g(t-\tau)\} = \min_{\tau \geq 0} \{f(\tau) + g(t-\tau)\}. \tag{3.4}$$

Remarque 3.6 (Extension au cas matriciel). Par extension, l'inf-convolution de deux matrices $D^{m\times n}$ et $F^{n\times p}$ est définie comme suit :

$$(D * F)_{ij} \triangleq \bigoplus_{k=1}^{n} \{D_{ik} * F_{kj}\}. \tag{3.5}$$

Remarque 3.7 L'inf-convolution est un opérateur commutatif qui distribue sur la somme $\oplus$. L'élément neutre qui lui est associé est la fonction e définie par :

$$e(t) = \begin{cases} 0 & \text{pour } t < 0, \\ +\infty & \text{pour } t \geq 0. \end{cases} \tag{3.6}$$

Remarque 3.8 Tout comme dans l'algèbre classique et lorsqu'il n'y aura aucune ambiguïté avec le produit $\otimes$ du dioïde (min,+) (soit la somme classique), le signe $*$ pourra être omis : $a * b = ab$.

Définition 3.9 (Dioïde de fonctions $(\mathcal{F}_c, \oplus, *)$). Le dioïde des fonctions compteurs de $\mathcal{F}_c$ munies du minimum point à point comme somme $\oplus$ et de l'inf-convolution comme produit $*$ est noté [5] $(\mathcal{F}_c, \oplus, *)$. Les éléments neutres de ce dioïde sont la fonction e telle que donnée par l'équation (3.6) et la fonction ε telle que :

$$\forall t,\ \varepsilon(t) = +\infty. \tag{3.7}$$

Grâce à cet opérateur d'inf-convolution, le comportement entrée/sortie d'un système (min,+)-linéaire donné par l'équation (3.3) peut être exprimé comme suit :

$$\forall t,\ y(t) = (h * u)(t) \quad \text{soit} \quad y = h * u, \tag{3.8}$$

avec y, h et $u \in (\mathcal{F}_c, \oplus, *)$. Ainsi, la sortie du système n'est plus dépendante de l'état du système mais uniquement de son entrée. La fonction h est appelée la *fonction de*

5. Cette notation $(\mathcal{F}_c, \oplus, *)$ donne le nom de l'ensemble des fonctions manipulées et est donc préférée à celle $(\mathcal{D}^{\mathbb{R}}, \oplus, *)$ donnée dans la définition 2.22, afin de différencier plusieurs dioïdes de fonctions munies du min et de l'inf-convolution.

transfert du système et est donnée par :

$$h(t) = \begin{cases} \varepsilon & \text{pour } t < 0, \\ CA^tB & \text{pour } t \geq 0. \end{cases} \tag{3.9}$$

Remarque 3.10 Cette représentation possède des analogies avec l'automatique classique. La fonction $h(t)$ équivaut à la réponse impulsionnelle si l'entrée $u = e$ (cf. équation (3.6)) est considérée comme l'impulsion.

Concernant les systèmes multi-entrées multi-sorties (MIMO : Multiple-Input Multiple-Output), la relation liant la sortie Y à l'entrée U (cf. équation (3.5)) est décrite par :

$$Y(t) = (H * U)(t).$$

La matrice H de dimension $(q \times p)$ est appelée la *matrice de transfert* du système.

1.4 Caractéristiques des systèmes (min,+)-linéaires

Tout d'abord, un système est appelé (min,+)-linéaire s'il répond à la définition 3.12 donnée ci-dessous.

Définition 3.11 (Produit scalaire). Le produit d'une fonction $u \in \mathcal{F}_c$ par un scalaire $\lambda \in \mathbb{R}$ est défini par :

$$\lambda \cdot u : \quad \forall t,\ (\lambda \cdot u)(t) \triangleq \lambda \otimes u(t) = \lambda + u(t).$$

Définition 3.12 (Système (min,+)-linéaire). Un système noté $\mathcal{S}$ est une application qui à une fonction d'entrée $u \in \mathcal{F}_c$, associe une fonction de sortie $y = \mathcal{S}(u) \in \mathcal{F}_c$. Autrement dit, $\mathcal{S} : \mathcal{F}_c \mapsto \mathcal{F}_c, u \mapsto y$. Un tel système $\mathcal{S}$ est appelé (min,+)-linéaire lorsque pour toutes fonctions d'entrée notées u_1 et u_2 :

$$\mathcal{S}(u_1 \oplus u_2) = \mathcal{S}(u_1) \oplus \mathcal{S}(u_2),$$

et pour tout scalaire $a \in \mathbb{R}$:

$$\mathcal{S}(a \cdot u_1) = a \cdot \mathcal{S}(u_1),$$

avec $\cdot$ le produit scalaire donné dans la définition 3.11.

Ensuite, un résultat fondamental dans la théorie des systèmes (min,+)-linéaires est donné par le théorème suivant.

Théorème 3.13 ([Baccelli 1992, Th. 5.39]). La fonction de transfert d'un système (min,+)-linéaire SISO (tel que représenté par l'équation d'état (3.2)) est une fonction $h : \mathbb{R} \to \overline{\mathbb{R}}_{min}$ croissante, constante par morceaux,

$$\text{ultimement pseudo-périodique :} \quad \begin{array}{c} \exists T_p, \exists K \in \overline{\mathbb{R}}_{min}, \exists T \in \mathbb{R}, \\ \text{tels que } \forall t \geq T_p,\ h(t+T) = K \otimes h(t), \end{array} \tag{3.10}$$

$$\text{et causale :} \quad \begin{cases} h(t) = h(0) & \text{pour } t < 0, \\ h(t) \geq 0 & \text{pour } t \geq 0. \end{cases} \tag{3.11}$$

En effet, après un comportement dit *transitoire* (soit jusqu'à T_p), le fonctionnement d'un système (min,+)-linéaire (pour lequel $u = e$) entre dans un *régime périodique* caractérisant le temps de cycle du système.

Idée de preuve. La fonction de transfert h est obtenue en réalisant un nombre fini d'opérations $\{\oplus, *, {}^\star\}$ sur des fonctions de $\mathcal{F}_{cp}$. Une preuve technique consiste à montrer que ces opérations sont internes à l'ensemble $\mathcal{F}_{cp}$ (cf. par exemple [Gaubert 1992] ou [Bouillard 2008b, Prop. 4 et 5]). □

D'après ce théorème, les systèmes (min,+)-linéaires disposent donc de modèles linéaires sur les dioïdes. Notamment, la fonction de transfert h d'un système SISO appartient au dioïde de fonctions noté $(\mathcal{F}_{cp}, \oplus, *)$, sous-dioïde de $(\mathcal{F}_c, \oplus, *)$. Dans le cas MIMO, H appartient au dioïde $(\mathcal{F}_{cp}, \oplus, *)^{q \times p}$.

Définition 3.14 (Ensemble $\mathcal{F}_{cp}$). Une fonction $f : \mathbb{R} \mapsto \overline{\mathbb{R}}_{min}$ appartient à l'ensemble noté $\mathcal{F}_{cp}$ (cf. figure 3.3) si $f \in \mathcal{F}_c$ et si f est ultimement pseudo-périodique (cf. équation (3.10)). Ainsi, $\mathcal{F}_{cp} \subset \mathcal{F}_c$.

Définition 3.15 (Pente asymptotique σ). Soit f une fonction de $\mathcal{F}_{cp}$, sa pente asymptotique est définie par le ratio $\sigma(f) = K/T$.

Remarque 3.16 Si f est une fonction compteur, sa pente asymptotique correspond alors à un nombre d'événements par unité de temps, c'est-à-dire à un taux de

production (plus la pente asymptotique est grande, plus le système est rapide).

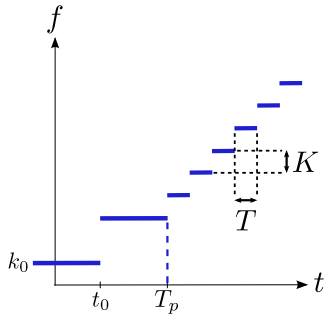


Figure 3.3 Une fonction ultimement pseudo-périodique $f \in \mathcal{F}_{cp}$.

Définition 3.17 (Dioïde de fonctions $(\mathcal{F}_{cp}, \oplus, *)$). L'ensemble des fonctions ultimement pseudo-périodiques appartenant à $\mathcal{F}_{cp}$, munies du minimum point à point comme somme $\oplus$ et de l'inf-convolution comme produit $*$, est un dioïde noté $(\mathcal{F}_{cp}, \oplus, *)$. Ainsi, $(\mathcal{F}_{cp}, \oplus, *) \subset (\mathcal{F}_c, \oplus, *)$.

En s'appuyant sur la représentation entrée/sortie de l'équation (3.8), tout système (min,+)-linéaire[6] peut être décrit par la combinaison de systèmes SISO notés h_i. Les combinaisons sont :

- *la mise en parallèle :* le comportement parallèle de deux systèmes dont les transferts sont h_1 et h_2 (cf. figure 3.4a) est équivalent au système :

  $$h_1 \oplus h_2,$$

  pour lequel l'opérateur $\oplus$ est le min des fonctions de transfert,
- *la mise en série :* le comportement en série de deux systèmes dont les transferts sont h_1 et h_2 (cf. figure 3.4b) est équivalent au système :

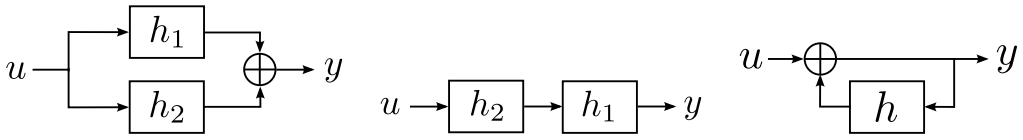
  $$h_1 * h_2,$$

  pour lequel l'opérateur $*$ correspond à l'inf-convolution,
- *le rebouclage :* pour un système h donné, le système dont la dynamique est décrite par l'équation implicite $y = hy \oplus u$ (la dynamique du système h est dans une boucle, cf. figure 3.4c), possède le comportement entrée-sortie :

  $$y = h^\star * u,$$

6. Systèmes MIMO compris.

dans lequel $h^\star$ représente l'étoile de Kleene de h (cf. théorème 2.68).



(a) Mise en parallèle (b) Mise en série (c) Rebouclage

Figure 3.4 Compositions de systèmes (min,+)-linéaires.

1.5 Caractéristiques des fonctions de $\mathcal{F}_{cp}$

Afin de conclure cette modélisation des systèmes (min,+)-linéaires, quelques caractéristiques des fonctions de l'ensemble $\mathcal{F}_{cp}$ (cf. définition 3.14) sont données.

Tout d'abord, une fonction de $\mathcal{F}_{cp}$ constante par morceaux et ultimement pseudo-périodique peut se décomposer en une combinaison $\oplus$ infinie de fonctions dites *élémentaires* définies ci-dessous.

Définition 3.18 (Fonction élémentaire Δ_T^K). Une fonction élémentaire notée Δ_T^K et illustrée figure 3.5a, est la fonction compteur appartenant à $\mathcal{F}_c$ (cf. définition 3.3) définie par :

$$\Delta_T^K(t) \triangleq \begin{cases} K & \text{si } t \leq T, \\ +\infty & \text{sinon.} \end{cases}$$

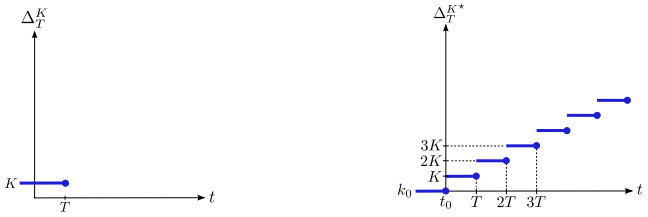
Remarque 3.19 (Étoile d'une fonction élémentaire $\Delta_T^{K\star}$). Soit Δ_T^K une fonction élémentaire. Son étoile de Kleene $\Delta_T^{K\star}$ illustrée figure 3.5b, est une fonction de $\mathcal{F}_{cp}$ telle que :

$$\Delta_T^{K\star} = \bigoplus_{i=0}^{+\infty} \Delta_{t_i}^{k_i},$$

où $t_0 = 0$, $k_0 = 0$, et $t_{(i+1)} = t_i + T$, $k_{(i+1)} = k_i + K$. Plus simplement :

$$\Delta_T^{K\star} = \bigoplus_{i=0}^{+\infty} \Delta_{i\times T}^{i\times K} = \Delta_0^0 \oplus \Delta_T^K \oplus \Delta_{2T}^{2K} \oplus \dots \ .$$

La pente asymptotique de cette étoile est définie par $\sigma(\Delta_T^{K\star}) = K/T$.



(a) Une fonction élémentaire Δ_T^K

(b) Etoile de Kleene d'une fonction élémentaire : $\Delta_T^{K\star} \in \mathcal{F}_{cp}$

Figure 3.5 Caractéristique d'une fonction élémentaire Δ_T^K.

Propriété 3.20 Quelques propriétés peuvent être associées à ces fonctions élémentaires :

$$\Delta_\tau^0 * \Delta_{-\tau}^0 = \Delta_0^0 = e, \tag{3.12}$$

$$\Delta_0^\nu * \Delta_0^{-\nu} = \Delta_0^0 = e. \tag{3.13}$$

De plus, en relation avec une fonction $u \in \mathcal{F}_c$:

$$\begin{aligned} \forall t,\ (\Delta_\tau^0 * u)(t) &= u(t-\tau), \\ \forall t,\ (\Delta_0^\nu * u)(t) &= \nu + u(t). \end{aligned}$$

Ainsi, soit f une fonction de $\mathcal{F}_{cp}$:

$$f = \bigoplus_{i=0}^{+\infty} \Delta_{t_i}^{k_i}. \tag{3.14}$$

En outre, la pseudo-périodicité des fonctions de $\mathcal{F}_{cp}$ donnée par l'équation (3.10) est rappelée ci-dessous avec l'introduction de t_0, temps de la première fonction élémentaire $\Delta_{t_0}^{k_0}$ de f :

$$\exists T_p \geq t_0, \exists K \in \mathbb{R}_{min}, \exists T \in \mathbb{R} \quad \text{tels que} \quad \forall t \geq T_p,\ f(t+T) = K \otimes f(t).$$

Selon [Gaubert 1992], chaque fonction de $\mathcal{F}_{cp}$ a une forme canonique où T_p et T sont minimums. Il est donc possible de définir une représentation canonique des fonctions de $\mathcal{F}_{cp}$.

Propriété 3.21 (Représentation canonique d'une fonction de $\mathcal{F}_{cp}$). Une fonction

$f \in \mathcal{F}_{cp}$ est dite sous forme canonique lorsqu'elle est définie comme suit :

$$f = \bigoplus_{i\geq 0}^{n} \Delta_{t_i}^{k_i} \oplus \left(\bigoplus_{j\geq 0}^{m} \Delta_{T_j}^{K_j}\right) * \Delta_T^{K\star},$$

avec n le nombre de fonctions élémentaires de f lors de son régime transitoire (avant T_p), m le nombre de fonctions élémentaires du motif périodique, et $\sigma(f) = K/T$ sa pente asymptotique représentée par la fonction $\Delta_T^{K\star}$.

Exemple 3.22 Dans la figure 3.6, la représentation canonique de la fonction $f \in \mathcal{F}_{cp}$ est donnée comme suit avec $n = 3$, $m = 2$ et $\sigma(f) = 3/3 = 1$:

$$f = \Delta_2^1 \oplus \Delta_3^2 \oplus \Delta_5^3 \oplus (\Delta_7^5 \oplus \Delta_8^7) * (\Delta_3^3)^\star.$$

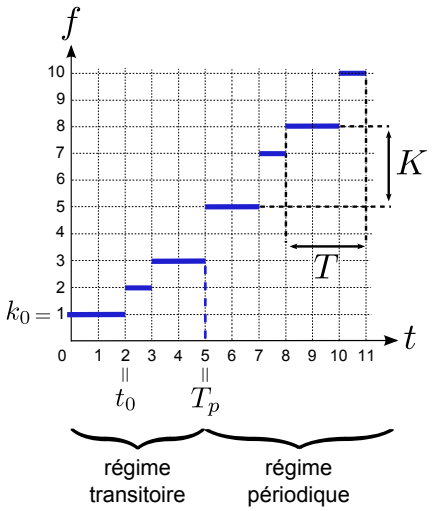


Figure 3.6 Représentation canonique d'une fonction ultimement pseudo-périodique $f \in \mathcal{F}_{cp}$.

Pour finir, il sera par la suite utile d'obtenir la projection causale d'une fonction de $(\mathcal{F}_{cp}, \oplus, *)$. Cette projection est donnée ci-dessous en tant que résiduation de l'injection canonique d'un sous-dioïde dans un autre (cf. théorème 2.54).

Définition 3.23 (Projection causale d'une fonction de $(\mathcal{F}_{cp}, \oplus, *)$). Soit $(\mathcal{F}_{cp}, \oplus, *)^{\mathsf{caus}}$ le sous-dioïde des fonctions causales de $(\mathcal{F}_{cp}, \oplus, *)$ (cf. équation (3.11)). D'après le théorème 2.54, l'injection canonique $\mathsf{I}_{\mathsf{caus}}$ définie de $(\mathcal{F}_{cp}, \oplus, *)^{\mathsf{caus}}$

dans $(\mathcal{F}_{cp}, \oplus, *)$ est résiduable et sa résiduée est notée $\mathsf{Pr}_{\mathsf{caus}} : (\mathcal{F}_{cp}, \oplus, *) \mapsto (\mathcal{F}_{cp}, \oplus, *)^{\mathsf{caus}}$. Formellement, la fonction $\mathsf{Pr}_{\mathsf{caus}}(f)$ est la plus grande fonction causale plus petite ou égale à f, *i.e.* :

$$\mathsf{Pr}_{\mathsf{caus}}(f)(t) = \begin{cases} f(t) & \text{pour } t \geq 0 \text{ et } f(t) \geq 0, \\ f(0) & \text{pour } t < 0 \text{ et } f(0) \geq 0, \\ 0 & \text{sinon.} \end{cases}$$

Remarque 3.24 Si une fonction x est causale, alors elle est invariante par $\mathsf{Pr}_{\mathsf{caus}}$, soit $\mathsf{Pr}_{\mathsf{caus}}(x) = x$.

2 Du point de vue du Network Calculus

Le Network Calculus est le nom consacré à une extension de la théorie des systèmes (min,+)-linéaires, visant à mesurer la qualité de service d'un réseau informatique (se référer à [Cruz 1991a, Cruz 1991b] pour les premiers travaux, puis à [Chang 2000] et [Le Boudec 2001] pour une généralisation). Les modèles du Network Calculus reposent d'une part sur une description des flux de données sous forme de fonctions compteurs, et d'autre part sur une description du comportement dynamique des équipements d'un réseau (routeur, switch, serveur, etc.).

De ce point de vue, le principal objectif du Network Calculus est d'obtenir des indicateurs de performance quant au comportement du réseau, indicateurs dont nous parlerons plus précisément dans la section 3 suivante (délai de bout-en-bout, taille des files d'attente, taux de transfert, etc.). A ces fins, des contraintes sont appliquées sur les flux d'entrée et de sortie, ainsi que sur les services fournis par les éléments du réseau. En effet, la qualité de service ne peut se faire qu'à la condition que les flux d'entrée aient des effets de « rafales de données » limités, et que les services fournis par les équipements aient un comportement « minimal » connu.

Les contraintes appliquées aux flux d'entrée et de sortie sont décrites au moyen de courbes d'arrivée, tandis que les comportements des éléments du réseau le sont grâce à des courbes de service. Une des principales difficultés de cette modélisation est de déterminer pour chaque élément une courbe représentant au mieux ses caractéristiques.

2.1 Contexte algébrique

La théorie du Network Calculus repose sur l'utilisation de l'algèbre (min,+) afin de décrire le fonctionnement d'un réseau. Les différentes courbes manipulées sont à valeurs réelles. En effet, malgré l'existence d'un certain niveau de discrétisation des données d'un réseau informatique (bit, mot, paquet), il est souvent plus utile lors des calculs de décrire les flux et les contraintes dans le domaine continu.

Remarque 3.25 Il est important de garder à l'esprit qu'en travaillant avec l'algèbre (min,+), l'ordre du dioïde $\preccurlyeq$ est l'inverse de l'ordre naturel $\leq$.

Les fonctions manipulées par cette théorie sont des fonctions croissantes et positives regroupées dans l'ensemble $\mathcal{F}$ défini ci-dessous [7].

Définition 3.26 (Ensemble $\mathcal{F}$). Une fonction $f \in \mathcal{F}$ est une fonction croissante définie de $\mathbb{R}$ dans $\overline{\mathbb{R}}_{min}$ et pour laquelle $f(t) = 0$ lorsque $t < 0$.

Remarque 3.27 D'après cette définition, les fonctions de $\mathcal{F}$ sont des fonctions dites causales (cf. équation (3.11)).

Définition 3.28 (Dioïde de fonctions $(\mathcal{F}, \oplus, *)$). Le dioïde des fonctions de $\mathcal{F}$ munies du minimum point à point comme somme $\oplus$ et de l'inf-convolution comme produit $*$ (cf. définition 3.5) est noté $(\mathcal{F}, \oplus, *)$. Ce dioïde est complet et commutatif, et a pour éléments neutres la fonction e (cf. équation (3.6)) et la fonction ε (cf. équation (3.7)).

Les opérations $\oplus$ et $*$ sont donc des opérations de base du Network Calculus auxquelles la loi $\wedge$ (borne inférieure du dioïde) s'ajoute en tant que maximum point à point.

Ensuite, l'opération utilisée jusqu'à présent sous le nom d'étoile de Kleene (cf. définition 2.64 et théorème 2.66), est appelée *clôture sous-additive* dans la théorie du Network Calculus. Elle est alors définie sur le dioïde complet $(\mathcal{F}, \oplus, *)$ et est

7. Cette notation provient des livres de référence du Network Calculus à savoir [Chang 2000] et [Le Boudec 2001].

rappelée ci-dessous pour tout f appartenant à ce dioïde :

$$f^\star = \bigoplus_{i=0}^{+\infty} f^i \quad \text{avec} \quad f^{i+1} = f * f^i \quad \text{et} \quad f^0 = e.$$

D'après le théorème 2.68, $f^\star$ est la plus petite solution de l'équation $x = f * x \oplus e$.

Pour finir, la résiduée de l'inf-convolution est également un opérateur couramment utilisé dans le Network Calculus. La proposition suivante présente cette résiduation.

Proposition 3.29 (Résiduation de l'inf-convolution $*$). L'opérateur d'inf-convolution $*$ du dioïde $(\mathcal{F}, \oplus, *)$ est une application résiduable. Sa résiduée appelée *déconvolution* dans le contexte du Network Calculus, est notée $\oslash$ et est définie comme suit :

$$f \oslash g : \quad \forall t,\ (f \oslash g)(t) \triangleq \bigwedge_{\tau \geq 0} \{f(\tau) \oslash g(\tau - t)\} = \sup_{\tau \geq 0} \{f(\tau) - g(\tau - t)\}. \tag{3.15}$$

Ainsi, la fonction $f \oslash g$ est la plus grande solution de l'inégalité $x * g \preccurlyeq f$ (cf. équation (2.10)), *i.e.* :

$$f \oslash g \ = \ \bigoplus \{x \mid x * g \preccurlyeq f\}.$$

Preuve.

- Tout d'abord, l'opération d'inf-convolution $*$ étant une loi interne du dioïde de fonctions $(\mathcal{F}, \oplus, *)$, il s'agit d'une application isotone. D'après la remarque 3.7, cet opérateur est commutatif et distribue sur la somme $\oplus$. Ainsi, $*$ est une application semi-continue inférieurement. De plus, ε est absorbant pour ce produit : $f * \varepsilon = \varepsilon * f = \varepsilon$. Donc, d'après le théorème 2.49, l'application $*$ est résiduable.
- Ensuite, puisque l'application $*$ est résiduable, alors l'équation $x * a \preccurlyeq b$ admet $b \oslash a$ comme plus grande solution. Autrement dit :

$$\begin{aligned}
& & x * a &\preccurlyeq b, & \\
\Leftrightarrow & & \forall t,\ (x * a)(t) &\preccurlyeq b(t), & \\
\Leftrightarrow & \forall t, \bigoplus_{\tau \geq 0} \{x(\tau) + a(t - \tau)\} &\preccurlyeq b(t) & & \text{cf. équation (3.4).}
\end{aligned}$$

Soit :

$$\forall t,\ \begin{cases} x(0)+a(t) & \preccurlyeq & b(t), \\ x(1)+a(t-1) & \preccurlyeq & b(t), \\ x(2)+a(t-2) & \preccurlyeq & b(t), \\ & \cdots & , \end{cases} \Leftrightarrow \begin{cases} x(0) & \preccurlyeq & \bigwedge\limits_{t\geq 0}\{b(t)-a(t)\}, \\ x(1) & \preccurlyeq & \bigwedge\limits_{t\geq 0}\{b(t)-a(t-1)\}, \\ x(2) & \preccurlyeq & \bigwedge\limits_{t\geq 0}\{b(t)-a(t-2)\}, \\ & \cdots & , \end{cases}$$

$$\Leftrightarrow \quad \forall s,\ x(s) \preccurlyeq \bigwedge_{t\geq 0}\{b(t)-a(t-s)\} = (b\oslash a)(s).$$

Ce calcul de la déconvolution se retrouve dans [Chang 2000, Lem. 4.1.9]. □

Remarque 3.30 Il est important de noter que cet opérateur de déconvolution est connu jusqu'à présent sous le nom de quotient à droite (cf. équation (2.8)). Toutes les propriétés associées au quotient à droite dans le théorème 2.51 s'appliquent évidemment à l'opération de déconvolution. En outre, puisque $(\mathcal{F}, \oplus, *)$ est un dioïde commutatif, alors $f \oslash g = g \backslash f$ (cf. remarque 2.50).

Remarque 3.31 L'opération de déconvolution $\oslash$ n'est pas une opération close à l'ensemble $\mathcal{F}$ puisque $(f \oslash g)(t) \geq 0$ pour $t < 0$.

2.2 Modélisation par des flux et des contraintes

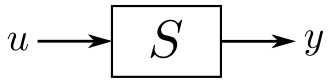
2.2.1 Flux d'entrée et de sortie

Un réseau de communication [8] peut être vu comme une boîte noire notée S avec un flux d'entrée u et un flux de sortie y (cf. figure 3.7a). La valeur $u(t)$ correspond à la quantité totale de données introduite dans le système jusqu'au temps t, tandis que $y(t)$ correspond à la quantité totale de données qui a quitté le système depuis t. Ainsi, ces flux d'entrée et de sortie, décrits par des fonctions cumulatives (donc croissantes), et considérés comme positifs (l'arrivée des données n'est supposée commencer qu'à l'instant $t = 0$), appartiennent à l'ensemble $\mathcal{F}$ (cf. figure 3.7b).

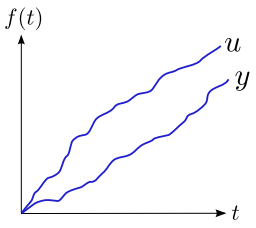
Remarque 3.32 Les flux $u, y \in \mathcal{F}$ peuvent être considérés comme les fonctions compteurs utilisées dans la modélisation de la section 1 de ce chapitre (cf. définition 3.3 de l'ensemble $\mathcal{F}_c$). Cependant ces fonctions ne sont ici pas nécessairement

8. Pouvant être découpé en simples éléments de réseau ou en sous-systèmes.

constantes par morceaux.



(a) Élément de réseau S



(b) Fonctions représentatives des flux u et y

Figure 3.7 Flux d'entrée u et de sortie y d'un réseau de communication S.

La principale hypothèse de fonctionnement faite entre le flux d'entrée et celui de sortie est donnée par l'inégalité suivante :

$$\forall t,\ u(t) \geq y(t) \quad \Leftrightarrow \quad u \preccurlyeq y.$$

Ainsi, la quantité totale de données quittant le réseau est toujours plus petite ou égale à celle entrant. Il peut arriver que cette relation se transforme en une stricte inégalité. Les périodes où cela arrive sont appelées des périodes chargées [9] et sont définies ci-dessous.

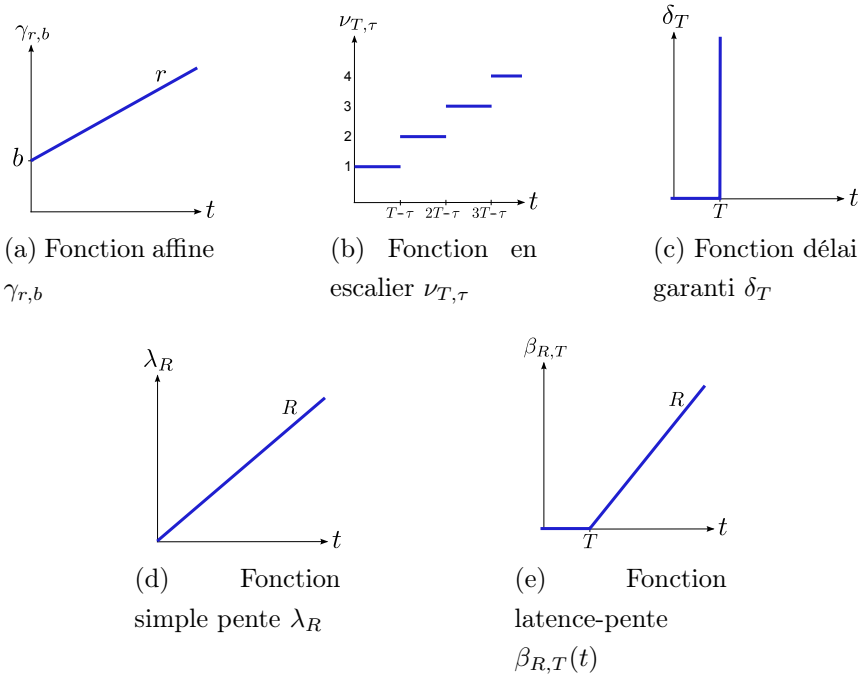
Définition 3.33 (Période chargée). Une période chargée est un intervalle de temps pour lequel $\forall t \geq 0$, il existe un $\tau \in [0, t]$ tel que $u(\tau) > y(\tau)$. En d'autres termes, à l'instant τ, il y a nécessairement des paquets dans S, soit $u(\tau) - y(\tau) > 0$.

2.2.2 Différentes courbes de contraintes

Le principe du Network Calculus repose sur l'application de courbes de contraintes tant sur les flux d'entrée et de sortie u et y, que sur le service rendu par le réseau S. Ces courbes sont appelées *courbe d'arrivée* ou *contrainte sur la sortie* quand elles sont appliquées aux flux et permettent alors de modéliser un contrat de trafic. Les courbes appelées *courbes de service* permettent quant à

9. « Backlogged period » en anglais.

elles de caractériser les éléments du réseau. Les formes les plus répandues dans la littérature [10] sont répertoriées ci-dessous, elles appartiennent toutes à l'ensemble $\mathcal{F}$ (cf. [Le Boudec 2001]).



(a) Fonction affine $\gamma_{r,b}$

(b) Fonction en escalier $\nu_{T,\tau}$

(c) Fonction délai garanti δ_T

(d) Fonction simple pente λ_R

(e) Fonction latence-pente $\beta_{R,T}(t)$

Figure 3.8 Quelques exemples de courbes de contraintes.

Définition 3.34 (Courbe affine $\gamma_{r,b}$). La courbe notée $\gamma_{r,b}$ et illustrée figure 3.8a, est une courbe de contrainte de forme affine définie par :

$$\gamma_{r,b}(t) \triangleq \begin{cases} 0 & \text{pour } t < 0, \\ rt + b & \text{pour } t \geq 0. \end{cases} \tag{3.16}$$

b représente la quantité maximale de données émise instantanément (phénomène également appelé *rafale maximale*) et r correspond au débit moyen maximal des données.

Définition 3.35 (Courbe en escalier $\nu_{T,\tau}$). La courbe notée $\nu_{T,\tau}$ et illustrée

10. La liste est donc non exhaustive.

figure 3.8b, est une courbe de contrainte en escalier [11] définie par :

$$\nu_{T,\tau}(t) = \begin{cases} 0 & \text{pour } t < 0, \\ \left\lceil \frac{t+\tau}{T} \right\rceil & \text{pour } t \geq 0, \end{cases} \tag{3.17}$$

où $\lceil a \rceil \in \mathbb{Z}$ représente le plus petit entier supérieur à $a \in \mathbb{Q}$. T correspond à l'intervalle fixé entre les données et τ à la tolérance d'arrivée au plus tôt.

Définition 3.36 (Courbe de délai garanti δ_T). La courbe notée δ_T et illustrée figure 3.8c, est une courbe de contrainte du type délai garanti [12] définie par :

$$\delta_T(t) = \begin{cases} 0 & \text{pour } t \leq T, \\ +\infty & \text{pour } t > T. \end{cases} \tag{3.18}$$

T correspond au délai maximum autorisé pour toutes les données.

Remarque 3.37 Notons que cette courbe δ_T est une fonction élémentaire (cf. définition 3.18), *i.e.* $\delta_T = \Delta_T^0$.

Définition 3.38 (Courbe simple pente λ_R). La courbe notée λ_R et illustrée figure 3.8d, est une courbe de contrainte du type simple pente [13] définie par :

$$\lambda_R(t) = \begin{cases} 0 & \text{pour } t < 0, \\ Rt & \text{pour } t \geq 0. \end{cases} \tag{3.19}$$

R correspond au débit moyen maximal des paquets.

Définition 3.39 (Courbe latence-pente $\beta_{R,T}$). La courbe notée $\beta_{R,T}$ et illustrée figure 3.8e, est une courbe de contrainte du type latence-pente [14] définie par :

$$\beta_{R,T}(t) = (\lambda_R * \delta_T)(t) = \begin{cases} 0 & \text{pour } t < 0, \\ R(t-T) & \text{pour } t \geq 0. \end{cases} \tag{3.20}$$

T correspond au délai de propagation des données avec un débit fixe donné par R.

11. « Stair case » en anglais.
12. « Burst delay » en anglais.
13. « Peak rate » en anglais.
14. « Latency rate » en anglais.

2.3 Courbe d'arrivée

Une première contrainte appliquée au réseau, ou plus particulièrement à son flux d'entrée, est la notion de courbe d'arrivée[15]. Il arrive en effet qu'une rafale de trafic caractérisée par une forte augmentation de la fonction u durant un intervalle temporel de faible amplitude, provienne lors du fonctionnement du réseau. Cette courbe d'arrivée permet alors de régulariser le flux en limitant, à tout instant, le nombre de données à entrer dans le système.

Définition 3.40 (Courbe d'arrivée). Un flux $u \in \mathcal{F}$ est contraint par une courbe d'arrivée $\alpha \in \mathcal{F}$ si $\forall 0 \leq \tau \leq t \in \mathbb{R}$, $u(t) - u(\tau) \leq \alpha(t-\tau)$. La quantité de données arrivant entre le temps τ et le temps t est donc au plus $\alpha(t-\tau)$ (u est dit α-lissé).

Cette définition peut s'écrire d'une manière plus algébrique comme le montre la proposition suivante issue de [Chang 2002, Lem 3.2] et illustrée figure 3.9.

Proposition 3.41 Un flux $u \in \mathcal{F}$ est contraint par une courbe d'arrivée $\alpha \in \mathcal{F}$ si :

$$u \succcurlyeq \alpha u \quad \Leftrightarrow \quad u = \alpha^\star u. \tag{3.21}$$

De par cette équivalence, $\alpha^\star$ est également une courbe d'arrivée pour u.

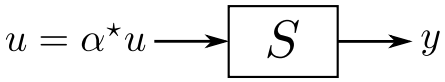


Figure 3.9 Courbe d'arrivée α appliquée au flux d'entrée u.

Preuve. Tout d'abord, grâce à la définition 3.5 de l'inf-convolution :

$$\begin{aligned} \forall 0 \leq \tau \leq t,\ u(t) - u(\tau) \leq \alpha(t-\tau) \quad &\Leftrightarrow \quad \forall 0 \leq \tau \leq t,\ u(t) \leq \alpha(t-\tau) + u(\tau), \\ &\Leftrightarrow \quad u \succcurlyeq \alpha u. \end{aligned}$$

Ensuite, grâce à l'isotonie de l'inf-convolution, cette inégalité peut être écrite comme suit :

$$u \succcurlyeq \alpha u \ \Rightarrow\ \alpha u \succcurlyeq (\alpha^2 u) \ \Rightarrow\ (\alpha^2 u) \succcurlyeq (\alpha^3 u) \ \Rightarrow\ \dots\ ,$$

15. Également appelée enveloppe, ou enveloppe de flux (cf. [Cruz 1991a, Cruz 1991b], [Chang 2000] et [Fidler 2010]).

et donc :

$$u = u \oplus (\alpha u) \oplus (\alpha^2 u) \oplus \ldots = \bigoplus_{n \geq 0} \alpha^n u.$$

Finalement :

$$u \succcurlyeq \alpha u \quad \Leftrightarrow \quad u = \alpha^\star u.$$

□

Proposition 3.42 (Courbe d'arrivée minimale). Une courbe d'arrivée minimale pour un flux d'entrée u est donnée par la déconvolution $u \oslash u$ soit :

$$u \oslash u \succcurlyeq \alpha \quad \text{et ainsi} \quad u \oslash u \succcurlyeq \alpha^\star. \tag{3.22}$$

Preuve. Puisque $x \mapsto x \oslash a$ est isotone (cf. équation(2.14)) et $(xa) \oslash a \succcurlyeq x$ (cf. équation(2.17)), alors :

$$u \succcurlyeq \alpha u \quad \Leftrightarrow \quad u \oslash u \succcurlyeq (\alpha u) \oslash u \succcurlyeq \alpha.$$

La preuve est la même pour $u \oslash u \succcurlyeq \alpha^\star$.

□

Lorsque plusieurs flux arrivent sur un même élément de réseau, une courbe d'arrivée globale peut alors être proposée.

Théorème 3.43 (Agrégation de trafic [Fidler 2010]). Soit S un élément de réseau dans lequel arrivent n flux d'entrée, chacun contraint par une courbe d'arrivée α_i, $\forall i \in [0, n]$. La courbe d'arrivée globale de cet élément réseau est donnée par :

$$\bigoplus_{i \in [0,n]} \alpha_i.$$

Cette courbe correspond alors au minimum des courbes d'arrivée.

Enfin, selon les besoins de la modélisation, les courbes d'arrivée peuvent prendre plusieurs formes parmi celles proposées dans la figure 3.8. Deux exemples d'utilisation sont proposés ci-dessous.

Exemple 3.44 (Sceau percé [Le Boudec 2001, Prop. 1.2.3]). Un équipement du type sceau percé[16] contraint le flux d'entrée par une courbe d'arrivée affine du type $\alpha = \gamma_{r,b}$ (cf. équation (3.16)). En effet, un tel élément ne laisse passer qu'une

16. « Leaky bucket » en anglais.

certaine quantité de données (il s'agit du buffer, soit la capacité du sceau) selon un certain débit (le diamètre du trou du sceau), et les données faisant exploser le buffer sont délaissées ou marquées selon une plus faible priorité.

Exemple 3.45 (Tampon de réception [Grieu 2004]). Un autre exemple est celui du tampon de réception qui contraint le flux d'entrée par une courbe d'arrivée en escalier du type $\alpha = \nu_{T,\tau}$ (cf. équation (3.17)). Ce type de tampon est utilisé lorsqu'il est nécessaire de recevoir un paquet en intégralité avant de le traiter, par exemple lors d'une vérification d'intégrité du paquet.

2.4 Courbe de service

Une seconde contrainte utilisée dans le Network Calculus est appliquée directement aux éléments du réseau et permet de caractériser les services qu'ils fournissent. La notion de courbe de service apparaît alors et peut être vue, en lien avec la modélisation des systèmes (min,+)-linéaires, comme une borne inférieure (il s'agit d'un service minimum) sur la réponse impulsionnelle du système (cf. remarque 3.10).

Définition 3.46 (Courbe de service). Un élément de réseau S fournit un service $\beta \in \mathcal{F}$ pour tout flux d'entrée u si $\forall t \geq 0$, $y(t) \geq \min_{\tau \in [0,t]}\{\beta(\tau) + u(t-\tau)\}$. Algébriquement, cela revient à dire que :

$$y \preccurlyeq \beta u.$$

Cette notion de courbe de service est pour l'instant définie dans le but de borner inférieurement le service rendu par le réseau. Il s'agit donc de service minimum. Il est également possible de borner ce service supérieurement comme le montre la proposition suivante.

Proposition 3.47 (Intervalle de courbe de service). Soit $[\ \underline{\beta}\ ,\ \overline{\beta}\ ]$ un intervalle de service fourni pour tout flux d'entrée u par un élément de réseau S (cf. figure 3.10). Dans cet intervalle, $\underline{\beta} \preccurlyeq \overline{\beta}$ avec $\overline{\beta} \in \mathcal{F}$ correspondant au service minimum fourni et $\underline{\beta} \in \mathcal{F}$ au service maximum. Ainsi, le flux de sortie y est également inclus dans un intervalle :

$$\underline{\beta}u \preccurlyeq y \preccurlyeq \overline{\beta}u \quad \Leftrightarrow \quad y \in [\ \underline{\beta}u\ ,\ \overline{\beta}u\ ]. \tag{3.23}$$

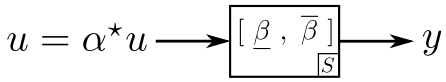


Figure 3.10 Intervalle de courbe de service d'un élément de réseau S.

Un résultat important dans la théorie du Network Calculus est lié à la recherche de la courbe de service globale d'un réseau dont les éléments, ayant chacun une courbe de service propre, sont mis en série[17].

Théorème 3.48 (Mise en série [Chang 2000, Th. 2.3.3]). Soit S un réseau composé de n éléments mis en série et dont les courbes de service sont notées β_i, $\forall i \in [0, n]$. La courbe de service globale de ce réseau est donnée par :

$$\mathop{\ast}_{i\in[0,n]} \beta_i.$$

Cette courbe correspond à l'inf-convolution des courbes de service de tous les éléments concaténés.

Contrairement à une analyse de proche en proche, soit élément par élément, une telle mise en série fournit des bornes sur les performances du réseau bien meilleures comparées à celles obtenues sans cette convolution des courbes de service (cf. [Le Boudec 2001, §1.4.3.]). Ce principe est connu en anglais sous le nom de « pay bursts only once ».

Les courbes de service assignées à chaque élément du réseau doivent être choisies avec soin selon les caractéristiques de l'élément représenté. Deux exemples issus de la thèse [Grieu 2004] et reprenant les formes décrites dans la figure 3.8 sont donnés ci-dessous.

Exemple 3.49 (Lien à délai garanti). La courbe de service à délai garanti notée $\beta = \delta_T$ est la forme la plus simple d'une courbe de service (cf. équation (3.18)). Le délai maximum pour toutes les données est alors borné par un temps fixe T. Ce type de courbe permet par exemple de modéliser le délai de propagation dû à un lien physique, ou un élément de réseau qui ne traite pas les données immédiatement.

Exemple 3.50 (Élément à débit constant). Lorsque l'élément du réseau peut

17. Également dit concaténés ou mis en tandem.

servir R données par unité de temps, la courbe de service utilisée est alors une courbe affine de simple pente (cf. équation (3.19)) notée $\beta = \lambda_R$. Ce type d'élément permet par exemple de modéliser un débit de sortie égal à celui du lien physique qui lui est accordé.

2.5 Contrainte sur la sortie

Grâce à la courbe d'arrivée et à la courbe de service, une courbe d'arrivée sur la sortie de l'élément de réseau, également appelée contrainte sur la sortie, peut être calculée. Ce résultat permet notamment de réaliser les calculs de proche en proche et de propager les résultats tout le long du réseau.

Définition 3.51 (Contrainte sur la sortie α'). Soit un flux u contraint par une courbe d'arrivée $\alpha^\star$ et qui traverse un réseau offrant un intervalle de service $[\ \underline{\beta}\ ,\ \overline{\beta}\ ]$. Le flux de sortie y de ce réseau est également contraint par une courbe α' définie par :

$$\alpha' = (\alpha^\star \underline{\beta}) \not\oslash \overline{\beta}.$$

Proposition 3.52 Soit un flux u contraint par une courbe d'arrivée $\alpha^\star$ tel que $u = \alpha^\star u$, et qui traverse un réseau offrant un intervalle de courbes de service $[\ \underline{\beta}\ ,\ \overline{\beta}\ ]$. Le flux de sortie y de ce réseau est contraint par la courbe α' si :

$$y \succcurlyeq \alpha' y \quad \Rightarrow \quad y \succcurlyeq ((\alpha^\star \underline{\beta}) \not\oslash \overline{\beta}) y.$$

Autrement dit, la contrainte α' peut être vue comme une courbe d'arrivée sur la sortie y.

Preuve. Puisque $\overline{\beta} u \succcurlyeq y \succcurlyeq \underline{\beta} u$ (cf. équation (3.23)), alors :

$$\begin{aligned}
y \not\oslash y &\succcurlyeq (\underline{\beta} u) \not\oslash y && \text{cf. équation (2.14)},\\
&\succcurlyeq (\underline{\beta} u) \not\oslash (\overline{\beta} u) && \text{cf. équation (3.23)},\\
&= ((\underline{\beta} u) \not\oslash u) \not\oslash \overline{\beta} && \text{cf. équation (2.24)},\\
&\succcurlyeq (\underline{\beta} (u \not\oslash u)) \not\oslash \overline{\beta} && \text{cf. équation (2.25)},\\
&\succcurlyeq (\alpha^\star \underline{\beta}) \not\oslash \overline{\beta} && \text{cf. équation (3.22) et } * \text{ commutatif.}
\end{aligned}$$

Donc, selon l'équivalence (2.12) :

$$y \succcurlyeq ((\alpha^\star \underline{\beta}) \not\oslash \overline{\beta}) y.$$

□

Le fait de considérer un intervalle de courbe de service $[\ \underline{\beta}\ ,\ \overline{\beta}\ ]$ permet d'obtenir une contrainte sur la sortie plus précise, *i.e.* :

$$(\alpha^\star \underline{\beta}) \not\phi \overline{\beta} \succcurlyeq \alpha^\star \not\phi \overline{\beta} \quad \Leftrightarrow \quad \forall t, ((\alpha^\star \underline{\beta}) \not\phi \overline{\beta})(t) \leq (\alpha^\star \not\phi \overline{\beta})(t),$$

puisque $\alpha^\star \underline{\beta} \succcurlyeq \alpha^\star$ et $\not\phi$ est isotone (cf. équation (2.14)).

3 Mesure des écarts : théorie du second ordre

Faisant suite à la théorie du premier ordre assimilée à la modélisation linéaire des SED (représentation d'état, réponse impulsionnelle, relation de transfert, etc.), la théorie du second ordre a été introduite par l'équipe Max-Plus dans les années 90 via les ouvrages [MaxPlus 1991], [Baccelli 1992, Sec. 6.6] et [Cohen 1993]. Elle permet d'étendre la notion classique de correlation aux algèbres (min,+) et (max,+).

En effet, les quantités considérées jusqu'à maintenant dans le cadre de la modélisation, le sont à travers des fonctions et opérateurs linéaires : fonctions cumulatives d'événements, lois des dioïdes sur lesquelles elles sont définies. Cependant, d'autres quantités telles que les temps de séjour ou les stocks, également appelés délais et backlogs dans le cadre du Network Calculus, sont pertinents pour ces systèmes et peuvent être considérés comme des indicateurs de performance.

En appliquant le principe de corrélation aux dioïdes, la théorie du second ordre permet d'évaluer ces informations en comparant les similitudes entres les signaux d'entrée et de sortie, ou entre les différentes contraintes appliquées au réseau. L'outil utilisé est la résiduation du produit d'inf-convolution.

3.1 Corrélation et autocorrélation

Le théorème suivant provenant de la théorie du second ordre des systèmes (min,+)-linéaires montre que les distances maximales entre deux fonctions croissantes v_1 et v_2 peuvent être calculées facilement tant dans le domaine événementiel que dans le domaine temporel. Ces distances sont obtenues grâce à la résiduée $\backslash\!\!\not$ (identique à l'opérateur $\not\phi$ dans le cas scalaire) de l'inf-convolution $*$ (cf. proposition 3.29).

L'opération d'inf-convolution $*$ étant commune aux dioïdes $(\mathcal{F}_c, \oplus, *)$, $(\mathcal{F}_{cp}, \oplus, *)$ et $(\mathcal{F}, \oplus, *)$ (cf. définitions 3.9, 3.17 et 3.28), l'ensemble des fonctions considéré peut être indifféremment $\mathcal{F}_c$ ou $\mathcal{F}_{cp}$ dans le cadre de la modélisation des systèmes (min,+)-linéaires, ou encore $\mathcal{F}$ dans le cadre du Network Calculus. Les résultats de cette section sont donnés à titre d'exemple sur l'ensemble $\mathcal{F}_c$.

Définition 3.53 (Corrélation et autocorrélation). Soient v_1 et v_2 deux fonctions de $\mathcal{F}_c$ pour lesquelles $v_1 \preccurlyeq v_2$ ($\Leftrightarrow \forall t, v_1(t) \geq v_2(t)$). La fonction $v_2 \diamond\!\!\backslash v_1$ est appelée corrélation de v_1 par v_2. Si $v_1 = v_2 = v$, alors $v \diamond\!\!\backslash v$ est appelée autocorrélation de v.

Théorème 3.54 (Distances maximales entre v_1 et v_2). La corrélation $v_2 \diamond\!\!\backslash v_1$ contient les distances maximales dans le domaine temporel et le domaine événementiel, notées respectivement $\Sigma_\tau(v_1, v_2)$ et $\Sigma_\nu(v_1, v_2)$, entre v_1 et $v_2 \in \mathcal{F}_c$. Plus précisément, $\Sigma_\tau(v_1, v_2)$ et $\Sigma_\nu(v_1, v_2)$ sont telles que :

$$\Sigma_\tau(v_1, v_2) = \inf_{\tau \geq 0} \{\tau \mid (v_2 \diamond\!\!\backslash v_1)(-\tau) \leq 0\}, \tag{3.24}$$

$$\Sigma_\nu(v_1, v_2) = (v_2 \diamond\!\!\backslash v_1)(0). \tag{3.25}$$

Ces distances sont illustrées dans la figure 3.11 dans le cas particulier où $v_1, v_2 \in \mathcal{F}_{cp}$.

Preuve. L'équation (3.15) donne l'expression de $v_2 \diamond\!\!\backslash v_1$ comme suit :

$$\forall t,\ (v_2 \diamond\!\!\backslash v_1)(t) = \bigwedge_{s \geq 0} \{v_2(s-t) \diamond\!\!\backslash v_1(s)\} = \sup_{s \geq 0} \{v_1(s) - v_2(s-t)\}.$$

Ainsi, pour la distance $\Sigma_\tau(v_1, v_2) = \inf_{\tau \geq 0}\{\tau \mid (v_2 \diamond\!\!\backslash v_1)(-\tau) \leq 0\}$:

$$(v_2 \diamond\!\!\backslash v_1)(-\tau) = \sup_{s \geq 0} \{v_1(s) - v_2(s+\tau)\}.$$

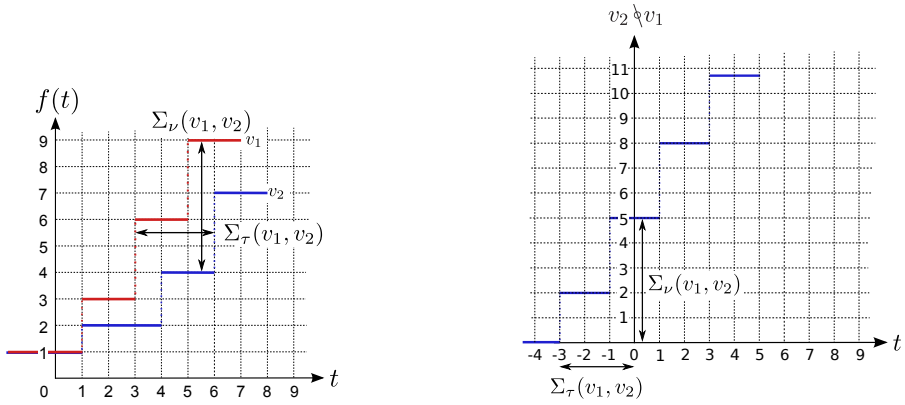
Il s'agit donc du plus petit τ tel que $\forall s,\ v_1(s) - v_2(s+\tau) \leq 0 \Leftrightarrow v_1(s) \leq v_2(s+\tau)$. Et pour la distance $\Sigma_\nu(v_1, v_2)$:

$$(v_2 \diamond\!\!\backslash v_1)(0) = \sup_{\tau \geq 0} \{v_1(\tau) - v_2(\tau)\}.$$

□

Propriété 3.55 (Périodicité). Si les fonctions v_1, v_2 et v sont périodiques, par exemple appartenant à l'ensemble $\mathcal{F}_{cp}$, alors $v_2 \diamond\!\!\backslash v_1$ et $v \diamond\!\!\backslash v$ le sont également.

Propriété 3.56 (Non-monotonie de l'autocorrélation). L'autocorrélation $v \diamond\!\!\backslash v$ n'est pas une opération monotone (cf. définition 2.42).



(a) Fonctions $v_1, v_2 \in \mathcal{F}_{cp}$ et leurs distances maximales $\Sigma_\tau(v_1, v_2)$ et $\Sigma_\nu(v_1, v_2)$

(b) Corrélation $v_2 \oslash v_1$ avec les distances maximales $\Sigma_\tau(v_1, v_2)$ et $\Sigma_\nu(v_1, v_2)$

Figure 3.11 Distances maximales dans les domaines temporel et événementiel entre deux fonctions v_1 et v_2 de $\mathcal{F}_{cp}$.

Exemple 3.57 Soit $v_1 = \Delta_1^1 \oplus \Delta_3^3 * (\Delta_2^3)^\star$ et $v_2 = \Delta_1^1 \oplus \Delta_4^2 \oplus \Delta_6^4 * (\Delta_2^3)^\star$ deux fonctions de $\mathcal{F}_{cp}$ (cf. figure 3.11a). La corrélation de v_1 par v_2 est la fonction suivante (cf. figure 3.11b) :

$$v_2 \oslash v_1 = \mathbf{\Delta_{-3}^0} \oplus \Delta_{-1}^2 \oplus \mathbf{\Delta_0^5} \oplus \Delta_1^5 * (\Delta_2^3)^\star.$$

La distance temporelle maximale est $\Sigma_\tau(v_1, v_2) = 3$, tandis que la distance événementielle maximale est $\Sigma_\nu(v_1, v_2) = 5$.

De manière plus générale, la corrélation $v_2 \oslash v_1$ contient la variation maximale entre v_1 et v_2 sur un horizon de taille t. Par exemple sur la figure 3.11, lorsque $t = 1$, la valeur de l'écart maximum entre $v_1(\tau)$ et $v_2(\tau - 1)$ est donnée par $(v_2 \oslash v_1)(1) = 5$. Cette variation maximale peut bien sûr être atteinte pour plusieurs τ inclus dans l'ensemble suivant :

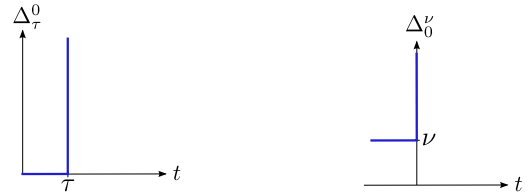
$$\arg\max_{\tau \geq 0} \{v_1(\tau) - v_2(\tau - t)\}. \tag{3.26}$$

Pour l'exemple de la figure 3.11 : $\tau \in \{5, 6, 7, 8, \ldots\}$.

Remarque 3.58 (Distances Σ_τ et Σ_ν infinies). Il est possible que $v_2 \oslash v_1 = \varepsilon$. Dans de tels cas, les distances maximales en temps et événements $\Sigma_\tau(v_1, v_2)$ et $\Sigma_\nu(v_1, v_2)$ sont infinies. En outre, $\Sigma_\tau(v_1, v_2)$ est également une distance infinie

lorsque $v_1(-\infty) \neq v_2(-\infty)$.

Lors des calculs liés à la manipulation des écarts, il est utile d'avoir des fonctions représentatives de ces distances Σ_τ et Σ_ν. Les fonctions élémentaires Δ_T^K fournissent une telle représentation avec Δ_τ^0 la fonction associée à un temps ou un délai τ (cf. figure 3.12a), et Δ_0^ν la fonction associée à un nombre d'événements ou une quantité de données ν (cf. figure 3.12b).



(a) Fonction Δ_τ^0 (b) Fonction Δ_0^ν

Figure 3.12 Fonctions Δ_τ^0 et Δ_0^ν représentatives d'un temps ou délai τ et d'un nombre d'événements ou d'une quantité de données ν.

3.2 Application aux systèmes (min,+)-linéaires

3.2.1 Stocks, temps de séjour et variations

Une des applications de cette théorie dans le cadre des systèmes (min,+)-linéaires est destinée au calcul des stocks et temps de séjour dans les GET. En particulier dans un contexte manufacturier, la distance $\Sigma_\nu(u, y)$ est une borne supérieure de l'encours du process, soient de toutes les pièces qui sont actuellement traitées ou en attente de l'être, entre l'entrée u et la sortie y.

Définition 3.59 (Stock et variation). Soit u et $y \in \mathcal{F}_c$ les fonctions compteurs en entrée et en sortie d'un système (min,+)-linéaire. Le stock à l'instant t à l'intérieur du système est donné par $\overline{\mathcal{S}}_{uy}(t)$ et est défini comme suit :

$$\overline{\mathcal{S}}_{uy}(t) = \overline{\mathcal{S}}_{uy}(0) + u(t) - y(t) = \overline{\mathcal{S}}_{uy}(0) + \mathcal{S}_{uy}(t),$$

avec $\overline{\mathcal{S}}_{uy}(0)$ le stock à l'instant $t = 0$ et $\mathcal{S}_{uy}(t)$ la variation du stock dans l'intervalle

$[0,t]$. Ainsi, $\forall t$:

$$\mathcal{S}_{uy}(t) = u(t) - y(t) = \overline{\mathcal{S}}_{uy}(t) - \overline{\mathcal{S}}_{uy}(0).$$

Théorème 3.60 ([Baccelli 1992, Th. 4.2]). Soient $u, y \in \mathcal{F}_c$ les fonctions compteurs d'entrée et de sortie d'un système (min,+)-linéaire. La variation du stock $\mathcal{S}_{uy}(t)$ est bornée supérieurement :

$$\mathcal{S}_{uy}(t) \succcurlyeq (y \oslash u)(0) \quad \Leftrightarrow \quad \mathcal{S}_{uy}(t) \leq (y \oslash u)(0).$$

Preuve. Il s'agit d'une conséquence directe de l'expression de la corrélation (cf. équation (3.15)) lorsque $t = 0$ soit :

$$(y \oslash u)(0) = \sup_{\tau \geq 0} \{u(\tau) - y(\tau)\} = \sup_{\tau \geq 0} \{\mathcal{S}_{uy}(\tau)\}.$$

□

Remarque 3.61 Le temps de séjour de pièces dans un système peut être borné de la même manière, en utilisant des fonctions dateurs d'événements. Cela est brièvement présenté dans [MaxPlus 1991].

3.2.2 Écarts de performances

Une autre application de l'utilisation de ces distances peut servir afin de mesurer des écarts de performance entre deux systèmes. L'objectif est par exemple la comparaison des différences entre l'état réel d'un système et son état estimé, calculé à partir d'une observation des trajectoires d'entrée et de sortie.

Corollaire 3.62 ([Hardouin 2010b]). Soient $x_i, \tilde{x}_i \in \mathcal{F}_c$ les trajectoires numérotées i d'un système (min,+)-linéaire. $\tilde{x}$ correspond à l'état estimé tandis que x correspond à l'état réel. En supposant que $\tilde{x}_i \preccurlyeq x_i$ (l'estimation tend à atteindre le système réel), la différence de mesure entre l'état réel et l'état estimé $\mathcal{E}_{\tilde{x}_i x}(t)$ est bornée supérieurement :

$$\mathcal{E}_{\tilde{x}_i x}(t) \succcurlyeq (x \oslash \tilde{x}_i)(0) \quad \Leftrightarrow \quad \mathcal{E}_{\tilde{x}_i x}(t) \leq (y \oslash u)(0).$$

Preuve. La preuve est la même que celle donnée pour le théorème 3.60.

□

3.3 Application au Network Calculus

3.3.1 Indicateurs de performances

Dans le contexte du Network Calculus (cf. [Bouillard 2007]), cette notion de corrélation apparaît afin d'analyser les pires cas de fonctionnement d'un réseau grâce à deux indicateurs de performance. Ces indicateurs appelés *délai* et *backlog* dans la littérature sont définis ci-dessous.

Définition 3.63 (Délai et backlog). Le délai noté $d(t)$ correspond au temps d'attente d'un paquet [18] selon un ordre FIFO tandis que le backlog noté $b(t)$ est la quantité de paquets dans un réseau à l'instant t. Soient u et $y \in \mathcal{F}$ les flux d'entrée et de sortie d'un réseau $\mathcal{S}$:

$$\begin{aligned} d(t) &\triangleq \inf_{\tau \geq 0}\{\tau \mid u(t) \leq y(t+\tau)\}, \\ b(t) &\triangleq u(t) - y(t). \end{aligned}$$

Ces informations sont données pour tout temps t dans le réseau. Cependant, selon le théorème 3.54 de la théorie du second ordre, des bornes supérieures sur leurs pires cas de fonctionnement peuvent être mesurées facilement. Ainsi, nous avons la possibilité de fournir deux sortes de distances pour un réseau S :

- si les flux d'entrée et de sortie u et y de S sont connus, alors les *délai et backlog maximums* de S peuvent être calculés,
- si la courbe d'arrivée $\alpha^\star$ appliquée à l'entrée u et la courbe de service minimum $\overline{\beta}$ de S sont connues alors des *bornes supérieures* des délai et backlog maximums peuvent être calculées.

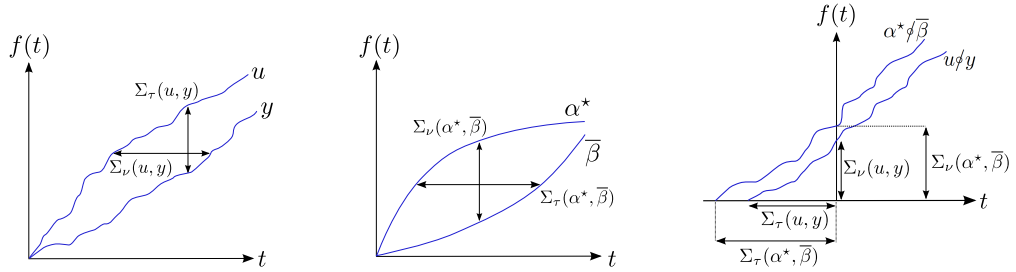
Ces mesures tout comme les liens existants entre elles, sont donnés dans la proposition suivante selon un point de vue strictement algébrique, à l'instar du théorème [Chang 2000, Th. 2.3.4].

Proposition 3.64 Soient $u, y \in \mathcal{F}$ les flux d'entrée et de sortie d'un réseau S tels que $u \preccurlyeq y$. Soient $\alpha^\star$ la courbe d'arrivée de l'entrée u telle que $u = \alpha^\star u$ et $\overline{\beta}$ la courbe de service minimum de S telle que $y \preccurlyeq \overline{\beta} u$. Alors :

$$\begin{array}{ccccccc} d(t) & \leq & \Sigma_\tau(u,y) = \inf\limits_{\tau \geq 0}\{(u \not\phi y)(-\tau) \leq 0\} & \leq & \Sigma_\tau(\alpha^\star, \overline{\beta}) = \inf\limits_{\tau \geq 0}\{(\alpha^\star \not\phi \overline{\beta})(-\tau) \leq 0\}, \\ b(t) & \leq & \Sigma_\nu(u,y) = (u \not\phi y)(0) & \leq & \Sigma_\nu(\alpha^\star, \overline{\beta}) = (\alpha^\star \not\phi \overline{\beta})(0). \end{array}$$

18. A nouveau, nous parlons indifféremment de bit, mot, paquet.

Les distances $\Sigma_\tau(u,y)$ et $\Sigma_\nu(u,y)$ sont les délai et backlog maximums de S tandis que $\Sigma_\tau(\alpha^\star,\overline{\beta})$ et $\Sigma_\nu(\alpha^\star,\overline{\beta})$ sont leurs bornes supérieures (voir figure 3.13).



(a) Flux u et y avec les distances $\Sigma_\tau(u,y)$ et $\Sigma_\nu(u,y)$

(b) Courbes $\alpha^\star$ et $\overline{\beta}$ avec les distances $\Sigma_\tau(\alpha^\star,\overline{\beta})$ et $\Sigma_\nu(\alpha^\star,\overline{\beta})$

(c) Corrélations $u \not\oslash y$ et $\alpha^\star \not\oslash \overline{\beta}$

Figure 3.13 Délai et backlog maximums et leurs bornes supérieures.

Preuve. Puisque $u = \alpha^\star u$ et $y \preccurlyeq \overline{\beta} u$ (cf. équations (3.21) et (3.23)) sont les relations entre les flux réels d'un réseau S et les contraintes sur ce réseau, l'inégalité suivante montre que depuis la corrélation $u \not\oslash y$, une autre corrélation entre $\alpha^\star$ et $\overline{\beta}$ est obtenue :

$$\begin{aligned}
y \preccurlyeq \overline{\beta} u \quad \Rightarrow \quad u \not\oslash y &\succcurlyeq u \not\oslash (\overline{\beta} u) && \text{cf. équation (2.14)},\\
&= (\alpha^\star u) \not\oslash (\overline{\beta} u) && \text{cf. équation (3.21)},\\
&= ((\alpha^\star u) \not\oslash u) \not\oslash (\overline{\beta}) && \text{cf. équation (2.24)},\\
&\succcurlyeq \alpha^\star \not\oslash \overline{\beta} && \text{cf. équation (2.17)}.
\end{aligned}$$

Donc, selon la relation d'ordre de $\overline{\mathbb{R}}_{min}$:

$$u \not\oslash y \succcurlyeq \alpha^\star \not\oslash \overline{\beta} \quad \Leftrightarrow \quad \forall t,\ (u \not\oslash y)(t) \leq (\alpha^\star \not\oslash \overline{\beta})(t).$$

□

Remarque 3.65 Puisque ces indicateurs de performances sont manipulés sous forme de fonctions, donc des scalaires et non des matrices, $(u \not\oslash y)(t) = (y \not\ominus u)(t)$. Nous conservons ici les habitudes d'écriture du Network Calculus en utilisant le quotient à droite $\not\oslash$.

Remarque 3.66 Dans la littérature du Network Calculus, les distances maximales $\Sigma_\tau(\alpha^\star, \overline{\beta})$ et $\Sigma_\nu(\alpha^\star, \overline{\beta})$ sont obtenues par les déviations horizontales et verticales entre les éléments de la corrélation $\alpha^\star \not| \overline{\beta}$, comme le montre la figure 3.13b.

Remarque 3.67 Ces notions d'indicateurs de performance sont indubitablement liées aux choix des courbes d'arrivée et de service. En effet, si $\alpha^\star \succcurlyeq \overline{\beta}$, alors $\alpha^\star \not| \overline{\beta} = \varepsilon$ (cf. remarque 3.58) et ainsi, les distances maximales Σ_τ et Σ_ν, *i.e.* le délai et le backlog, ne sont plus garanties.

3.3.2 Courbe d'arrivée optimale selon un délai ou backlog donné

Lorsque le service minimum $\overline{\beta}$ fourni par un réseau S est connu, il est possible de trouver la courbe d'arrivée optimale $\alpha^\star$ afin de respecter un délai ou backlog maximal donné. Par courbe d'arrivée optimale il faut comprendre la contrainte minimale appliquée sur les entrées afin d'éventuellement atteindre, mais surtout ne pas dépasser, le délai ou backlog donné. Cette contrainte minimale est obtenue en recherchant la courbe d'arrivée la plus petite selon l'ordre du dioïde, la plus grande selon l'ordre naturel. Si une courbe d'arrivée moins restrictive est choisie, le réseau sera sujet à des congestions et le délai de bout-en-bout maximum augmentera.

Dans un premier temps, le problème du point de vue de la performance temporelle est donné dans la proposition suivante.

Proposition 3.68 Soit $\overline{\beta}$ une courbe de service minimum d'un réseau S et τ un pire délai de bout-en-bout fixé. La courbe d'arrivée optimale $\hat{\alpha}^\star$ qui garantit le respect de τ est donné par :

$$\hat{\alpha}^\star = \bigwedge \{\alpha^\star \mid \alpha^\star \succcurlyeq \Delta^0_{-\tau} \overline{\beta}\} = (\Delta^0_{-\tau} \overline{\beta})^\star,$$

avec $\Delta^0_{-\tau}$ la fonction associée à τ (cf. figure 3.12a).

Preuve. Tout d'abord, la proposition 3.64 montre que la borne supérieure $\Sigma_\tau(\alpha^\star, \overline{\beta})$ d'un pire délai de bout-en-bout est donnée par la corrélation $\alpha^\star \not| \overline{\beta}$ comme suit (cf. équation (3.24)) :

$$\begin{aligned} \Sigma_\tau(\alpha^\star, \overline{\beta}) &= \inf_{\tau \geq 0} \{\tau \mid (\alpha^\star \not| \overline{\beta})(-\tau) \leq 0\}, \\ &= -\sup_{\tau \geq 0} \{\tau \mid (\alpha^\star \not| \overline{\beta})(\tau) \leq 0\}. \end{aligned}$$

En associant la distance $\Sigma_\tau(\alpha^\star, \overline{\beta})$ à la fonction $\Delta^0_{-\tau}$ avec $\tau = \Sigma_\tau(\alpha^\star, \overline{\beta})$, le pire délai de bout-en-bout à respecter, ce calcul est équivalent à l'inégalité suivante :

$$\alpha^\star \not\oslash \overline{\beta} \succcurlyeq \Delta^0_{-\tau}.$$

En effet, la fonction $\alpha^\star \not\oslash \overline{\beta}$ est choisie telle qu'elle soit graphiquement plus petite que la fonction $\Delta^0_{-\tau}$ représentant le délai maximum. Et grâce à l'équation (2.12) :

$$\alpha^\star \not\oslash \overline{\beta} \succcurlyeq \Delta^0_{-\tau} \quad \Leftrightarrow \quad \alpha^\star \succcurlyeq \Delta^0_{-\tau} \overline{\beta}.$$

Pour finir, il s'agit de trouver la courbe d'arrivée $\alpha^\star$ la plus petite selon l'ordre du dioïde, qui soit plus grande que $\Delta^0_{-\tau}\overline{\beta}$. Ainsi d'après l'équation (2.44) :

$$\bigwedge \{\alpha^\star \mid \alpha^\star = (\alpha^\star)^\star,\ \alpha^\star \succcurlyeq \Delta^0_{-\tau}\overline{\beta}\} = (\Delta^0_{-\tau}\overline{\beta})^\star.$$

□

Exemple 3.69 Soit un réseau S décrit par la figure 3.14. Le service fourni par l'élément S est borné inférieurement par la courbe de service $\overline{\beta}$ du type latence-pente (cf. équation (3.20)). Sa latence est de $16ms$ tandis que sa pente est de $100Mb/s$:

$$\overline{\beta}(t) = 16ms + 100Mb/s \cdot t.$$

Pour ce réseau, le pire délai de bout-en-bout τ à ne pas dépasser est fixé à $\tau = 90ms$. Ce délai est représenté par la fonction $\Delta^0_{-\tau}$. La proposition 3.68 est appliquée afin de trouver la courbe d'arrivée optimale $\hat{\alpha}^\star$ qui permet de respecter le délai τ. Ainsi :

$$\begin{aligned} \hat{\alpha}^\star = (\Delta^0_{-\tau}\overline{\beta})^\star \quad \Rightarrow \quad \hat{\alpha}^\star(t) &= (-90ms) * (16ms + 100Mb/s \cdot t), \\ &= 7,4Mb + 100Mb/s \cdot t. \end{aligned}$$

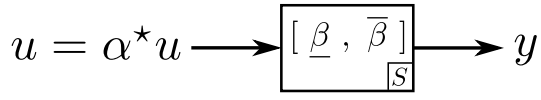


Figure 3.14 Réseau S dont le service minimum fourni est borné inférieurement par $\overline{\beta}$ et dont l'entrée u est contrainte par une courbe d'arrivée $\alpha^\star$.

Dans un second temps, la proposition suivante établit le problème du point de vue de la performance événementielle.

Proposition 3.70 Soit $\overline{\beta}$ une courbe de service minimum d'un réseau S et ν un pire backlog fixé. La courbe d'arrivée optimale $\hat{\alpha}^\star$ qui garantit le respect de ν est donnée par :

$$\hat{\alpha}^\star = \bigwedge\{\alpha^\star \mid \alpha^\star \succcurlyeq \Delta_0^\nu \overline{\beta}\} = (\Delta_0^\nu \overline{\beta})^\star,$$

avec Δ_0^ν la fonction associée à ν (cf. figure 3.12b).

Preuve. La preuve est identique à celle de la proposition 3.68. □

4 Conclusion

Les sections 1 et 2 de ce chapitre ont permis de présenter la modélisation des systèmes (min,+)-linéaires grâce à la théorie des dioïdes. Lors de cette modélisation, les trajectoires, ou flux, d'entrée u et de sortie y des éléments sont représentés par des fonctions compteurs définies de $\mathbb{R}$ dans $\overline{\mathbb{R}}_{min}$. Ces fonctions décrivent le cumul des événements en fonction du temps (cf. définition 3.3 de l'ensemble $\mathcal{F}_c$ et définition 3.26 de l'ensemble $\mathcal{F}$).

La notion classique de la réponse impulsionnelle est également disponible à travers la fonction de transfert h du système, ou grâce à une borne inférieure représentée par une courbe de service minimum fourni par l'élément de réseau (une borne supérieure peut également être proposée grâce à une courbe de service maximum). Dans tous les cas, la dynamique entrée/sortie d'un système (min,+)-linéaire peut alors s'exprimer par une inf-convolution (cf. équations (3.8) et (3.23)) :

$$y = h * u \quad \text{et} \quad \underline{\beta} * u \preccurlyeq y \preccurlyeq \overline{\beta} * u.$$

En outre, dans le cadre du Network Calculus, une contrainte supplémentaire appelée courbe d'arrivée est appliquée à l'entrée du réseau (cf. proposition 3.41) :

$$u = \alpha^\star * u,$$

afin notamment d'éviter les rafales de données qui pourraient le congestionner.

Un autre aspect de la modélisation abordé dans la section 3, est le calcul des écarts entre les entrées et les sorties des systèmes, ou encore entre les différentes

courbes de contraintes appliquées aux réseaux. En effet, grâce au principe de corrélation (cf. théorème 3.54) utilisant l'opérateur de déconvolution (soit la résiduation de l'inf-convolution, cf. proposition 3.29), il est possible de mesurer les performances des systèmes étudiés telles que les temps de séjour des produits / délais de transmission des paquets, ou encore les stocks internes / nombres de données en attente de service (cf. théorème 3.60 et proposition 3.64). Dans un contexte de système de production, d'autres utilisations de ces écarts sont également faites, comme par exemple les différences de performance entre un système réel et son estimation (cf. corollaire 3.62).

Une conclusion importante de ce chapitre est que selon le théorème 3.13, corroboré par l'introduction de [Chang 2000, Chap. 2], tout système (min,+)-linéaire peut s'exprimer par un nombre fini des combinaisons suivantes :

- mise en parallèle grâce à la somme $\oplus$,
- mise en série grâce à l'inf-convolution $*$,
- rebouclage grâce à l'étoile de Kleene * (ou clôture sous-additive).

Ces trois opérations sont donc indispensable à la modélisation de ces systèmes. Cependant, réaliser des calculs exacts de sommes, d'inf-convolutions ou d'étoiles de Kleene sur des fonctions croissantes et ultimement pseudo-périodiques de $\mathcal{F}_{cp}$ (cf. définition 3.14) peut consommer beaucoup de temps et d'espace mémoire si les régimes transitoires des résultats s'avèrent être très grands [19] (cf. exemple de la section 2.3.2 du chapitre 1). Le chapitre suivant tente de répondre à cette problématique en proposant des conteneurs pour ces fonctions de $\mathcal{F}_{cp}$. Les calculs réalisés sur les bornes de ces intervalles sont de faible complexité et les représentations informatiques sont également diminuées.

19. Les calculs sur des fonctions ultimement affines sont moins complexes mais l'opération de l'étoile de Kleene n'étant pas close pour ces fonctions (selon [Bouillard 2008b], le calcul peut fournir une fonction ultimement pseudo-périodique), le problème apparaît également.

Chapitre 4

Conteneur de systèmes (min,+)-linéaires

Sommaire

1 Motivations 79
2 Définition du conteneur 81
2.1 Transformée de Legendre-Fenchel des fonctions de $\mathcal{F}_{cp}$ 81
2.2 Fonctions affines et ultimement affines 83
2.3 Opérateurs d'approximation 86
2.4 Définition du conteneur 89
3 Opérations sur les conteneurs 93
3.1 Approximations convexe $\mathcal{C}_{vx}$ et concave $\mathcal{C}_{cv}$ 94
3.2 Fonction d'inclusion de la somme : $[\oplus]$ 94
3.3 Fonction d'inclusion de l'inf-convolution : $[*]$ 96
3.4 Fonction d'inclusion de l'étoile de Kleene : $^{[\star]}$ 98
4 Conclusion 103

1 Motivations

Le chapitre précédent a montré que les opérations de somme $\oplus$, d'inf-convolution $*$ et d'étoile de Kleene *, réalisées sur des fonctions ultimement pseudo-périodiques de l'ensemble $\mathcal{F}_{cp}$ (cf. définition 3.14), sont des opérations essentielles lors de la modélisation de systèmes (min,+)-linéaires. Cependant, les complexités de ces opérations dépendent de développements des régimes transitoires jusqu'à des valeurs pouvant être arbitrairement grandes (cf. exemple de la section 2.3.2 du chapitre 1). Ces développements coûtent à la fois en espace mémoire, et donc subséquemment en temps de calculs.

C'est pourquoi, l'objectif principal du travail présenté dans ce chapitre est d'obtenir des algorithmes efficaces permettant de manipuler ces fonctions ultimement pseudo-périodiques. Afin d'atteindre cet objectif, la fonction $f \in \mathcal{F}_{cp}$ n'est pas représentée de manière exacte[1], mais est approximée par un ensemble, plus précisément par un intervalle de fonctions, c'est-à-dire : $f \in \mathbf{f} = [\ \underline{f}\ ,\ \overline{f}\ ]$ avec $\underline{f} \preccurlyeq \overline{f}$.

Puis, les opérations entre deux intervalles de fonctions doivent être définies dans le but de contenir de manière garantie le résultat qui aurait été trouvé avec les fonctions exactes de $\mathcal{F}_{cp}$. Ces opérations sont inspirées de l'analyse par intervalles (cf. [Moore 1979, Jaulin 2001]) qui propose, pour $\mathbf{f}$ et $\mathbf{g}$ deux intervalles de fonctions, de réaliser le calcul sur tous les f et g appartenant respectivement à $[\ \underline{f}\ ,\ \overline{f}\ ]$ et $[\ \underline{g}\ ,\ \overline{g}\ ]$. Formellement, les opérations intervalles notées $\circ \in \{\oplus, *, ^\star\}$ sont définies par :

$$\mathbf{f} \circ \mathbf{g} = \{f \circ g \mid f \in \mathbf{f} = [\ \underline{f}\ ,\ \overline{f}\ ] \ \text{ et } \ g \in \mathbf{g} = [\ \underline{g}\ ,\ \overline{g}\ ]\}.$$

Par ailleurs, lorsque l'opération $\circ$ est monotone croissante, l'intervalle résultat s'exprime comme suit :

$$\mathbf{f} \circ \mathbf{g} = [\ \underline{f} \circ \underline{g}\ ,\ \overline{f} \circ \overline{g}\ ].$$

Malheureusement, les calculs sur les bornes $\underline{f} \circ \underline{g}$ et $\overline{f} \circ \overline{g}$ sont de la même complexité que les calculs sur les éléments de l'intervalle. En adoptant cette démarche, la précision du calcul exact est perdue et la complexité des algorithmes de calculs n'en est pas améliorée.

Ainsi, pour obtenir des algorithmes plus efficaces, des fonctions d'inclusion sont utilisées. Il s'agit d'opérations intervalles notées $[\circ] \in \{[\oplus], [*], {}^{[\star]}\}$ et qui sont telles que :

$$\mathbf{f}[\circ]\mathbf{g} \ \supset\ \mathbf{f} \circ \mathbf{g}.$$

En d'autres termes, la fonction d'inclusion $[\circ]$ qui s'applique à un intervalle contient de façon garantie le résultat de $f \circ g$, en ajoutant intrinsèquement du pessimisme. Le but est donc de trouver des fonctions d'inclusion de complexité algorithmique intéressante permettant toutefois d'obtenir des intervalles de taille aussi petite que possible.

Notre idée principale est donc de considérer des bornes pour ces intervalles, qui seront appelés des *conteneurs*, dont la représentation informatique nécessite moins

1. Ce qui est réalisé dans la librairie MinMaxGD (cf. [Cottenceau 2000] et annexe A.2).

d'espace mémoire que les fonctions de $\mathcal{F}_{cp}$ et menant à des algorithmes ayant une complexité algorithmique plus petite que celle utilisée dans MinMaxGD et COINC. En outre, le calcul de la borne supérieure du conteneur est réalisé directement à partir de la fonction exacte $f \in \mathcal{F}_{cp}$. En effet, la borne $\overline{f}$ correspond au plus grand élément de la classe d'équivalence de f modulo une transformée appelée transformée de Legendre-Fenchel et présentée ci-après. En complément, la borne $\underline{f}$ jouera le rôle de minorant de cette classe d'équivalence.

2 Définition du conteneur

2.1 Transformée de Legendre-Fenchel des fonctions de $\mathcal{F}_{cp}$

La transformée de Legendre-Fenchel appliquée aux fonctions de $\mathcal{F}_{cp}$ permet de définir un dioïde quotient (cf. section 2.4 du chapitre 2) noté $\mathcal{F}_{cp/\mathcal{L}}$ qui servira de contexte algébrique pour la borne supérieure du conteneur.

Définition 4.1 (Fonctions convexe et concave, [Baccelli 1992, Def. 3.37]). Une fonction $f : \mathbb{R} \mapsto \overline{\mathbb{R}}_{min}$ est dite convexe, respectivement concave, si son épigraphe noté $epi(f)$, respectivement son hypographe noté $hypo(f)$, est un ensemble convexe[2], *i.e.* :

$$\begin{aligned} epi(f) &\triangleq \{(x,y) \mid x \in \mathbb{R}, y \in \overline{\mathbb{R}}_{min},\ y \geq f(x) \Leftrightarrow y \preccurlyeq f(x)\}, \\ hypo(f) &\triangleq \{(x,y) \mid x \in \mathbb{R}, y \in \overline{\mathbb{R}}_{min},\ y \leq f(x) \Leftrightarrow y \succcurlyeq f(x)\}. \end{aligned}$$

L'ensemble des fonctions convexes, respectivement concaves, est noté $\mathcal{F}_{convex}$, respectivement $\mathcal{F}_{concav}$.

Définition 4.2 (Dioïde des fonctions convexes $(\mathcal{F}_{convex}, \oplus, \otimes)$). Le dioïde des fonctions convexes de $\mathcal{F}_{convex}$, munies du maximum point-à-point comme somme $\oplus$ et de la somme point-à-point comme produit $\otimes$ est noté $(\mathcal{F}_{convex}, \oplus, \otimes)$.

Définition 4.3 (Transformée de Legendre-Fenchel $\mathcal{L}$). La transformée de Legendre-Fenchel (cf. [Baccelli 1992, Rockafellar 1997, Fidler 2006]) appliquée à une fonction $f \in \mathcal{F}_{cp}$ est l'application $\mathcal{L}$ définie de $\mathcal{F}_{cp}$ dans $(\mathcal{F}_{convex}, \oplus, \otimes)$ par :

$$\mathcal{L}(f)(s) \triangleq \sup_t \{s.t - f(t)\}.$$

2. On rappelle que l'ordre $\preccurlyeq$ du dioïde $\overline{\mathbb{R}}_{min}$ est l'inverse de l'ordre naturel $\geq$.

Cette transformée est illustrée figure 4.1.

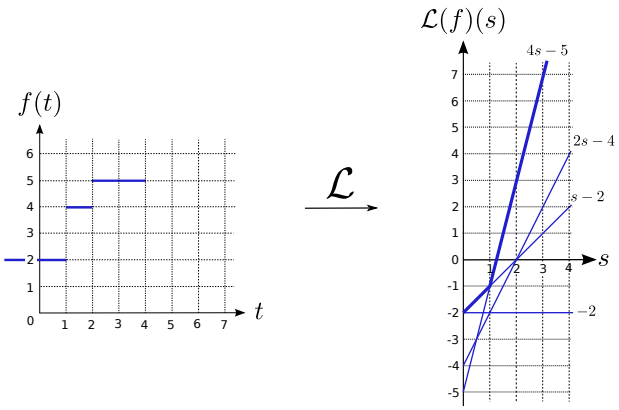


FIGURE 4.1 Transformée de Legendre-Fenchel d'une fonction $f = \Delta_1^2 \oplus \Delta_2^4 \oplus \Delta_4^5 \in \mathcal{F}_{cp}$.

Propriété 4.4 L'application $\mathcal{L}$ est un homomorphisme non injectif[3] de $(\mathcal{F}_{cp}, \oplus, *)$ dans $(\mathcal{F}_{convex}, \oplus, \otimes)$, c'est-à-dire que $\forall f, g \in \mathcal{F}_{cp}$:

$$\begin{aligned} \mathcal{L}(f \oplus g) &= \mathcal{L}(f) \oplus \mathcal{L}(g) &= \max(\mathcal{L}(f), \mathcal{L}(g)), \\ \mathcal{L}(f * g) &= \mathcal{L}(f) \otimes \mathcal{L}(g) &= \mathcal{L}(f) + \mathcal{L}(g). \end{aligned}$$

Définition 4.5 (Dioïde $\mathcal{F}_{cp/\mathcal{L}}$). Soit la relation d'équivalence suivante $\forall f, g \in \mathcal{F}_{cp}$:

$$\mathcal{L}(f) = \mathcal{L}(g) \quad \Leftrightarrow \quad f \overset{\mathcal{L}}{\equiv} g.$$

D'après le lemme 2.35 et la propriété 4.4, la relation $\overset{\mathcal{L}}{\equiv}$ est une congruence. Ainsi, le quotient de $\mathcal{F}_{cp}$ par $\overset{\mathcal{L}}{\equiv}$ forme le dioïde noté $\mathcal{F}_{cp/\mathcal{L}}$ (cf. lemme 2.36). Un élément de $\mathcal{F}_{cp/\mathcal{L}}$ est une classe d'équivalence modulo $\mathcal{L}$, notée $[f]_{\mathcal{L}}$, et qui contient toutes les fonctions de $\mathcal{F}_{cp}$ ayant la même transformée de Legendre-Fenchel. Les opérations $\oplus$ et $\otimes$ du dioïde quotient $\mathcal{F}_{cp/\mathcal{L}}$ sont alors définies comme suit :

$$\begin{aligned} [f]_{\mathcal{L}} \oplus [g]_{\mathcal{L}} &\triangleq [f \oplus g]_{\mathcal{L}}, \\ [f]_{\mathcal{L}} \otimes [g]_{\mathcal{L}} &\triangleq [f * g]_{\mathcal{L}}. \end{aligned}$$

3. Injectif : tout élément de $(\mathcal{F}_{convex}, \oplus, \otimes)$ admet au plus un antécédent dans $\mathcal{F}_{cp}$.

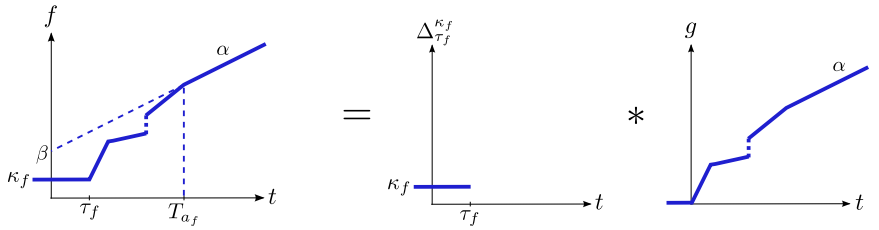
2.2 Fonctions affines et ultimement affines

L'encadrement des fonctions de $\mathcal{F}_{cp}$ est réalisé grâce à des fonctions croissantes, affines par morceaux et ultimement affines définies ci-dessous.

Définition 4.6 (Ensemble $\mathcal{F}_{aa}$). Une fonction $f : \mathbb{R} \mapsto \overline{\mathbb{R}}_{min}$ appartient à l'ensemble noté $\mathcal{F}_{aa}$ (cf. figure 4.2a) si elle est constante sur l'intervalle $]-\infty, \tau_f]$, et vérifie les propriétés suivantes sur $]\tau_f, +\infty[$:

- affine par morceaux, *i.e.* composée de combinaisons finies d'intervalles dans lesquels la fonction est affine[4],
- croissante,
- ultimement affine à partir d'un temps noté T_{a_f}, *i.e.* :

$$\exists T_{a_f} \geq \tau_f \text{ et } \exists \alpha, \beta \in \mathbb{R} \quad \text{tels que} \quad \forall t \geq T_{a_f},\ f(t) = \alpha t + \beta.$$



(a) Une fonction $f \in \mathcal{F}_{aa}$

(b) Qui peut être factorisée par l'inf-convolution $\Delta_{\tau_f}^{\kappa_f} * g$

Figure 4.2 Une fonction ultimement affine $f \in \mathcal{F}_{aa}$ et sa factorisation : $f = \Delta_{\tau_f}^{\kappa_f} * g$.

Définition 4.7 (Pente asymptotique σ). Soit f une fonction de $\mathcal{F}_{aa}$, sa pente asymptotique est définie par celle de sa partie ultimement affine : $\sigma(f) = \alpha$.

Propriété 4.8 (Factorisation d'une fonction $f \in \mathcal{F}_{aa}$). Une fonction $f \in \mathcal{F}_{aa}$ peut être décomposée de façon unique comme l'inf-convolution suivante illustrée figure 4.2b :

$$f = \Delta_{\tau_f}^{\kappa_f} * g.$$

4. Les parties affines sont liées par des points de non-différentiabilité.

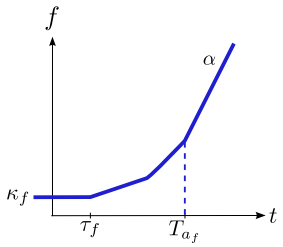
κ_f, τ_f et $g \in \mathcal{F}_{aa}$ sont donnés tels que :

$$\begin{cases} \kappa_f = \lim\limits_{t \to -\infty} f(t), \\ \tau_f = \max\{\ t \mid f(t) = \kappa_f\}, \end{cases} \qquad \text{et} \qquad g(t) = f(t - \tau_f) - \kappa_f,$$

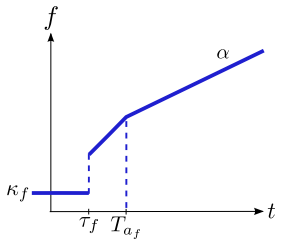
donc $g(0) = 0$ et $\sigma(g) = \sigma(f)$. La convolution par $\Delta_{\tau_f}^{\kappa_f}$ revient donc à faire une translation dans le plan de la fonction $f \in \mathcal{F}_{aa}$: κ_f est le déplacement événementiel et τ_f est le déplacement temporel.

Définition 4.9 (Ensemble $\mathcal{F}_{acx}$). Une fonction $f : \mathbb{R} \mapsto \overline{\mathbb{R}}_{min}$ appartient à l'ensemble noté $\mathcal{F}_{acx}$ (cf. figure 4.3a) si elle appartient à $\mathcal{F}_{aa}$ et si elle est convexe[5].

Définition 4.10 (Ensemble $\mathcal{F}_{acv}$). Une fonction $f : \mathbb{R} \mapsto \overline{\mathbb{R}}_{min}$ appartient à l'ensemble noté $\mathcal{F}_{acv}$ (cf. figure 4.3b) si elle appartient à $\mathcal{F}_{aa}$ et si elle est concave[6] sur $]\tau_f, +\infty[$.



(a) Une fonction convexe $f \in \mathcal{F}_{acx} \subset \mathcal{F}_{aa}$



(b) Une fonction $f \in \mathcal{F}_{acv} \subset \mathcal{F}_{aa}$ concave sur $]\tau_f, +\infty[$

Figure 4.3 Exemples de fonctions ultimement affines.

Définition 4.11 (Point extrémal, [Baccelli 1992, Def. 3.37]). Dans les fonctions convexes et concaves, un point de non-différentiabilité est appelé un point extrémal.

La factorisation d'une fonction de $\mathcal{F}_{aa}$ donnée par la propriété 4.8 fournit des propriétés intéressantes pour les fonctions de l'ensemble $\mathcal{F}_{acv}$.

5. Soit qui appartient à l'ensemble $\mathcal{F}_{convex}$.
6. Soit qui appartient à l'ensemble $\mathcal{F}_{concav}$.

Proposition 4.12 (Factorisation de $f \in \mathcal{F}_{acv}$). Une fonction $f \in \mathcal{F}_{acv}$ peut être décomposée comme suit (cf. figure 4.4) :

$$f = \Delta_{\tau_f}^{\kappa_f} * \Gamma_f, \tag{4.1}$$

avec $\Gamma_f \in \mathcal{F}_{acv}$, $\Gamma_f(0) = 0$ et $\sigma(\Gamma_f) = \sigma(f)$.

Preuve. Selon la propriété 4.8, $f = \Delta_{\tau_f}^{\kappa_f} * g$ avec $g \in \mathcal{F}_{acv}$ et $g(0) = 0$. Donc $g = \Gamma_f$.

□

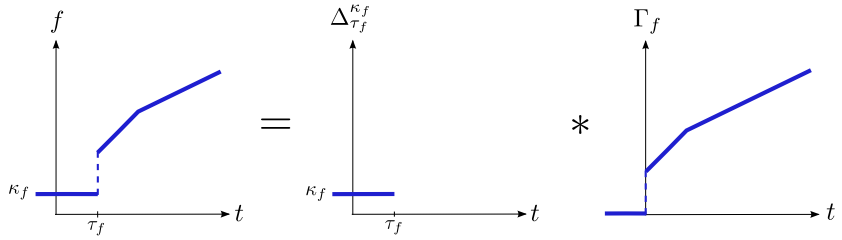


Figure 4.4 Factorisation d'une fonction concave $f \in \mathcal{F}_{acv}$: $f = \Delta_{\tau_f}^{\kappa_f} * \Gamma_f$.

Ainsi, la fonction Γ_f est la partie concave de f déplacée dans le plan par la fonction élémentaire $\Delta_{\tau_f}^{\kappa_f}$. Le théorème suivant sur les fonctions notées Γ fournit des égalités utiles afin de manipuler les opérations $*$ et * dans la section suivante.

Théorème 4.13 ([Le Boudec 2001, Th. 3.1.3, 3.1.6 et 3.1.9]). Soient Γ_1 et Γ_2 deux fonctions de $\mathcal{F}_{acv}$ pour lesquelles $\Gamma_1(0) = \Gamma_2(0) = 0$ (cf. proposition 4.12) :

$$\Gamma_1 * \Gamma_2 = \Gamma_1 \oplus \Gamma_2, \tag{4.2}$$

avec $\Gamma_1 \oplus \Gamma_2 \in \mathcal{F}_{acv}$ et $(\Gamma_1 \oplus \Gamma_2)(0) = 0$. De plus, Γ_1 est une fonction dite *sous-additive*, c'est-à-dire que :

$$\Gamma_1 = \Gamma_1^\star. \tag{4.3}$$

Remarque 4.14 En lien avec la notation de la propriété 4.8 et afin de faciliter la compréhension de cette section et de la suivante, la première fonction élémentaire $\Delta_{t_0}^{k_0}$ d'une fonction $f = \bigoplus_{i=0}^{+\infty} \Delta_{t_i}^{k_i} \in \mathcal{F}_{cp}$ (cf. équation (3.14)), sera notée $\Delta_{\tau_f}^{\kappa_f}$ par la suite.

2.3 Opérateurs d'approximation

Afin de construire le conteneur d'une fonction $f \in \mathcal{F}_{cp}$, deux opérateurs d'approximation sont définis et seront utilisés lors des calculs des fonctions d'inclusion. Le premier est convexe et permet une approximation de f par au-dessus (au sens de l'ordre $\preccurlyeq$), le second est concave et permet une approximation de f par en-dessous.

2.3.1 Approximation convexe $\mathcal{C}_{vx}$

Définition 4.15 (Approximation convexe $\mathcal{C}_{vx}$). Soit f une fonction de $\mathcal{F}_{cp}$. L'enveloppe convexe de f, également appelée approximation convexe et notée $\mathcal{C}_{vx}(f)$, est la plus petite fonction convexe supérieure à f (cf. figure 4.5). Ainsi :

$$\mathcal{C}_{vx}(f) \succcurlyeq f \quad \text{et} \quad \mathcal{C}_{vx}(f) \in \mathcal{F}_{acx}.$$

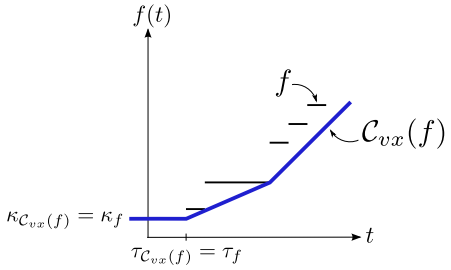


Figure 4.5 Approximation convexe de $f \in \mathcal{F}_{cp} : \mathcal{C}_{vx}(f) \in \mathcal{F}_{acx}$.

Propriété 4.16 L'application $\mathcal{C}_{vx}$ est un projecteur (cf. remarque 2.55), *i.e.* :

$$\mathcal{C}_{vx}(f) = \mathcal{C}_{vx}(\mathcal{C}_{vx}(f)).$$

En outre, les pentes asymptotiques de $\mathcal{C}_{vx}(f)$ et f sont égales, *i.e.* :

$$\sigma(\mathcal{C}_{vx}(f)) = \sigma(f),$$

et les points extrémaux de $\mathcal{C}_{vx}(f)$ appartiennent à la fonction f. En particulier :

$$\Delta^{\kappa_{\mathcal{C}_{vx}(f)}}_{\tau_{\mathcal{C}_{vx}(f)}} = \Delta^{\kappa_f}_{\tau_f}.$$

Cette approximation convexe $\mathcal{C}_{vx}(f)$ est fortement liée à la transformée de Legendre-Fenchel de f comme le montre le lemme ci-dessous.

Lemme 4.17 ([Baccelli 1992, Th. 3.38]). Soient $\mathcal{C}_{vx}(f)$ et $\mathcal{C}_{vx}(g)$ les enveloppes convexes de f et $g \in \mathcal{F}_{cp}$. Les fonctions f et g ont la même transformée de Legendre-Fenchel $\mathcal{L}$ si, et seulement si, elles ont la même enveloppe convexe :

$$\mathcal{L}(f) = \mathcal{L}(g) \quad \Leftrightarrow \quad \mathcal{C}_{vx}(f) = \mathcal{C}_{vx}(g).$$

Ainsi, différents éléments dans $\mathcal{F}_{cp}$ peuvent avoir la même transformée et être représentés par la même fonction convexe dans $\mathcal{F}_{acx}$.

D'après le lemme 4.17, il est donc possible de déterminer l'équivalence modulo la transformée de Legendre-Fenchel $\mathcal{L}$ en utilisant l'enveloppe convexe, et les calculs modulo la transformée $\mathcal{L}$ sont donc équivalents aux calculs modulo l'enveloppe convexe. Formellement, $\forall f, g \in \mathcal{F}_{cp}$:

$$[f]_{\mathcal{L}} \oplus [g]_{\mathcal{L}} = [f \oplus g]_{\mathcal{L}} \quad \Leftrightarrow \quad \mathcal{C}_{vx}(\mathcal{C}_{vx}(f) \oplus \mathcal{C}_{vx}(g)) = \mathcal{C}_{vx}(f \oplus g), \tag{4.4}$$

$$[f]_{\mathcal{L}} \otimes [g]_{\mathcal{L}} = [f * g]_{\mathcal{L}} \quad \Leftrightarrow \quad \mathcal{C}_{vx}(\mathcal{C}_{vx}(f) * \mathcal{C}_{vx}(g)) = \mathcal{C}_{vx}(f * g), \tag{4.5}$$

$$[f]^{\star}_{\mathcal{L}} = [f^{\star}]_{\mathcal{L}} \quad \Leftrightarrow \quad \mathcal{C}_{vx}(\mathcal{C}_{vx}(f)^{\star}) = \mathcal{C}_{vx}(f^{\star}). \tag{4.6}$$

Théorème 4.18 ([Baccelli 1992, Th. 6.19]). Soient f une fonction de $\mathcal{F}_{cp}$ et $[f]_{\mathcal{L}}$ sa classe d'équivalence dans le dioïde $\mathcal{F}_{cp/\mathcal{L}}$ (cf. definition 4.5). La fonction $\mathcal{C}_{vx}(f) \in \mathcal{F}_{acx}$ est le plus grand représentant de $[f]_{\mathcal{L}}$, *i.e.* :

$$[\ \mathcal{C}_{vx}(f)\ ]_{\mathcal{L}} = [f]_{\mathcal{L}} \quad \text{et} \quad \forall g \in [f]_{\mathcal{L}},\ g \preccurlyeq \mathcal{C}_{vx}(f).$$

Grâce au théorème 4.18, nous obtenons le moyen de réaliser les calculs sur le dioïde $\mathcal{F}_{cp/\mathcal{L}}$, puisqu'il détermine pour chaque classe d'équivalence un représentant canonique qui est l'enveloppe convexe des fonctions de la classe.

Il découle de toutes ces propriétés que si l'on applique l'enveloppe convexe à tous les calculs sur des fonctions de $\mathcal{F}_{cp}$, on simplifie les résultats en conservant néanmoins leurs classes d'équivalence modulo $\mathcal{L}$.

2.3.2 Approximation concave $\mathcal{C}_{cv}$

Tout d'abord, rappelons qu'une fonction $f \in \mathcal{F}_{cp}$ est constante sur $]-\infty, \tau_f]$ (puisque composée d'une première fonction élémentaire $\Delta^{\kappa_f}_{\tau_f}$, cf. remarque 4.14)

et ensuite croissante et constante par morceaux sur $]\tau_f, +\infty[$. Son enveloppe concave *classique* notée $conc(f)$ et correspondant à la plus grande fonction concave inférieure à f (selon l'ordre $\preccurlyeq$), est donc nécessairement la fonction $\varepsilon : t \mapsto +\infty$. C'est pourquoi, cette enveloppe concave $conc(f)$ n'est pas intéressante en tant qu'approximation de f.

Néanmoins, une borne inférieure de f peut être définie comme une fonction de $\mathcal{F}_{acv}$ qui est constante sur $]-\infty, \tau_f]$ et concave sur $]\tau_f, +\infty[$. Cette projection de $\mathcal{F}_{cp}$ dans $\mathcal{F}_{acv}$ est appelée approximation concave par la suite et est définie ci-dessous.

Définition 4.19 (Approximation concave $\mathcal{C}_{cv}$). Soit f une fonction de $\mathcal{F}_{cp}$. L'approximation concave de f, notée $\mathcal{C}_{cv}(f)$ et illustrée figure 4.6, est définie par :

$$\mathcal{C}_{cv}(f)(t) \triangleq \begin{cases} f(t) & \text{pour } t \leq \tau_f, \\ conc(f)(t) & \text{pour } t > \tau_f, \end{cases}$$

avec $\tau_f = \max\{t \mid f(t) = f(-\infty)\}$. Ainsi :

$$\mathcal{C}_{cv}(f) \preccurlyeq f \quad \text{et} \quad \mathcal{C}_{cv}(f) \in \mathcal{F}_{acv}.$$

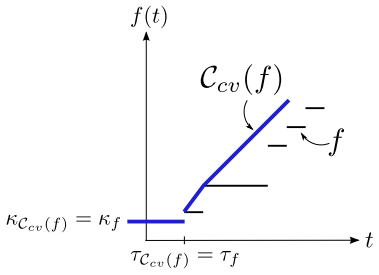


Figure 4.6 Approximation concave de $f \in \mathcal{F}_{cp}$: $\mathcal{C}_{cv}(f) \in \mathcal{F}_{acv}$.

Propriété 4.20 L'application $\mathcal{C}_{cv}$ est un projecteur (cf. remarque 2.55), *i.e.* :

$$\mathcal{C}_{cv}(f) = \mathcal{C}_{cv}(\mathcal{C}_{cv}(f)).$$

De plus :

$$\sigma(\mathcal{C}_{cv}(f)) = \sigma(f) \quad \text{et} \quad \Delta_{\tau_{\mathcal{C}_{cv}(f)}}^{\kappa_{\mathcal{C}_{cv}(f)}} = \Delta_{\tau_f}^{\kappa_f}.$$

Cette approximation concave $\mathcal{C}_{cv}(f)$ n'est pas symétrique à celle convexe $\mathcal{C}_{vx}(f)$. En particulier, les propriétés équivalentes aux équations (4.4) à (4.6) ne sont pas vérifiées. Néanmoins, $\mathcal{C}_{cv}$ est une application isotone, donc les propriétés suivantes sont satisfaites. Soient f et g deux fonctions de $\mathcal{F}_{cp}$:

$$\begin{cases} \mathcal{C}_{cv}(f) \preccurlyeq f \\ \mathcal{C}_{cv}(g) \preccurlyeq g \end{cases} \quad \Rightarrow \quad \begin{cases} \mathcal{C}_{cv}(f) \oplus \mathcal{C}_{cv}(g) \preccurlyeq f \oplus g, \\ \mathcal{C}_{cv}(f) * \mathcal{C}_{cv}(g) \preccurlyeq f * g, \\ \mathcal{C}_{cv}(f)^\star \preccurlyeq f^\star, \end{cases}$$

et

$$\begin{aligned} \mathcal{C}_{cv}(\mathcal{C}_{cv}(f) \oplus \mathcal{C}_{cv}(g)) &\preccurlyeq \mathcal{C}_{cv}(f \oplus g), \\ \mathcal{C}_{cv}(\mathcal{C}_{cv}(f) * \mathcal{C}_{cv}(g)) &\preccurlyeq \mathcal{C}_{cv}(f * g), \\ \mathcal{C}_{cv}(\mathcal{C}_{cv}(f)^\star) &\preccurlyeq \mathcal{C}_{cv}(f^\star). \end{aligned}$$

2.4 Définition du conteneur

L'objectif de ce chapitre est de construire une approximation de fonctions ultimement pseudo-périodiques de $\mathcal{F}_{cp}$, telle que les calculs sur la fonction approximée soient plus efficaces que sur l'originale. Pour ce faire, le conteneur proposé ci-dessous est un intervalle de fonctions couplé à une classe d'équivalence modulo $\mathcal{L}$.

Définition 4.21 (Ensemble F des conteneurs). L'ensemble des conteneurs considérés par la suite est l'ensemble noté $\mathbf{F}$ et défini par :

$$\mathbf{F} \triangleq \{ [\, \underline{f}\, ,\, \overline{f}\,]_{\mathcal{L}} \mid \underline{f} \in \mathcal{F}_{acv}\, ,\, \overline{f} \in \mathcal{F}_{acx}\, ,\, \sigma(\underline{f}) = \sigma(\overline{f}) \},$$

avec $[\, \underline{f}\, ,\, \overline{f}\,]_{\mathcal{L}}$ le sous-ensemble défini comme suit :

$$\begin{aligned} [\, \underline{f}\, ,\, \overline{f}\,]_{\mathcal{L}} &\triangleq [\, \underline{f}\, ,\, \overline{f}\,] \cap [\, \overline{f}\,]_{\mathcal{L}}, \\ &= \{ f \mid \underline{f} \preccurlyeq f \preccurlyeq \overline{f}\, ,\, [f]_{\mathcal{L}} = [\, \overline{f}\,]_{\mathcal{L}} \}. \end{aligned}$$

Un conteneur de $\mathbf{F}$ est donc un sous-ensemble d'un intervalle $[\, \underline{f}\, ,\, \overline{f}\,]$ dont les bornes $\underline{f}$ et $\overline{f}$ sont respectivement concave[7] et convexe, et dont les éléments sont équivalents à $\overline{f}$ modulo la transformée de Legendre-Fenchel $\mathcal{L}$: $\forall f \in [\, \underline{f}\, ,\, \overline{f}\,]_{\mathcal{L}}$, $\overline{f} = \mathcal{C}_{vx}(f)$ (avec $\mathcal{C}_{vx}$ l'approximation convexe définie en 4.15).

7. Sur $]\tau_f, +\infty[$.

2.4.1 Représentation canonique d'un conteneur de F

Grâce à la définition suivante d'un minorant de $[\ \overline{f}\ ]_{\mathcal{L}}$, il est possible d'obtenir une représentation canonique d'un conteneur de **F**.

Définition 4.22 (Minorant de $[\ \overline{f}\ ]_{\mathcal{L}}$). Soit $[\ \overline{f}\ ]_{\mathcal{L}}$ une classe d'équivalence du dioïde quotient $\mathcal{F}_{cp/\mathcal{L}}$ et $\overline{f} \in \mathcal{F}_{acx}$ son plus grand élément, c'est-à-dire $\overline{f} = \mathcal{C}_{vx}(\overline{f})$ (cf. théorème 4.18). Un minorant naturel de $[\ \overline{f}\ ]_{\mathcal{L}}$ est une fonction *finie* et constante par morceaux de $\mathcal{F}_c$ (cf. définition 3.3), notée $\Omega(\overline{f})$ et définie ci-dessous :

$$\Omega(\overline{f}) \triangleq \bigoplus_{i=0}^{n} \Delta_{t_i}^{k_i} \quad \text{et} \quad \forall t > t_n,\ \Omega(\overline{f})(t) = +\infty. \tag{4.7}$$

Les couples (t_i, k_i) sont les coordonnées des n points extrémaux de $\overline{f}$. Ainsi :

$$\Omega(\overline{f}) \preccurlyeq \overline{f} \quad \text{et} \quad \Delta_{t_0}^{k_0} = \Delta_{\tau_{\overline{f}}}^{\kappa_{\overline{f}}}.$$

Ce minorant $\Omega(\overline{f})$ est illustré figure 4.7, figure dans laquelle la zone grisée représente la classe d'équivalence[8] $[\ \overline{f}\ ]_{\mathcal{L}}$.

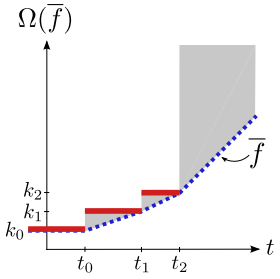


Figure 4.7 Minorant $\Omega(\overline{f})$ de la classe d'équivalence $[\ \overline{f}\ ]_{\mathcal{L}}$.

Remarque 4.23 La fonction $\Omega(\overline{f})$ est un minorant de $[\ \overline{f}\ ]_{\mathcal{L}}$ mais puisqu'elle n'a pas la même pente asymptotique que $\overline{f}$ ($\Omega(\overline{f})$ est infinie à partir de t_n), elle n'appartient pas à cette classe d'équivalence.

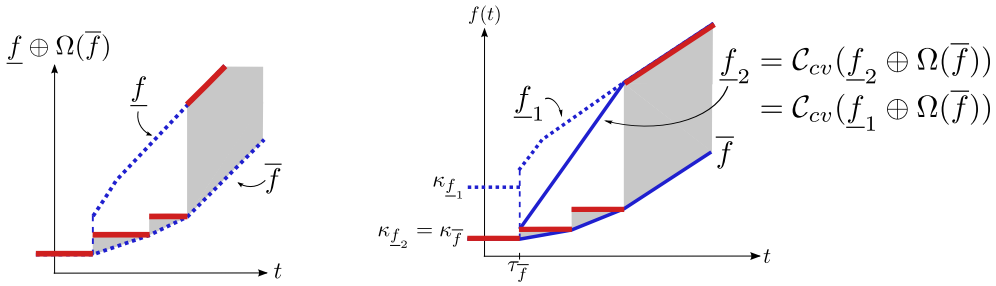
Grâce à ce minorant, si $f \in [\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}}$, alors $f \succcurlyeq \underline{f} \oplus \Omega(\overline{f})$. On peut donc noter que $\underline{f} \oplus \Omega(\overline{f})$, qui est illustré figure 4.8a, est le plus petit élément du conteneur $[\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}}$ (cf. définition 2.8), et que l'intervalle $[\ \underline{f} \oplus \Omega(\overline{f})\ ,\ \overline{f}\ ]$

8. Les fonctions qui ont les mêmes points extrémaux et la même pente asymptotique que f appartiennent à la classe d'équivalence $[\ \overline{f}\ ]_{\mathcal{L}}$.

(zone grisée de la figure 4.8a) contient de manière garantie la fonction $f \in [\, \underline{f}\ ,\ \overline{f}\,]_{\mathcal{L}}$.

Par conséquent, selon la définition d'un conteneur donné en 4.21, des intervalles différents $[\, \underline{f}_1\ ,\ \overline{f}\,]$ et $[\, \underline{f}_2\ ,\ \overline{f}\,]$ peuvent représenter un même conteneur dans $\mathbf{F}$ tant que $\underline{f}_1 \oplus \Omega(\overline{f}) = \underline{f}_2 \oplus \Omega(\overline{f})$ (cf. figure 4.8b). Formellement, soient $[\, \underline{f}_1\ ,\ \overline{f}\,]_{\mathcal{L}}$ et $[\, \underline{f}_2\ ,\ \overline{f}\,]_{\mathcal{L}}$ deux conteneurs de $\mathbf{F}$:

$$[\, \underline{f}_1\ ,\ \overline{f}\,] \neq [\, \underline{f}_2\ ,\ \overline{f}\,] \quad \text{tandis que} \quad [\, \underline{f}_1\ ,\ \overline{f}\,]_{\mathcal{L}} = [\, \underline{f}_2\ ,\ \overline{f}\,]_{\mathcal{L}}.$$



(a) Plus petit élément $\underline{f} \oplus \Omega(\overline{f})$ du conteneur

(b) Intervalles $[\, \underline{f}_1\ ,\ \overline{f}\,]$ et $[\, \underline{f}_2\ ,\ \overline{f}\,]$ différents, mais conteneurs identiques dans $\mathbf{F}$: $[\, \underline{f}_1\ ,\ \overline{f}\,]_{\mathcal{L}} = [\, \underline{f}_2\ ,\ \overline{f}\,]_{\mathcal{L}}$

Figure 4.8 Utilisation du minorant $\Omega(\overline{f})$ dans la construction d'un conteneur de $\mathbf{F}$.

Afin d'éviter de telles ambiguïtés, une représentation canonique pour ces conteneurs est définie en réalisant l'approximation concave de $\underline{f} \oplus \Omega(\overline{f})$.

Définition 4.24 (Représentation canonique d'un conteneur de F). La représentation canonique d'un conteneur $\mathbf{f} = [\, \underline{f}\ ,\ \overline{f}\,]_{\mathcal{L}} \in \mathbf{F}$ s'écrit comme suit :

$$[\, \mathcal{C}_{cv}\big(\underline{f} \oplus \Omega(\overline{f})\big)\ ,\ \overline{f}\,]_{\mathcal{L}} = [\, \underline{f}'\ ,\ \overline{f}\,]_{\mathcal{L}},$$

avec $\Delta_{\tau_{\underline{f}'}}^{\kappa_{\underline{f}'}} = \Delta_{\tau_{\overline{f}}}^{\kappa_{\overline{f}}}$ et $\mathcal{C}_{cv}$ l'approximation concave définie en 4.19. Dans la figure 4.8b, cette représentation canonique est donnée par le conteneur $[\, \underline{f}_2\ ,\ \overline{f}\,]_{\mathcal{L}}$ puisque $\underline{f}_2 = \mathcal{C}_{cv}\big(\underline{f}_2 \oplus \Omega(\overline{f})\big) = \mathcal{C}_{cv}\big(\underline{f}_1 \oplus \Omega(\overline{f})\big)$.

Remarque 4.25 Cette forme canonique sera également la représentation informatique retenue. En effet, outre l'avantage de représenter de manière unique un conteneur de $\mathbf{F}$, elle permet également de ne pas prendre en compte certains points « inutiles » de $\underline{f}$. Dans l'exemple de la figure 4.8b, les points de $\underline{f}_1$ situés

dans la zone entre $\underline{f}_1$ et $\underline{f}_2$ peuvent donc être supprimés afin de ne conserver que la représentation canonique $[\ \underline{f}_2\ ,\ \overline{f}\ ]_{\mathcal{L}}$.

2.4.2 Incertitude maximale d'un conteneur de F

Grâce à la forme canonique d'un conteneur $\mathbf{f} = [\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}} \in \mathbf{F}$ pour laquelle $\Delta^{\kappa_{\underline{f}}}_{\tau_{\underline{f}}} = \Delta^{\kappa_{\overline{f}}}_{\tau_{\overline{f}}}$ et $\sigma(\underline{f}) = \sigma(\overline{f})$ (cf. définitions 4.24 et 4.21), la distance maximale dans les domaines événementiel et temporel entre $\underline{f}$ et $\overline{f}$ est finie et la perte de précision due aux approximations est bornée. Cette incertitude maximale est définie ci-dessous.

Définition 4.26 (Incertitude maximale $\Theta_{\mathbf{f}}$ d'un conteneur f). Soit $\mathbf{f} = [\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}}$ un conteneur de $\mathbf{F}$ donné sous forme canonique. Son incertitude maximale notée $\Theta_{\mathbf{f}}$ est définie comme suit :

$$\Theta_{\mathbf{f}} \triangleq \{\ D_{\max}\ ,\ B_{\max}\ \},$$

avec $D_{\max}$ et $B_{\max}$ (cf. figure 4.9) respectivement appelés le *délai* et le *backlog* du conteneur[9]. Le premier élément de $\Theta_{\mathbf{f}}$ correspond à la distance maximale entre $\underline{f}$ et $\overline{f}$ dans le domaine temporel :

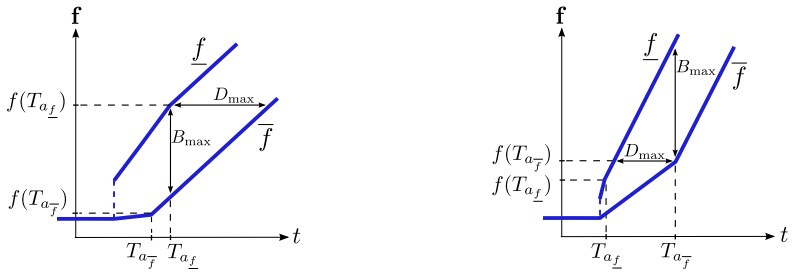
$$D_{\max} \triangleq \inf_{\tau \geq 0}\{\tau \mid \underline{f}(t_0) \leq \overline{f}(t_0+\tau)\} \quad \text{avec} \quad t_0 = \begin{cases} T_{a_{\underline{f}}} & \text{si } \underline{f}(T_{a_{\underline{f}}}) > \overline{f}(T_{a_{\overline{f}}}), \\ T_{a_{\overline{f}}} & \text{sinon.} \end{cases}$$

Tandis que le second élément de $\Theta_{\mathbf{f}}$ correspond à la distance maximale entre $\underline{f}$ et $\overline{f}$ dans le domaine événementiel :

$$B_{\max} \triangleq \underline{f}(t_0) - \overline{f}(t_0) \quad \text{avec} \quad t_0 = \max\{T_{a_{\underline{f}}}, T_{a_{\overline{f}}}\}.$$

Ces calculs, issus de [Bouillard 2007, Lem. 3 et 4], sont assez simples en raison des caractéristiques de convexité des fonctions $\underline{f}$ et $\overline{f}$. En outre, les temps $T_{a_{\underline{f}}}$ et $T_{a_{\overline{f}}}$ sont les temps à partir desquels les fonctions $\underline{f}$ et $\overline{f}$ sont ultimement affines (cf. définition 4.6).

9. Ces désignations viennent du Network Calculus (cf. définition 3.63 et remarque 3.66).



(a) Lorsque $\underline{f}(T_{a_{\underline{f}}}) > \overline{f}(T_{a_{\overline{f}}})$ et $\max\{T_{a_{\underline{f}}}, T_{a_{\overline{f}}}\} = T_{a_{\underline{f}}}$

(b) Lorsque $\underline{f}(T_{a_{\underline{f}}}) \leq \overline{f}(T_{a_{\overline{f}}})$ et $\max\{T_{a_{\underline{f}}}, T_{a_{\overline{f}}}\} = T_{a_{\overline{f}}}$

Figure 4.9 Incertitude maximale $\Theta_{\mathbf{f}} = \{ D_{\max} , B_{\max} \}$ d'un conteneur $\mathbf{f} \in \mathbf{F}$.

3 Opérations sur les conteneurs

A présent, considérons les opérations entre les conteneurs de $\mathbf{F}$ définis en 4.21. Ces opérations entre deux éléments $\mathbf{f}$ et $\mathbf{g} \in \mathbf{F}$, sont les opérations $\circ \in \{\oplus, *, ^\star\}$ monotones croissantes, et définies dans la section 4.1 par :

$$[\ \underline{f} \circ \underline{g}\ ,\ \overline{f} \circ \overline{g}\ ].$$

Malheureusement, il est possible que les bornes $\underline{f} \circ \underline{g}$ et $\overline{f} \circ \overline{g}$ n'appartiennent pas respectivement aux ensemble $\mathcal{F}_{acv}$ et $\mathcal{F}_{acx}$, et donc que cet intervalle résultat ne soit pas nécessairement un conteneur de $\mathbf{F}$. Il est donc nécessaire d'introduire des fonctions d'inclusion, à l'instar de l'approche ensembliste présentée dans [Moore 1979, Jaulin 2001], qui soient des opérations internes à $\mathbf{F}$.

Définition 4.27 (Fonctions d'inclusion des opérateurs $\{\oplus, *, ^\star\}$). Soient $\mathbf{f} = [\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}}$ et $\mathbf{g} = [\ \underline{g}\ ,\ \overline{g}\ ]_{\mathcal{L}}$ deux conteneurs appartenant à $\mathbf{F}$ et $\circ \in \{\oplus, *, ^\star\}$ un ensemble d'opérations. Les fonctions d'inclusion de ces opérateurs pour des conteneurs de $\mathbf{F}$ notées :

$$[\circ] \in \{[\oplus], [*], {}^{[\star]}\} \quad \text{sont telles que} \quad \begin{cases} \mathbf{f}[\circ]\mathbf{g} \supset \mathbf{f} \circ \mathbf{g}, \\ \mathbf{f}[\circ]\mathbf{g} \in \mathbf{F}. \end{cases} \tag{4.8}$$

Afin d'assurer la condition (4.8) de la définition des fonctions d'inclusion, notamment pour les fonctions $[\oplus]$ et ${}^{[\star]}$, l'ensemble $\mathbf{f} \circ \mathbf{g}$ sera approximé supérieurement et inférieurement par les approximations convexe et concave $\mathcal{C}_{vx}$ et $\mathcal{C}_{cv}$.

Ces approximations sont intéressantes puisqu'elles se calculent avec une complexité linéaire, comme cela est montré ci-dessous.

En outre, selon les besoins des opérations, les formes canoniques des conteneurs $[\ \mathcal{C}_{cv}(\underline{f} \oplus \Omega(\overline{f}))\ ,\ \overline{f}\ ]_{\mathcal{L}}$ (cf. définition 4.24), seront utilisés pour les fonctions $[\oplus]$ et $[*]$, tandis que pour la fonction $^{[\star]}$, l'intervalle $[\ \underline{f} \oplus \Omega(\overline{f})\ ,\ \overline{f}\ ]$ sera préféré. Ce choix est pertinent afin d'obtenir dans tous les cas une complexité algorithmique faible, soit par exemple une complexité linéaire ou quasi-linéaire.

3.1 Approximations convexe $\mathcal{C}_{vx}$ et concave $\mathcal{C}_{cv}$

La complexité de l'algorithme permettant d'obtenir les approximations convexe et concave d'une fonction $f \in \mathcal{F}_{aa}$ est donnée dans la proposition suivante.

Proposition 4.28 (Complexité algorithmique de $\mathcal{C}_{vx}$ et $\mathcal{C}_{cv}$). Soit N_f (respectivement N_g) le nombre de points de $f \in \mathcal{F}_{aa}$ (respectivement $g \in \mathcal{F}_{aa}$). Le calcul de $\mathcal{C}_{vx}(f)$ (respectivement $\mathcal{C}_{cv}(g)$) est de complexité linéaire, soit en $\mathcal{O}(N_f)$ (respectivement $\mathcal{O}(N_g)$).

Preuve. Les approximations convexe et concave reposent sur des algorithmes connus de calcul d'enveloppe convexe en géométrie algorithmique (cf. [Graham 1972]). Dans le pire des cas, seulement deux parcours de la liste des points sont requis. □

Remarque 4.29 Selon les définitions 4.19 et 4.15 et les propriétés 4.20 et 4.16, rappelons que :

- $\mathcal{C}_{cv}(f) \in \mathcal{F}_{acv}$, $\mathcal{C}_{cv}(f) \preccurlyeq f$, $\sigma(\mathcal{C}_{cv}(f)) = \sigma(f)$ et $\Delta^{\kappa_{\mathcal{C}_{cv}(f)}}_{\tau_{\mathcal{C}_{cv}(f)}} = \Delta^{\kappa_f}_{\tau_f}$,
- $\mathcal{C}_{vx}(f) \in \mathcal{F}_{acx}$, $\mathcal{C}_{vx}(f) \succcurlyeq f$, $\sigma(\mathcal{C}_{vx}(f)) = \sigma(f)$ et $\Delta^{\kappa_{\mathcal{C}_{vx}(f)}}_{\tau_{\mathcal{C}_{vx}(f)}} = \Delta^{\kappa_f}_{\tau_f}$,
- $\mathcal{C}_{cv}$ et $\mathcal{C}_{vx}$ sont des projecteurs.

En outre, ces projections s'appliquent aussi bien à des fonctions de $\mathcal{F}_{cp}$ qu'à des fonctions de $\mathcal{F}_{aa}$.

3.2 Fonction d'inclusion de la somme : $[\oplus]$

Avant d'étudier la fonction d'inclusion $[\oplus]$, rappelons que puisque l'opérateur $\oplus$ est isotone, alors $f \preccurlyeq g \Rightarrow a \oplus f \preccurlyeq a \oplus g$ et ainsi $\underline{f} \oplus \underline{g} \preccurlyeq \overline{f} \oplus \overline{g}$.

Proposition 4.30 (Fonction d'inclusion $[\oplus]$). Soient $\mathbf{f} = [\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}}$ et $\mathbf{g} = [\ \underline{g}\ ,\ \overline{g}\ ]_{\mathcal{L}}$ deux conteneurs de $\mathbf{F}$ donnés sous forme canonique. L'opération notée $[\oplus]$ et définie par :

$$\mathbf{f}[\oplus]\mathbf{g} = [\ \underline{\mathbf{f}[\oplus]\mathbf{g}}\ ,\ \overline{\mathbf{f}[\oplus]\mathbf{g}}\ ]_{\mathcal{L}} \triangleq [\ \mathcal{C}_{cv}(\underline{f} \oplus \underline{g})\ ,\ \mathcal{C}_{vx}(\overline{f} \oplus \overline{g})\ ]_{\mathcal{L}},$$

est une fonction d'inclusion pour la somme $\oplus$, *i.e.* :

$$\forall f \in [\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}} \text{ et } \forall g \in [\ \underline{g}\ ,\ \overline{g}\ ]_{\mathcal{L}} \quad \Rightarrow \quad \begin{cases} f \oplus g \in \mathbf{f}[\oplus]\mathbf{g}, \\ \mathbf{f}[\oplus]\mathbf{g} \in \mathbf{F}. \end{cases}$$

Preuve. D'après la remarque 4.29 et puisque $\oplus$ est isotone, $\forall f \in [\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}}$ et $\forall g \in [\ \underline{g}\ ,\ \overline{g}\ ]_{\mathcal{L}}$, $\mathcal{C}_{cv}(\underline{f} \oplus \underline{g}) \preccurlyeq f \oplus g \preccurlyeq \mathcal{C}_{vx}(\overline{f} \oplus \overline{g})$, et les projections $\mathcal{C}_{cv}(\underline{f} \oplus \underline{g})$ et $\mathcal{C}_{vx}(\overline{f} \oplus \overline{g})$ appartiennent respectivement aux ensembles $\mathcal{F}_{\text{acv}}$ et $\mathcal{F}_{\text{acx}}$. Ensuite, puisque $\sigma(\underline{f}) = \sigma(\overline{f})$ et $\sigma(\underline{g}) = \sigma(\overline{g})$, que la somme $\oplus$ est l'opération minimum, et que $\sigma(\mathcal{C}_{cv}(\underline{f} \oplus \underline{g})) = \sigma(\underline{f} \oplus \underline{g})$ (idem pour $\overline{f}$, $\overline{g}$ et $\mathcal{C}_{vx}$), alors $\sigma(\mathcal{C}_{cv}(\underline{f} \oplus \underline{g})) = \min(\sigma(\underline{f}), \sigma(\underline{g})) = \min(\sigma(\overline{f}), \sigma(\overline{g})) = \sigma(\mathcal{C}_{vx}(\overline{f} \oplus \overline{g}))$.

□

Cette fonction d'inclusion $[\oplus]$ ne fournit pas nécessairement un résultat sous forme canonique. Il faut pour cela calculer l'approximation concave de $(\underline{\mathbf{f}[\oplus]\mathbf{g}} \oplus \Omega(\overline{\mathbf{f}[\oplus]\mathbf{g}}))$ et obtenir ainsi le conteneur suivant :

$$[\ \mathcal{C}_{cv}(\underline{\mathbf{f}[\oplus]\mathbf{g}} \oplus \Omega(\overline{\mathbf{f}[\oplus]\mathbf{g}}))\ ,\ \overline{\mathbf{f}[\oplus]\mathbf{g}}\ ]_{\mathcal{L}} = [\ \mathcal{C}_{cv}(\mathcal{C}_{cv}(\underline{f} \oplus \underline{g}) \oplus \Omega(\mathcal{C}_{vx}(\overline{f} \oplus \overline{g})))\ ,\ \mathcal{C}_{vx}(\overline{f} \oplus \overline{g})\ ]_{\mathcal{L}}.$$

Proposition 4.31 (Complexité algorithmique de $[\oplus]$). Soient $N_{\underline{f}}, N_{\overline{f}}, N_{\underline{g}}$ et $N_{\overline{g}}$ les nombres des points extrémaux de respectivement $\underline{f}$, $\overline{f}$, $\underline{g}$ et $\overline{g}$. Le calcul de $\mathbf{f}[\oplus]\mathbf{g}$ est de complexité linéaire, soit en $\mathcal{O}(N_{\underline{f}} + N_{\overline{f}} + N_{\underline{g}} + N_{\overline{g}})$.

Preuve. Le minimum de deux fonctions ultimement affine est de complexité linéaire puisque cela requiert dans le pire des cas un seul parcours des points extrémaux de chaque fonction. En outre, selon la proposition 4.28, la complexité des approximations convexe et concave est linéaire.

□

3.3 Fonction d'inclusion de l'inf-convolution : $[*]$

Tout comme la somme $\oplus$, l'opération d'inf-convolution $*$ est isotone, donc $\underline{f} * \underline{g} \preccurlyeq \overline{f} * \overline{g}$. De plus, l'opération $*$ est close pour les ensembles $\mathcal{F}_{acv}$ et $\mathcal{F}_{acx}$ comme le montrent les propositions ci-dessous.

Proposition 4.32 (Clôture de l'inf-convolution dans $\mathcal{F}_{acv}$). Si $\underline{f}$ et $\underline{g} \in \mathcal{F}_{acv}$, alors $\underline{f} * \underline{g} \in \mathcal{F}_{acv}$.

Preuve. Grâce à la proposition 4.12 permettant la factorisation des fonctions de $\mathcal{F}_{acv}$:

$$\begin{aligned}
\underline{f} * \underline{g} &= (\Delta^{\kappa_{\underline{f}}}_{\tau_{\underline{f}}} * \Gamma_{\underline{f}}) \ * \ (\Delta^{\kappa_{\underline{g}}}_{\tau_{\underline{g}}} * \Gamma_{\underline{g}}) && \text{cf. équation (4.1)},\\
&= (\Delta^{\kappa_{\underline{f}}}_{\tau_{\underline{f}}} * \Delta^{\kappa_{\underline{g}}}_{\tau_{\underline{g}}}) \ * \ (\Gamma_{\underline{f}} * \Gamma_{\underline{g}}) && \text{cf. remarque 3.7},\\
&= (\Delta^{\kappa_{\underline{f}}+\kappa_{\underline{g}}}_{\tau_{\underline{f}}+\tau_{\underline{g}}}) \ * \ (\Gamma_{\underline{f}} \oplus \Gamma_{\underline{g}}) && \text{cf. équation (4.2)}, \qquad (4.9)\\
&= \Delta^{\kappa_{\underline{f}*\underline{g}}}_{\tau_{\underline{f}*\underline{g}}} * \Gamma_{\underline{f}*\underline{g}}.
\end{aligned}$$

Selon le théorème 4.13, la somme des fonctions $\Gamma_{\underline{f}}$ et $\Gamma_{\underline{g}}$ appartient à $\mathcal{F}_{acv}$. Donc, $\underline{f} * \underline{g}$ est également une fonction de $\mathcal{F}_{acv}$ avec $\Gamma_{\underline{f}*\underline{g}} = \Gamma_{\underline{f}} \oplus \Gamma_{\underline{g}}$ et $\Delta^{\kappa_{\underline{f}*\underline{g}}}_{\tau_{\underline{f}*\underline{g}}} = \Delta^{\kappa_{\underline{f}}+\kappa_{\underline{g}}}_{\tau_{\underline{f}}+\tau_{\underline{g}}}$. Remarquons qu'un résultat similaire peut être trouvé dans [Pandit 2006, Th. 3.2] et dans [Schmitt 2006, Lem. 2].

□

Proposition 4.33 (Clôture de l'inf-convolution dans $\mathcal{F}_{acx}$). Si $\overline{f}$ et $\overline{g} \in \mathcal{F}_{acx}$, alors $\overline{f} * \overline{g} \in \mathcal{F}_{acx}$.

Preuve. Selon [Le Boudec 2001, Th. 3.1.6] et [Bouillard 2008a, Th. 5], si a et b sont des fonctions convexes de $\mathcal{F}_{acx}$, alors $a * b$ est également convexe. En outre, le calcul de $a * b$ est obtenu en mettant bout-à-bout les différents morceaux linéaires de a et b, classés par pentes croissantes.

□

Proposition 4.34 (Fonction d'inclusion $[*]$). Soient $\mathbf{f} = [\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}}$ et $\mathbf{g} = [\ \underline{g}\ ,\ \overline{g}\ ]_{\mathcal{L}}$ deux conteneurs de $\mathbf{F}$ donnés sous forme canonique. L'opération notée $[*]$ et définie par :

$$\mathbf{f}[*]\mathbf{g} = [\ \underline{\mathbf{f}[*]\mathbf{g}}\ ,\ \overline{\mathbf{f}[*]\mathbf{g}}\ ]_{\mathcal{L}} \ \triangleq \ [\ \underline{f} * \underline{g}\ ,\ \overline{f} * \overline{g}\ ]_{\mathcal{L}},$$

est une fonction d'inclusion pour l'inf-convolution $*$, *i.e.* :

$$\forall f \in [\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}} \text{ et } \forall g \in [\ \underline{g}\ ,\ \overline{g}\ ]_{\mathcal{L}} \quad \Rightarrow \quad \begin{cases} f * g \ \in \ \mathbf{f}[*]\mathbf{g}, \\ \mathbf{f}[*]\mathbf{g} \ \in \ \mathbf{F}. \end{cases}$$

Preuve. Tout d'abord, grâce à l'isotonie de l'opérateur $*$, $\forall f \in [\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}}$ et $\forall g \in [\ \underline{g}\ ,\ \overline{g}\ ]_{\mathcal{L}} : \underline{f} * \underline{g} \preccurlyeq f * g \preccurlyeq \overline{f} * \overline{g}$. Puis, grâce aux propositions 4.32 et 4.33 $\underline{f} * \underline{g} \in \mathcal{F}_{acv}$ et $\overline{f} * \overline{g} \in \mathcal{F}_{acx}$. Enfin concernant les pentes asymptotiques $\sigma(\underline{f} * \underline{g})$ et $\sigma(\overline{f} * \overline{g})$:

- grâce à la proposition 4.12 et à l'équation (4.9), $\sigma(\underline{f} * \underline{g}) = \sigma(\Gamma_{\underline{f}} \oplus \Gamma_{\underline{g}})$. Ensuite, puisque $\Gamma_{\underline{f}} \oplus \Gamma_{\underline{g}} = \min(\Gamma_{\underline{f}}, \Gamma_{\underline{g}})$, que $\sigma(\Gamma_{\underline{f}}) = \sigma(\underline{f})$ et que $\sigma(\Gamma_{\underline{g}}) = \sigma(\underline{g})$, alors $\sigma(\underline{f} * \underline{g}) = \min(\sigma(\underline{f}), \sigma(\underline{g}))$,
- grâce à la preuve de la proposition 4.33, le calcul de $\overline{f} * \overline{g}$ est obtenu en mettant bout-à-bout les différents morceaux linéaires de $\overline{f}$ et $\overline{g}$. Puisque $\overline{f}$ et $\overline{g}$ sont ultimement affines, cette manipulation s'arrête lorsque la plus petite pente asymptotique entre $\overline{f}$ et $\overline{g}$ est atteinte, *i.e.* jusqu'à $\min(\sigma(\overline{f}), \sigma(\overline{g}))$.

Donc, puisque $\sigma(\underline{f}) = \sigma(\overline{f})$ et $\sigma(\underline{g}) = \sigma(\overline{g})$, alors $\sigma(\underline{f} * \underline{g}) = \min(\sigma(\underline{f}), \sigma(\underline{g})) = \min(\sigma(\overline{f}), \sigma(\overline{g})) = \sigma(\overline{f} * \overline{g})$.

□

A nouveau, cette fonction d'inclusion $[*]$ ne fournit pas nécessairement de résultat sous forme canonique. La représentation canonique de $\mathbf{f}[*]\mathbf{g}$ est donnée par :

$$[\ \mathcal{C}_{cv}(\underline{\mathbf{f}[*]\mathbf{g}} \oplus \Omega(\overline{\mathbf{f}[*]\mathbf{g}}))\ ,\ \overline{\mathbf{f}[*]\mathbf{g}}\ ]_{\mathcal{L}} = [\ \mathcal{C}_{cv}((\underline{f} * \underline{g}) \oplus \Omega(\overline{f} * \overline{g}))\ ,\ \overline{f} * \overline{g}\ ]_{\mathcal{L}}.$$

Proposition 4.35 (Complexité algorithmique de $[*]$). Soient $N_{\underline{f}}, N_{\overline{f}}, N_{\underline{g}}$ et $N_{\overline{g}}$ les nombres des points extrémaux de respectivement $\underline{f}$, $\overline{f}$, $\underline{g}$ et $\overline{g}$. Le calcul de $\mathbf{f}[*]\mathbf{g}$ est de complexité linéaire, soit en $\mathcal{O}(N_{\underline{f}} + N_{\overline{f}} + N_{\underline{g}} + N_{\overline{g}})$.

Preuve. D'après la preuve de la proposition 4.33, le calcul de $\overline{f} * \overline{g}$ est obtenu en mettant bout-à-bout les différents morceaux linéaires de $\overline{f}$ et $\overline{g}$, classés par pentes croissantes. Donc ce calcul ne requiert qu'un simple parcours des fonctions [10] et peut se faire en complexité linéaire. Concernant la borne inférieure, l'inf-convolution étant définie comme dans l'équation (4.9) par :

$$\underline{f} * \underline{g} = (\Delta_{\tau_{\underline{f}} + \tau_{\underline{g}}}^{\kappa_{\underline{f}} + \kappa_{\underline{g}}}) * (\Gamma_{\underline{f}} \oplus \Gamma_{\underline{g}}),$$

10. Le tri est supposé fait grâce à la structure de données informatique utilisée pour leurs représentations.

il suffit de réaliser le minimum des parties concaves Γ avec une complexité linéaire et un déplacement dans le plan par la fonction élémentaire $\Delta_{\tau_{\underline{f}}+\tau_{\underline{g}}}^{\kappa_{\underline{f}}+\kappa_{\underline{g}}}$ avec une complexité en $\mathcal{O}(1)$.

□

3.4 Fonction d'inclusion de l'étoile de Kleene : $^{[\star]}$

L'opération étoile de Kleene * est isotone, donc $\underline{f}^\star \preccurlyeq \overline{f}^{\,\star}$.

Proposition 4.36 (Fonction d'inclusion $^{[\star]}$). Soit $\mathbf{f} = [\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}}$ un conteneur de $\mathbf{F}$. L'opération notée $^{[\star]}$ et définie par :

$$\mathbf{f}^{[\star]} = [\ \underline{\mathbf{f}^{[\star]}}\ ,\ \overline{\mathbf{f}^{[\star]}}\ ]_{\mathcal{L}} \quad \text{avec} \quad \begin{cases} \underline{\mathbf{f}^{[\star]}} & \triangleq & \bigoplus_{i=0}^{n} \mathcal{C}_{cv}(\Delta_{t_i}^{k_i\star}) \\ & & \oplus\ \mathcal{C}_{cv}\Big(e \oplus \Delta_{\tau_{\underline{f}}}^{\kappa_{\underline{f}}} * \Big(\mathcal{C}_{cv}(\Delta_{\tau_{\underline{f}}}^{\kappa_{\underline{f}}\star}) \oplus \Gamma_{\underline{f}}\Big)\Big), \\ \overline{\mathbf{f}^{[\star]}} & \triangleq & \mathcal{C}_{vx}(\overline{f}^{\,\star}), \end{cases}$$

est une fonction d'inclusion pour l'étoile de Kleene *, *i.e.* :

$$\forall f \in [\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}} \quad \Rightarrow \quad \begin{cases} f^\star \ \in\ \mathbf{f}^{[\star]}, \\ \mathbf{f}^{[\star]} \ \in\ \mathbf{F}. \end{cases}$$

Les couples (t_i, k_i) de $\underline{\mathbf{f}^{[\star]}}$ sont les n points extrémaux de $\overline{f}$, $\Delta_{\tau_{\underline{f}}}^{\kappa_{\underline{f}}}$ est la fonction élémentaire de $\underline{f}$ et $\Gamma_{\underline{f}}$ est la partie concave de $\underline{f}$ (cf. proposition 4.12). Remarquons que le calcul de la borne inférieure $\underline{\mathbf{f}^{[\star]}}$ n'est pas immédiat. Il sera détaillé par la suite.

La preuve de cette proposition est donnée à partir des résultats préliminaires suivants : la borne supérieure $\overline{\mathbf{f}^{[\star]}}$ est décrite, suivie du calcul de $\underline{\mathbf{f}^{[\star]}}$.

Tout d'abord, puisque $\overline{\mathbf{f}^{[\star]}} = \mathcal{C}_{vx}(\overline{f}^{\,\star})$, alors $\overline{\mathbf{f}^{[\star]}} \in \mathcal{F}_{acx}$. De plus, $\forall f \in [\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}}$, $f \preccurlyeq \overline{f} \Rightarrow f^\star \preccurlyeq \overline{f}^{\,\star} \Rightarrow f^\star \preccurlyeq \mathcal{C}_{vx}(f^\star) \preccurlyeq \mathcal{C}_{vx}(\overline{f}^{\,\star})$ donc $f^\star \preccurlyeq \overline{\mathbf{f}^{[\star]}}$.

Lemme 4.37 (Pente asymptotique de $\overline{\mathbf{f}^{[\star]}}$). La pente asymptotique de $\overline{\mathbf{f}^{[\star]}}$ est donnée par :

$$\sigma(\overline{\mathbf{f}^{[\star]}}) = \min\Big(\sigma(\overline{f}), \min_{i=0}^{n}\big(\frac{k_i}{t_i}\big)\Big),$$

où les couples (t_i, k_i) sont les n points extrémaux de $\overline{f}$.

Preuve. Grâce aux caractéristiques de l'approximation convexe et de l'étoile de Kleene, la fonction $\overline{\mathbf{f}^{[\star]}} = \mathcal{C}_{vx}(\overline{f}^{\,\star})$ appartient à l'ensemble $\mathcal{F}_{acx}$ avec un seul point extrémal [11]. Donc, selon la propriété 4.8 :

$$\overline{\mathbf{f}^{[\star]}} = \left(\Delta_{\tau_{\overline{\mathbf{f}^{[\star]}}}}^{\kappa_{\overline{\mathbf{f}^{[\star]}}}}\right) * g,$$

avec $\kappa_{\overline{\mathbf{f}^{[\star]}}} = 0$ et $\tau_{\overline{\mathbf{f}^{[\star]}}} = 0$. Selon la figure 4.10, g est une demie-droite de pente asymptotique égale à $\sigma(\overline{\mathbf{f}^{[\star]}})$. Cette pente asymptotique provient soit des calculs des ratios k_i/t_i, soit de $\sigma(\overline{f})$.

□

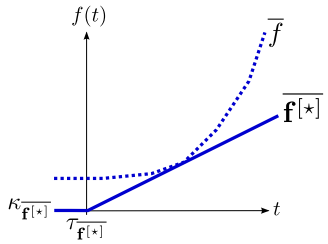


Figure 4.10 Borne supérieure $\overline{f}$ d'un conteneur $\mathbf{f} \in \mathbf{F}$ et l'approximation convexe de sa clôture sous-additive $\overline{\mathbf{f}^{[\star]}} = \mathcal{C}_{vx}(\overline{f}^{\,\star})$.

Concernant le calcul de $\underline{\mathbf{f}^{[\star]}}$, contrairement aux fonctions d'inclusion $[\oplus]$ et $[*]$, la proposition 4.36 ne propose pas de réaliser le calcul de $\mathbf{f}^{[\star]}$ avec la forme canonique du conteneur mais avec l'intervalle suivant :

$$\mathbf{f} = [\ \underline{f} \oplus \Omega(\overline{f})\ ,\ \overline{f}\ ].$$

Rappelons que $\underline{f} \oplus \Omega(\overline{f})$ est le plus petit élément du conteneur. Donc naturellement si $f \in [\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}}$, alors $f^\star \succcurlyeq (\underline{f} \oplus \Omega(\overline{f}))^\star$. A titre indicatif, le développement de $(\underline{f} \oplus \Omega(\overline{f}))^\star$ est détaillé ci-dessous :

$$\begin{aligned}
(\underline{f} \oplus \Omega(\overline{f}))^\star &= \left(\left(\Delta_{\tau_{\underline{f}}}^{\kappa_{\underline{f}}} * \Gamma_{\underline{f}}\right) \oplus \left(\bigoplus_{i=0}^{n} \Delta_{t_i}^{k_i}\right)\right)^\star && \text{cf. équations (4.1) et (4.7),}\\
&= \left(\Delta_{t_0}^{k_0} \oplus \Delta_{t_1}^{k_1} \oplus \ldots \oplus \Delta_{t_n}^{k_n} \oplus \Delta_{\tau_{\underline{f}}}^{\kappa_{\underline{f}}} * \Gamma_{\underline{f}}\right)^\star,\\
&= \left(\Delta_{t_0}^{k_0} \oplus \Delta_{t_1}^{k_1} \oplus \ldots \oplus \Delta_{t_n}^{k_n} \oplus \Delta_{\tau_{\underline{f}}}^{\kappa_{\underline{f}}} * \Gamma_{\underline{f}}^\star\right)^\star && \text{cf. équation (4.3),}\\
&= \Delta_{t_0}^{k_0\star} * \Delta_{t_1}^{k_1\star} * \ldots * \Delta_{t_n}^{k_n\star} * \left(\Delta_{\tau_{\underline{f}}}^{\kappa_{\underline{f}}} * \Gamma_{\underline{f}}^\star\right)^\star && \text{cf. équation (2.38),}\\
&= \Delta_{t_0}^{k_0\star} * \Delta_{t_1}^{k_1\star} * \ldots * \Delta_{t_n}^{k_n\star} * \left(e \oplus \Delta_{\tau_{\underline{f}}}^{\kappa_{\underline{f}}} * \Delta_{\tau_{\underline{f}}}^{\kappa_{\underline{f}}\star} * \Gamma_{\underline{f}}\right) && (4.10)\\
& && \text{cf. équations (2.36), (2.38) et (4.3).}
\end{aligned}$$

11. L'étoile de Kleene est une opération close : $(\mathcal{C}_{vx}(\overline{f}^{\,\star}))^\star = \mathcal{C}_{vx}(\overline{f}^{\,\star})$.

Cependant, ce calcul de $(\underline{f} \oplus \Omega(\overline{f}))^\star$ est composé d'inf-convolutions de fonctions ultimement pseudo-périodiques de $\mathcal{F}_{cp}$ qui sont très consommatrices de mémoire comme nous l'avons décrit dans la section 2.3.2 du chapitre 1. En effet, soient $f = \Delta_{t_0}^{k_0\star}$ et $f' = \Delta_{t_1}^{k_1\star}$ deux étoiles de fonctions élémentaires avec $\sigma(f) < \sigma(f')$. Ces fonctions sont des fonctions en escalier comme définies en 3.35. Selon le théorème A.28 de l'annexe A, puisque les fonctions ont des pentes asymptotiques différentes, l'inf-convolution de f par f' est donnée par :

$$f * f' = (e \oplus \Delta_{t_1}^{k_1} \oplus \ldots \oplus \Delta_{(K-1)t_1}^{(K-1)k_1}) * (\Delta_{t_0}^{k_0})^\star,$$

avec K un entier positif déterminant à partir de quel moment la fonction f est définitivement « au-dessus » de la fonction f'. La complexité algorithmique de ce calcul est linéaire selon la taille de K, soit en $\mathcal{O}(K)$, mais sans condition sur la grandeur de la valeur de K. La complexité algorithmique de calcul ne peut donc être contrôlée et les efforts entrepris afin de réaliser des calculs simples et efficaces deviennent inutiles en utilisant cette méthode.

Afin d'obtenir une complexité algorithmique plus intéressante, nous allons approcher ce calcul par valeur inférieure, en introduisant des approximations concaves $\mathcal{C}_{cv}$ dans l'équation (4.10) qui devient alors :

$$\mathcal{C}_{cv}(\Delta_{t_0}^{k_0\star}) * \mathcal{C}_{cv}(\Delta_{t_1}^{k_1\star}) * \ldots * \mathcal{C}_{cv}(\Delta_{t_n}^{k_n\star}) * \mathcal{C}_{cv}(e \oplus \Delta_{\tau_{\underline{f}}}^{\kappa_{\underline{f}}} * \mathcal{C}_{cv}(\Delta_{\tau_{\underline{f}}}^{\kappa_{\underline{f}}\star}) * \Gamma_{\underline{f}}).$$

D'autre part, les fonctions suivantes étant des fonctions sous-additives (cf. théorème 4.13) :

· l'approximation concave de l'étoile d'une fonction élémentaire (cf. figure 4.11, il s'agit alors d'une fonction affine simple comme définie en 3.34) :

$$\mathcal{C}_{cv}(\Delta_{t_i}^{k_i\star}),$$

· l'approximation concave d'une fonction contenant e :

$$\mathcal{C}_{cv}(e \oplus \Delta_{\tau_{\underline{f}}}^{\kappa_{\underline{f}}} * \mathcal{C}_{cv}(\Delta_{\tau_{\underline{f}}}^{\kappa_{\underline{f}}\star}) * \Gamma_{\underline{f}}),$$

· la fonction concave $\Gamma_{\underline{f}}$,

en appliquant le théorème 4.13, la définition de la borne inférieure $\underline{\mathbf{f}^{[\star]}}$ est alors donnée par :

$$\begin{aligned}\underline{\mathbf{f}^{[\star]}} &\triangleq \mathcal{C}_{cv}(\Delta_{t_0}^{k_0\star}) \oplus \mathcal{C}_{cv}(\Delta_{t_1}^{k_1\star}) \oplus \ldots \oplus \mathcal{C}_{cv}(\Delta_{t_n}^{k_n\star}) \oplus \mathcal{C}_{cv}\Big(e \oplus \Delta_{\tau_{\underline{f}}}^{\kappa_{\underline{f}}} * \Big(\mathcal{C}_{cv}(\Delta_{\tau_{\underline{f}}}^{\kappa_{\underline{f}}\star}) \oplus \Gamma_{\underline{f}}\Big)\Big), \\ &= \bigoplus_{i=0}^{n} \mathcal{C}_{cv}(\Delta_{t_i}^{k_i\star}) \oplus \mathcal{C}_{cv}\Big(e \oplus \Delta_{\tau_{\underline{f}}}^{\kappa_{\underline{f}}} * \Big(\mathcal{C}_{cv}(\Delta_{\tau_{\underline{f}}}^{\kappa_{\underline{f}}\star}) \oplus \Gamma_{\underline{f}}\Big)\Big). \qquad (4.11)\end{aligned}$$

En résumé, nous approchons l'inf-convolution des fonctions en escalier par des minimums de fonctions sous-additives et concaves.

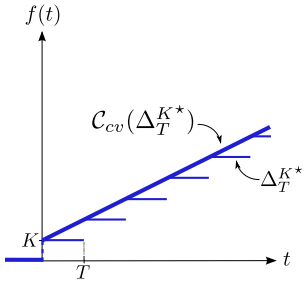


Figure 4.11 Approximation concave de l'étoile d'une fonction élémentaire : $\mathcal{C}_{cv}(\Delta_T^{K\star}) \in \mathcal{F}_{acv}$.

Enfin, puisque $\forall f \in [\ \underline{f} \oplus \Omega(\overline{f})\ ,\ \overline{f}\ ]$, alors $(\underline{f} \oplus \Omega(\overline{f})) \preccurlyeq f \Rightarrow (\underline{f} \oplus \Omega(\overline{f}))^\star \preccurlyeq f^\star$, et puisque les approximations concaves ajoutées au calcul de $(\underline{f} \oplus \Omega(\overline{f}))^\star$ approximent les fonctions par en-dessous ($\mathcal{C}_{cv}(f) \preccurlyeq f$), alors $\underline{\mathbf{f}^{[\star]}} \preccurlyeq f^\star$. La fonction $\underline{\mathbf{f}^{[\star]}}$ est composée de sommes (soit de min) de fonctions sous-additives et concaves, donc $\underline{\mathbf{f}^{[\star]}} \in \mathcal{F}_{acv}$.

Lemme 4.38 (Pente asymptotique de $\underline{\mathbf{f}^{[\star]}}$). La pente asymptotique de $\underline{\mathbf{f}^{[\star]}}$ est donnée ci-dessous :

$$\sigma(\underline{\mathbf{f}^{[\star]}}) = \sigma((\underline{f} \oplus \Omega(\overline{f}))^\star) = \min\Big(\sigma(\overline{f}), \min_{i=0}^{n}\big(\frac{k_i}{t_i}\big)\Big),$$

où les couples (t_i, k_i) sont les n points extrémaux de $\overline{f}$.

Preuve. L'étoile des fonctions élémentaires $\Delta_{t_i}^{k_i\star}$ (issus de $\overline{f}$) fournit une fonction ultimement pseudo-périodique de $\mathcal{F}_{cp}$ (cf. remarque 3.19) avec pour pente asymptotique : $\sigma(\Delta_{t_i}^{k_i\star}) = k_i/t_i$. De plus, selon le théorème A.23 donné en annexe, soient f et g deux fonctions de $\mathcal{F}_{cp}$, alors $\sigma(f * g) = \min(\sigma(f), \sigma(g))$. Et finalement, $\sigma(\Gamma_{\underline{f}}) = \sigma(\underline{f}) = \sigma(\overline{f})$ (cf. proposition 4.12 et définition 4.21). Donc, la pente asymptotique de $(\underline{f} \oplus \Omega(\overline{f}))^\star$ est donnée par :

$$\sigma((\underline{f} \oplus \Omega(\overline{f}))^\star) = \min\Big(\sigma(\overline{f}), \min_{i=0}^{n}\big(\frac{k_i}{t_i}\big)\Big).$$

□

Tous ces résultats intermédiaires permettent la démonstration de la proposition 4.36.

Preuve. (Preuve de la proposition 4.36). Tout d'abord, $\forall f \in \mathbf{f} = [\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}}$, $\underline{\mathbf{f}^{[\star]}} \preccurlyeq f^{\star} \preccurlyeq \overline{\mathbf{f}^{[\star]}}$ donc $f^{\star} \in \mathbf{f}^{[\star]}$. Ensuite, nous savons également que $\underline{\mathbf{f}^{[\star]}} \in \mathcal{F}_{acv}$ et $\overline{\mathbf{f}^{[\star]}} \in \mathcal{F}_{acx}$. Concernant les pentes asymptotiques, les lemmes 4.38 et 4.37 ont montré que $\sigma(\underline{\mathbf{f}^{[\star]}}) = \sigma(\overline{\mathbf{f}^{[\star]}})$. Donc $\mathbf{f}^{[\star]} \in \mathbf{F}$.

□

Pour finir, les complexités algorithmiques du calcul de ces bornes sont données ci-dessous.

Proposition 4.39 (Complexité algorithmique de $\overline{\mathbf{f}^{[\star]}}$). Soit $N_{\overline{f}}$ le nombre de points extrémaux de $\overline{f}$. Le calcul de $\overline{\mathbf{f}^{[\star]}}$ est de complexité linéaire, soit en $\mathcal{O}(N_{\overline{f}})$.

Preuve. Selon le lemme 4.37, le calcul de $\overline{\mathbf{f}^{[\star]}}$ requiert seulement la recherche de sa pente asymptotique en parcourant les $N_{\overline{f}}$ points extrémaux de $\overline{f}$ et la pente asymptotique $\sigma(\overline{f})$.

□

Proposition 4.40 (Complexité algorithmique de $\underline{\mathbf{f}^{[\star]}}$). Soit n le nombre de fonctions élémentaires $\Delta_{t_i}^{k_i}$ de $(\underline{f} \oplus \Omega(\overline{f}))$ et $N_{\Gamma_{\underline{f}}}$ le nombre de points extrémaux de $\Gamma_{\underline{f}}$. Le calcul de $\underline{\mathbf{f}^{[\star]}}$ est en $\mathcal{O}((n + N_{\Gamma_{\underline{f}}})log(n + N_{\Gamma_{\underline{f}}}))$.

Preuve. Afin de prouver cette proposition, l'équation (4.11) est divisée en deux parties.

– Considérons d'abord $\bigoplus_{i=0}^{n} \mathcal{C}_{cv}(\Delta_{t_i}^{k_i\star})$.

Selon la proposition 4.31, la somme de deux fonctions $f_1, f_2 \in \mathcal{F}_{acv}$ est de complexité linéaire en la taille de leurs entrées N_1 et N_2. Mais puisque le calcul de $\bigoplus_{i=0}^{n} \mathcal{C}_{cv}(\Delta_{t_i}^{k_i\star})$ requiert la somme de n fonctions de $\mathcal{F}_{acv}$, la complexité va normalement s'étendre à n^2 $(n \times (N_1 + N_2 + \ldots + N_n))$. Néanmoins, grâce à des algorithmes tels que « diviser pour régner » où la récursion est utilisée afin de diviser un problème en sous-problèmes, jusqu'à ce qu'ils deviennent assez simples pour être résolus directement, cette complexité peut être réduite à un problème quasi-linéaire. Ainsi, lorsqu'un min deux-par-deux des fonctions $\mathcal{C}_{cv}(\Delta_{t_i}^{k_i\star})$ est réalisé selon le même principe que l'algorithme du tri-fusion, cette partie de l'équation est résolue en $\mathcal{O}(nlog(n))$.

– Ensuite, pour la partie $\mathcal{C}_{cv}\Big(e \oplus \Delta_{\tau_{\underline{f}}}^{\kappa_{\underline{f}}} * \Big(\mathcal{C}_{cv}(\Delta_{\tau_{\underline{f}}}^{\kappa_{\underline{f}}\star}) \oplus \Gamma_{\underline{f}}\Big)\Big)$.

La somme de $\mathcal{C}_{cv}(\Delta_{\tau_{\underline{f}}}^{\kappa_{\underline{f}}\star})$ et $\Gamma_{\underline{f}}$ est faite en temps linéaire, soit en $\mathcal{O}(N_{\Gamma_{\underline{f}}})$, où $N_{\Gamma_{\underline{f}}}$ est le nombre de points extrémaux de $\Gamma_{\underline{f}}$. Ensuite, le déplacement dans le plan de

$\Delta^{\kappa_{\underline{f}}}_{\tau_{\underline{f}}}$ est en $\mathcal{O}(1)$, tout comme l'addition de e. L'approximation concave ne modifie pas ce résultat puisque son calcul est fait en temps linéaire (cf. proposition 4.28). Donc, la complexité de cette partie de l'équation est en $\mathcal{O}(N_{\Gamma_{\underline{f}}})$.

Ainsi, la complexité de l'équation (4.11) est en $\mathcal{O}((n + N_{\Gamma_{\underline{f}}})log(n + N_{\Gamma_{\underline{f}}}))$.

□

4 Conclusion

Nous avons présenté dans ce chapitre un conteneur $\mathbf{F}$ pour les systèmes (min,+)-linéaires (cf. définition 4.21) tel que pour $\mathbf{f} \in \mathbf{F}$:

$$\mathbf{f} = [\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}} = [\ \underline{f}\ ,\ \overline{f}\ ]\ \cap\ [\ \overline{f}\ ]_{\mathcal{L}}.$$

Ce conteneur et les fonctions d'inclusion $[\oplus]$ et $[*]$ permettent notamment d'approximer les opérations de l'algèbre $(\min, *)$ appliquées à des fonctions ultimement pseudo-périodiques (cf. définitions des approximations convexe et concave 4.15 et 4.19), tout en conservant la classe d'équivalence modulo la transformée de Legendre-Fenchel $\mathcal{L}$ (cf. définition 4.3) du résultat.

La section 3 a introduit des fonctions d'inclusion $[\circ] \in \{[\oplus], [*], {}^{[\star]}\}$ (cf. propositions 4.30, 4.34 et 4.36) pour les opérations de somme, d'inf-convolution et d'étoile de Kleene réalisées entre les conteneurs de $\mathbf{F}$. Grâce aux caractéristiques particulières des bornes des conteneurs, ces fonctions peuvent être calculées avec une complexité linéaire pour la somme et l'inf-convolution, et une complexité quasi-linéaire pour l'étoile.

Il faut en outre garder à l'esprit que dans toutes les fonctions d'inclusion proposées, la borne supérieure $\overline{f}$ est le représentant canonique de la classe d'équivalence $[\ \overline{f}\ ]_{\mathcal{L}}$ des éléments du conteneur. Il est donc possible de voir ces opérations sur les conteneurs de $\mathbf{F}$ comme des opérations sur le quotient $\mathcal{F}_{cp/\mathcal{L}}$ pour lesquelles les classes d'équivalence manipulées sont restreintes : un conteneur $[\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}}$ décrit uniquement les éléments de la classe de $[\ \overline{f}\ ]_{\mathcal{L}}$ supérieurs à $\underline{f}$.

Ainsi, à partir d'une représentation d'état telle que donnée dans (3.2) et rappelée ci-dessous avec les entrées des matrices A, B et C composées de fonctions

élémentaires :

$$\begin{cases} x = A * x \oplus B * u, \\ y = C * x, \end{cases} \tag{4.12}$$

les calculs réalisés sur des conteneurs de $\mathbf{F}$ à l'aide des fonctions d'inclusion permettent d'obtenir un encadrement du calcul $CA^{\star}B$ par le conteneur $[C[*]A^{[\star]}[*]B]_{\mathcal{L}}$. Autrement dit, la représentation d'état (4.12) est approximée par :

$$\begin{cases} x = A[*]x \ [\oplus] \ B[*]u, \\ y = C[*]x, \end{cases}$$

et la fonction de transfert $H = C * A^{\star} * B$ du système exact est alors incluse dans :

$$H \in \mathbf{H} = [C[*]A^{[\star]}[*]B]_{\mathcal{L}} = [\ \underline{C[*]A^{[\star]}[*]B}\ ,\ \overline{C[*]A^{[\star]}[*]B}\ ]_{\mathcal{L}}.$$

La borne supérieure de cet intervalle correspond à l'élément maximal de la classe d'équivalence du résultat soit :

$$\overline{C[*]A^{[\star]}[*]B} = \mathcal{C}_{vx}(CA^{\star}B) = \sup[\ CA^{\star}B\ ]_{\mathcal{L}},$$

tandis que la borne inférieure est un minorant de cette classe d'équivalence :

$$\underline{C[*]A^{[\star]}[*]B} \preccurlyeq CA^{\star}B.$$

Le conteneur et les fonctions d'inclusion présentés dans ce chapitre ont été implémentés dans une librairie de calculs appelée **ContainerMinMaxGD**. Cette librairie est décrite dans l'annexe B et quelques expérimentations sur cet outil y sont également proposées.

Pour conclure, les publications [Le Corronc 2009a] (proposant uniquement la borne supérieure du conteneur) et [Le Corronc 2009c] ont permis de diffuser ces résultats à la communauté francophone. Un autre article [Le Corronc 2011a] présentant ce conteneur de systèmes (min,+)-linéaires est actuellement en cours de soumission dans un journal international.

Chapitre 5

Commande de systèmes (min,+)-linéaires incertains

Sommaire

1 Contrôleurs existants 106
1.1 Caractérisation de la commande 106
1.2 Précompensateur 108
1.3 Retour de sortie 110
1.4 Contexte de travail 113
2 Retour de sortie : problème du Window Flow Control . . . 114
2.1 Objectif 114
2.2 Calcul de la taille de fenêtre optimale $\hat{w}$ 116
2.3 Exemple d'application 117
3 Réduction de l'incertitude grâce à un précompensateur . . 118
3.1 Objectif 118
3.2 Calcul de l'incertitude avec un précompensateur 119
3.3 Réduction de l'incertitude d'un système SISO 121
3.4 Exemple d'application 125
4 Précompensateur neutre optimal 127
4.1 Objectif 127
4.2 Dans le cas d'un système SISO 128
4.3 Extension au cas MIMO 133
4.4 Exemple d'application 134
5 Conclusion 134

Tout comme en automatique classique, la commande d'un système (min,+)-linéaire consiste à en contrôler les entrées dans le but d'atteindre un comportement et des performances spécifiés au préalable. Pour ce faire, différents correcteurs et équipements sont ajoutés aux systèmes nominaux.

Par exemple, appliqués aux systèmes manufacturiers, des problèmes dits de poursuite de modèles ont été résolus par le passé grâce à différentes structures de contrôle en boucle ouverte et en boucle fermée. Les systèmes manipulés peuvent être exacts (cf. [Cottenceau 2001] et [Maia 2003a]), ou incertains (cf. [Lhommeau 2004a]).

Dans le cadre du Network Calculus, la régulation du traffic est considérée afin d'obtenir des garanties de performances d'un réseau. Un exemple de ce contrôle de flux est donné dans [Agrawal 1999] à travers la résolution d'un problème du type boucle fermée et appelé « Window Flow Control ». Le service considéré alors est uniquement le service minimum prodigué, soit une courbe exacte, et non un intervalle de courbes.

Ainsi, ce chapitre va dans un premier temps rappeler les contrôleurs existants dans un contexte de système dont le transfert est exactement connu. Une fois ces stratégies de contrôle exposées, nous allons les appliquer aux systèmes (min,+)-linéaires incertains, *i.e.* dont le transfert h n'est pas connu exactement mais appartient à un intervalle $[\ \underline{h}\ ,\ \overline{h}\ ]$, ou lorsqu'un intervalle de service $[\ \underline{\beta}\ ,\ \overline{\beta}\ ]$ est donné. Les bornes de ces intervalles sont connues et encadrent de manière garantie le transfert h ou le service β, c'est-à-dire que $\forall u,\ \underline{h} * u \preccurlyeq y \preccurlyeq \overline{h} * u$ et $\underline{\beta} * u \preccurlyeq y \preccurlyeq \overline{\beta} * u$.

Il est important de préciser que les calculs sur les conteneurs proposés dans le chapitre 4 nous amènent directement dans ce contexte. En effet, les résultats des calculs sur les conteneurs fournissent des intervalles, contenant de manière garantie le système exact. Il est donc intéressant de développer des stratégies de contrôle dans ce contexte d'incertitude.

1 Contrôleurs existants

1.1 Caractérisation de la commande

Lors de la commande de systèmes (min,+)-linéaires, il est nécessaire de réaliser les étapes suivantes :

- définir un critère à optimiser,
- spécifier l'objectif de commande,

· définir la structure adoptée pour la réalisation de la commande.

Un des critères généralement utilisé est celui appelé le critère du **juste-à-temps**. Ce critère doit être interprété comme la quantité juste au temps voulu, *i.e.* satisfaisant le besoin, au plus tard à la date du besoin. Cela permet de garantir un service tout en minimisant l'utilisation des buffers internes. Dans le contexte des GET, il s'agit d'effectuer le nombre minimum de tirs des transitions d'entrée et de retarder au maximum le tir de ces transitions.

Concernant l'objectif de la commande, nous allons principalement nous intéresser à la technique de **poursuite de modèle** ou **modèle de référence**. Par cette méthode, le modèle de référence décrit la performance désirée du système, tandis qu'un *contrôleur*, également appelé *correcteur*, est intégré au système dit nominal afin d'en modifier le comportement et atteindre cette performance. La conception d'un tel contrôleur permet ainsi de s'approcher au mieux du fonctionnement imposé par le modèle de référence.

De manière générale, le système nominal est un système (min,+)-linéaire MIMO à p entrées et q sorties, représenté par sa matrice de transfert H, de taille $(q \times p)$ et appartenant au dioïde des fonctions ultimement pseudo-périodiques et causales $(\mathcal{F}_{cp}, \oplus, *)^{\mathsf{caus}}$ (cf. équation (3.11)). Le contrôleur C appliqué à ce système nominal est également un système (min,+)-linéaire et le transfert souhaité du système contrôlé, donc le modèle de référence, est noté G_{ref}.

Ainsi, le système H muni du correcteur C possède un transfert noté G_c et la synthèse du correcteur est effectuée dans l'objectif d'atteindre pour G_c, un transfert aussi proche que possible de G_{ref}, voire identique. Du point de vue du critère du juste-à-temps, le correcteur optimal recherché est le plus grand répondant à cet objectif. Ainsi, le correcteur optimal $\widehat{C}$ est tel que :

$$\widehat{C} \triangleq \begin{cases} G_c \preccurlyeq G_{ref}, \\ C \text{ est le plus grand possible.} \end{cases}$$

L'inégalité $G_c \preccurlyeq G_{ref}$ signifie que le système corrigé sera plus rapide que le modèle de référence. Le plus grand contrôleur est celui qui induit la plus grande commande pour atteindre cet objectif, c'est-à-dire celui qui retarde le plus les entrées dans le système. Dans un contexte manufacturier ou de réseau de communication, cela a

pour effet d'améliorer les flux internes (les stocks internes diminuent) et d'éviter les congestions inutiles (les files d'attente diminuent).

Enfin, concernant la structure adoptée pour la réalisation de la commande, deux types de contrôleurs sont utilisés par la suite. Le premier est appelé **précompensateur**, dans le sens où placé en amont du système, il va avoir une influence directe sur les entrées de celui-ci (cf. [Maia 2003a] pour les systèmes (min,+)-linéaires et [Fidler 2010] pour le Network Calculus). Le second est appelé **retour de sortie** [1] et permet de modifier les trajectoires d'entrée en fonction des sorties du système (cf. [Maia 2003a] pour les systèmes (min,+)-linéaires, [Agrawal 1999] pour le Network Calculus).

1.2 Précompensateur

1.2.1 Définition et objectif

Un précompensateur est un contrôleur noté p placé en amont d'un système h tel que :

$$u = p * v,$$

avec u le vecteur d'entrée du système nominal h, et v l'entrée du précompensateur. La sortie du système contrôlé devient alors :

$$y = h * p * v = G_p * v,$$

avec $G_p = h * p$ le transfert du système contrôlé. Cette configuration est représentée sur la figure 5.1.

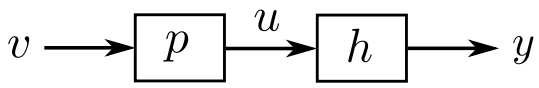


Figure 5.1 Précompensateur p placé en amont d'un système h.

Remarque 5.1 Cette stratégie de contrôle est une commande de type boucle ouverte qui ne nécessite aucune mesure des variables du système. En effet, une valeur de commande est appliquée à l'entrée du système considéré, et on suppose

1. Également appelé « feedback » en anglais.

que sa sortie réagira de façon connue en fonction de cette valeur, sans pour autant contrôler le bon déroulement de cette action.

Remarque 5.2 Avec ce type de contrôleur, la trajectoire d'entrée est a priori inconnue, mais le transfert du système est connu, ainsi que le modèle de référence décrivant le comportement entrée/sortie désiré du système contrôlé.

1.2.2 Synthèse du précompensateur

La synthèse d'un précompensateur se fait en choisissant le transfert de p tel que le système contrôlé G_p possède la dynamique décrite par le modèle de référence G_{ref}.

Afin de respecter le critère du juste-à-temps et l'objectif de poursuite du modèle de référence G_{ref}, cela revient à chercher le contrôleur optimal $\hat{p}$ tel que :

$$\hat{p} = \bigoplus \{p \mid G_p \preccurlyeq G_{ref}\} \quad \Leftrightarrow \quad \hat{p} = \bigoplus \{p \mid h * p \preccurlyeq G_{ref}\}. \tag{5.1}$$

La solution de cette expression est immédiate puisque l'inf-convolution est une application résiduable (cf. proposition 3.29) :

$$\hat{p} = h \backslash G_{ref}. \tag{5.2}$$

Remarque 5.3 Le précompensateur optimal $\hat{p} = h \backslash G_{ref} \in (\mathcal{F}_{cp}, \oplus, *)$ n'est pas toujours une fonction causale. Mais la projection causale de $\hat{p} = h \backslash G_{ref}$ dans le dioïde $(\mathcal{F}_{cp}, \oplus, *)^{\mathsf{caus}}$ (cf. définition 3.23) est également solution de $h * p \preccurlyeq G_{ref}$ et constitue le plus grand précompensateur causal.

1.2.3 Précompensateur neutre

Un modèle de référence intéressant du point de vue pratique est donné par :

$$G_{ref} = h.$$

Le précompensateur optimal, est alors donné comme suit (cf. équations (5.1) et (5.2)) :

$$\hat{p} = \bigoplus \{p \mid h * p \preccurlyeq h\} = h \backslash h. \tag{5.3}$$

Ce contrôleur optimal est appelé *neutre* puisqu'il laisse le comportement entrée/sortie inchangé. Néanmoins, il ralentit les entrées du système u autant

que possible afin d'éviter des accumulations inutiles dans h. Ainsi, il est toujours possible de filtrer le flux d'entrée d'un système h par un précompensateur sans dégrader les performances initiales.

1.2.4 Dans le cadre du Network Calculus

Lors d'une modélisation par la théorie du Network Calculus, un précompensateur p est un équipement placé en amont d'un élément de réseau S dont le flux d'entrée u est contraint par une courbe d'arrivée $\alpha^\star$ et le service par un intervalle de service $[\ \underline{\beta}\ ,\ \overline{\beta}\ ]$.

Selon [Fidler 2010], le précompensateur permettant d'assurer la contrainte $\alpha^\star$ à l'entrée u est celui dont la courbe de service maximum $\underline{\beta}_p$ est égale à cette courbe d'arrivée, soit :

$$\underline{\beta}_p = \alpha^\star.$$

La sortie du système est donc encadrée par :

$$\underline{\beta} * \alpha^\star * v \preccurlyeq y \preccurlyeq \overline{\beta} * \alpha^\star * v,$$

avec v l'entrée du système contrôlé. Cette configuration est représentée figure 5.2.

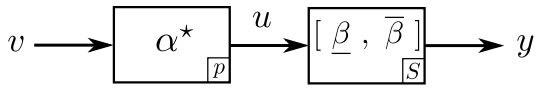


Figure 5.2 Précompensateur p placé en amont d'un réseau S et dont la courbe de service est $\underline{\beta}_p = \alpha^\star$.

A nouveau, le précompensateur agit comme un filtre sur les entrées en empêchant de congestionner le réseau avec des rafales de paquets.

1.3 Retour de sortie

1.3.1 Définition et objectifs

Un contrôleur de type retour de sortie est un contrôleur placé entre la sortie et l'entrée du système h tel que :

$$\begin{cases} u &= v \oplus fy, \\ y &= hu, \end{cases}$$

avec u le vecteur d'entrée du système nominal h, y la sortie de h et v l'entrée du système contrôlé. Cette configuration est représentée figure 5.3.

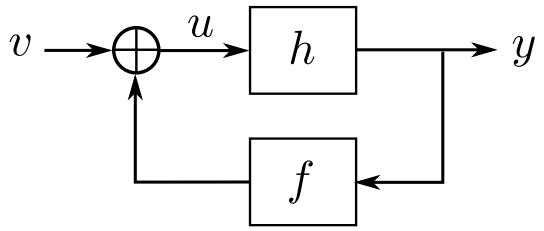


Figure 5.3 Contrôleur f placé en retour de sortie d'un système h.

La sortie y du système devient alors :

$$y = h(v \oplus fy).$$

En accord avec le théorème 2.68 de l'étoile de Kleene, la plus petite solution (soit celle qui correspond au fonctionnement au plus tôt) de cette équation implicite est donnée par :

$$y = h * (f * h)^\star * v = G_F * v,$$

avec $G_f = h(fh)^\star$ le transfert du système bouclé.

Remarque 5.4 Ce type de contrôle en boucle fermée doit prélever l'information en sortie du système afin de piloter au mieux le système. Cela implique de connaître à tout instant la valeur de la sortie y du système.

1.3.2 Synthèse du retour de sortie

A nouveau, la synthèse d'un contrôleur du type retour de sortie se fait en choisissant le transfert de f tel que le système bouclé G_f possède la dynamique décrite par le modèle de référence G_{ref}.

Afin de respecter le critère du juste-à-temps et l'objectif de poursuite du modèle de référence G_{ref}, cela revient à chercher le contrôleur optimal $\hat{f}$ tel que :

$$\hat{f} = \bigoplus\{f \mid G_f \preccurlyeq G_{ref}\} \quad \Leftrightarrow \quad \hat{f} = \bigoplus\{f \mid h(fh)^\star \preccurlyeq G_{ref}\}. \tag{5.4}$$

Proposition 5.5 ([Cottenceau 2001, Prop. 29]). Ce problème n'admet pas nécessairement de solution optimale. En revanche, il est suffisant que $G_{ref} = a^\star h$ ou $G_{ref} = hb^\star$ pour que la solution optimale soit :

$$\hat{f} = h \backslash G_{ref} \not\phi h. \tag{5.5}$$

Preuve. Selon l'équation du contrôleur optimal (5.4) et l'équivalence donnée en (2.11) :

$$h(fh)^\star \preccurlyeq G_{ref} \quad \Leftrightarrow \quad (fh)^\star \preccurlyeq h \backslash G_{ref}.$$

Or si $G_{ref} = a^\star h$, alors :

$$\begin{aligned} h \backslash G_{ref} &= h \backslash (a^\star h), \\ &= h \backslash (a^\star \backslash (a^\star h)) && \text{cf. équation (2.41)}, \\ &= (a^\star h) \backslash (a^\star h) && \text{cf. équation (2.24)}, \\ &= ((a^\star h) \backslash (a^\star h))^\star && \text{cf. équation (2.39)}. \end{aligned}$$

Donc, $h \backslash G_{ref} = (h \backslash G_{ref})^\star$ (*i.e.* $h \backslash G_{ref}$ est une étoile) et ainsi :

$$\begin{aligned} (fh)^\star \preccurlyeq (h \backslash G_{ref})^\star \quad &\Leftrightarrow \quad fh \preccurlyeq h \backslash G_{ref} && \text{cf. équation (2.43) puis (2.39)}, \\ &\Leftrightarrow \quad f \preccurlyeq h \backslash G_{ref} \not\phi h && \text{cf. équation (2.12)}. \end{aligned}$$

De la même manière lorsque $G_{ref} = hb^\star$, une solution optimale est trouvée telle que $(hf)^\star h \preccurlyeq G_{ref}$, puisque $h(fh)^\star = (hf)^\star h$ (cf. équation (2.33)). □

Remarque 5.6 Dans le cas scalaire pour un dioïde commutatif, les opérateurs $\not\phi$ et $\backslash$ sont commutatifs, donc $a \not\phi b = b \backslash a$. Mais cela n'est plus vrai dans le cas matriciel.

1.3.3 Retour de sortie neutre

Il est également possible de synthétiser un retour de sortie neutre. Le modèle de référence est donné par :

$$G_{ref} = h,$$

modèle pour lequel l'équation (5.4) possède une solution optimale qui s'écrit :

$$\hat{f} = \bigoplus \{f \mid h(fh)^\star \preccurlyeq G_{ref}\} = h \backslash h \not\phi h.$$

Dans ce cas, $G_{ref} = a^\star h$ avec [2] $a = e$.

2. En effet, $e^\star = e$ et $e * h = e$.

1.3.4 Window Flow Control

Dans le cadre du Network Calculus, un correcteur f de type retour de sortie trouve une application dans le cas connu sous le nom anglais Window Flow Control. L'objectif est de limiter la quantité de données admise dans un réseau S afin que la quantité totale de données en transit soit toujours plus petite qu'un nombre donné. Ce principe est par exemple utilisé pour des réseaux de type TCP/IP (cf. [Agrawal 1999]).

Le contrôleur f est alors considéré comme une fenêtre de taille w strictement positive, représentée par la fonction Δ_0^w et limitant les entrées dans le réseau. Lorsque le réseau est contraint par une courbe de service minimale $\overline{\beta}$, la fenêtre optimale $\hat{w}$, soit la plus petite selon le critère du juste-à-temps, se calcule comme suit (cf. équations (5.4) et (5.5)) :

$$\begin{aligned} \Delta_0^{\hat{w}} &= \bigoplus\{\Delta_0^w \mid \overline{\beta}(\Delta_0^w\overline{\beta})^\star \preccurlyeq G_{ref}\}, \\ &= \bigoplus\{\Delta_0^w \mid \Delta_0^w \preccurlyeq \overline{\beta}\backslash G_{ref}\not\phi\overline{\beta}\}, \end{aligned} \tag{5.6}$$

avec $G_{ref} = a^\star\overline{\beta}$ ou $G_{ref} = \overline{\beta}b^\star$. Donc :

$$\hat{w} = (\overline{\beta}\backslash G_{ref}\not\phi\overline{\beta})(0).$$

Remarque 5.7 Notons que pour $G_{ref} = \overline{\beta}$, la fenêtre optimale est donnée par $(\overline{\beta}\backslash\overline{\beta}\not\phi\overline{\beta})(0)$. Une autre écriture de cette expression est donnée dans [Chang 2000, Eq. (2.82)] et reprise ci-dessous :

$$\hat{w} = w_{\min} = \sup_{t\geq 0}\sup_{s\geq 0}\{\overline{\beta}(t+s) - \overline{\beta}(s) - \overline{\beta}(t)\}.$$

L'illustration de la fenêtre optimale $\hat{w}$ trouvée pour $(\overline{\beta}\backslash\overline{\beta}\not\phi\overline{\beta})(0)$ est donnée par la figure 5.4.

Remarque 5.8 Ce problème du Window Flow Control a été traité plusieurs fois dans la littérature. Cependant, la modélisation introduite dans [Le Boudec 2001] n'a pas le même modèle que celle de [Chang 2000]. Nous adoptons ici la représentation de Chang qui est homogène à celle introduite dans [Cottenceau 2001].

1.4 Contexte de travail

Usuellement, ces contrôleurs de type précompensateur et retour de sortie sont donnés pour des systèmes (min,+)-linéaires dont le transfert h est parfaitement

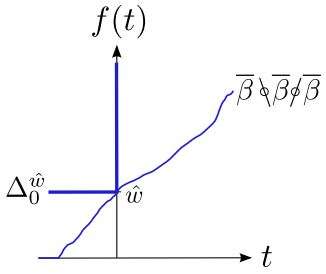


Figure 5.4 Fonction $\overline{\beta} \oslash \overline{\beta} \otimes \overline{\beta}$ dont la valeur en 0 est la taille de la fenêtre optimale $\hat{w}$.

connu. Or la commande présentée dans les sections suivantes se place dans un contexte d'un système modélisé avec des paramètres incertains. Plus précisément, du fait d'incertitudes, h n'est pas connu exactement mais appartient à un intervalle $[\ \underline{h}\ ,\ \overline{h}\ ]$ dont les bornes $\underline{h}$ et $\overline{h} \in (\mathcal{F}_{cp}, \oplus, *)$ sont connues. Les bornes inférieure et supérieure de l'intervalle sont alors des bornes sur l'incertitude du système.

Cette configuration est également valable dans le cadre du Network Calculus puisqu'un intervalle de courbes de service $[\ \underline{\beta}\ ,\ \overline{\beta}\ ]$ est utilisé. Le service fourni par le réseau peut varier entre $\underline{\beta}$ et $\overline{\beta}$.

En outre, dans la section 4, une restriction supplémentaire est faite telle que le système h est un système (min,+)-linéaire. Formellement, $\forall u$ et $\forall h \in [\ \underline{h}\ ,\ \overline{h}\ ]$, $y = h * u$.

2 Retour de sortie : problème du Window Flow Control

2.1 Objectif

Un premier problème de régulation de traffic et de garantie de performances est celui du Window Flow Control, où la taille de fenêtre optimale est calculée. Ce problème est étudié ici avec deux configurations associées.

Tout d'abord, le service fourni par le réseau est inclus dans un intervalle $[\ \underline{\beta}\ ,\ \overline{\beta}\ ]$. Ainsi, en supposant que les courbes de service minimum et maximum sont connues, la taille de la fenêtre peut être calculée aussi bien pour le pire cas du traffic que pour le meilleur, sans endommager le service fourni.

Ensuite, une différence est faite entre le flux d'entrée (de la source à la destination) et le flux retour d'acquittement [3] (de la destination à la source). En effet, ces acquittements requièrent beaucoup moins de bande passante que les données elles-mêmes (cf. [Agrawal 1999]). Ainsi, le calcul de la taille de la fenêtre aura les bénéfices de ce profit de bande passante.

Cette configuration est illustrée figure 5.5. Le flux de données est représenté par S_1, avec $[\ \underline{\beta}_1\ ,\ \overline{\beta}_1\ ]$ son intervalle de service ($\underline{\beta}_1 \preccurlyeq \overline{\beta}_1$). Le flux de retour est quant à lui représenté par S_2, avec $[\ \underline{\beta}_2\ ,\ \overline{\beta}_2\ ]$ son intervalle de service ($\underline{\beta}_2 \preccurlyeq \overline{\beta}_2$). La taille w de la fenêtre de ce contrôleur est représentée par la fonction Δ_0^w.

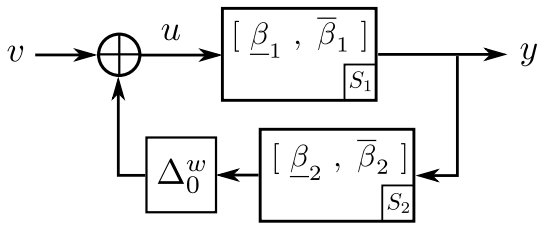


Figure 5.5 Configuration d'un Window Flow Control.

D'après cette configuration, le service global du système contrôlé est également compris dans un intervalle.

Proposition 5.9 (Service du système contrôlé G_{wfc}). Le service du système contrôlé noté G_{wfc} et tel que représenté par la figure 5.5, est inclus dans l'intervalle suivant :

$$[\ \underline{\beta}_1(\Delta_0^w\underline{\beta}_2\underline{\beta}_1)^\star\ ,\ \overline{\beta}_1(\Delta_0^w\overline{\beta}_2\overline{\beta}_1)^\star\ ].$$

Preuve. Le flux de sortie y est décrit par l'équation suivante (cf. équation (3.23)) :

$$\underline{\beta}_1 u \ \preccurlyeq\ y\ \preccurlyeq\ \overline{\beta}_1 u,$$

tandis que le flux intermédiaire u est inclus dans :

$$\begin{array}{rcl} v \oplus \Delta_0^w\underline{\beta}_2\underline{\beta}_1 u & \preccurlyeq\ u\ \preccurlyeq & v \oplus \Delta_0^w\overline{\beta}_2\overline{\beta}_1 u, \\ (\Delta_0^w\underline{\beta}_2\underline{\beta}_1)^\star v & \preccurlyeq\ u\ \preccurlyeq & (\Delta_0^w\overline{\beta}_2\overline{\beta}_1)^\star v \qquad \text{cf. théorème 2.68.} \end{array}$$

3. Flux connu en anglais sous le nom « acknowledgments ».

Donc :

$$\underline{\beta}_1(\Delta_0^w \underline{\beta}_2 \underline{\beta}_1)^\star v \preccurlyeq \ y \ \preccurlyeq \overline{\beta}_1(\Delta_0^w \overline{\beta}_2 \overline{\beta}_1)^\star v.$$

□

Le modèle de référence G_{ref} choisi pour ce problème est celui du système en boucle ouverte S_1, dont le service est inclus dans $[\ \underline{\beta}_1\ ,\ \overline{\beta}_1\ ]$. De ce point de vue, le flux d'acquittement S_2 ne freine en aucun cas les données circulant sur le réseau. La taille de la fenêtre optimale $\hat{w}$ est donc la plus petite permettant ce comportement et s'exprime par (cf. équation (5.6)) :

$$\Delta_0^{\hat{w}} \ = \ \bigoplus\{\Delta_0^w \mid \underline{\beta}_1(\Delta_0^w \underline{\beta}_2 \underline{\beta}_1)^\star \preccurlyeq \underline{\beta}_1 \quad \text{et} \quad \overline{\beta}_1(\Delta_0^w \overline{\beta}_2 \overline{\beta}_1)^\star \preccurlyeq \overline{\beta}_1\}, \tag{5.7}$$

avec $[\ \underline{\beta}_1(\Delta_0^w \underline{\beta}_2 \underline{\beta}_1)^\star\ ,\ \overline{\beta}_1(\Delta_0^w \overline{\beta}_2 \overline{\beta}_1)^\star\ ]$ l'intervalle de service du système en boucle fermée G_{wfc}.

Remarque 5.10 Puisque[4] $\underline{\beta}_1(\Delta_0^w \underline{\beta}_2 \underline{\beta}_1)^\star \succcurlyeq \underline{\beta}_1$ et $\overline{\beta}_1(\Delta_0^w \overline{\beta}_2 \overline{\beta}_1)^\star \succcurlyeq \overline{\beta}_1$, la solution optimale de (5.7) conduit aux égalités $\underline{\beta}_1(\Delta_0^w \underline{\beta}_2 \underline{\beta}_1)^\star = \underline{\beta}_1$ et $\overline{\beta}_1(\Delta_0^w \overline{\beta}_2 \overline{\beta}_1)^\star = \overline{\beta}_1$ et donc au comportement $[\ \underline{\beta}_1\ ,\ \overline{\beta}_1\ ]$.

2.2 Calcul de la taille de fenêtre optimale $\hat{w}$

La proposition suivante fournit le calcul de la taille de fenêtre $\hat{w}$ donnée dans l'équation (5.7).

Proposition 5.11 Afin d'obtenir le comportement du système en boucle fermée G_{wfc}, inchangé par rapport à celui en boucle ouverte S_1 (cf. équation (5.7)), la taille de la fenêtre optimale $\hat{w}$ est donnée ci-dessous :

$$\hat{w} = \Big((\underline{\beta}_1 \backslash \underline{\beta}_1 \not\backslash (\underline{\beta}_2 \underline{\beta}_1)) \wedge (\overline{\beta}_1 \backslash \overline{\beta}_1 \not\backslash (\overline{\beta}_2 \overline{\beta}_1))\Big)(0).$$

Preuve. Tout d'abord, considérons les bornes inférieures $\underline{\beta}_1(\Delta_0^w \underline{\beta}_2 \underline{\beta}_1)^\star$ et $\underline{\beta}_1$ de l'objectif donné en (5.7) :

$$\begin{aligned}
\underline{\beta}_1(\Delta_0^w \underline{\beta}_2 \underline{\beta}_1)^\star \preccurlyeq \underline{\beta}_1 \quad &\Leftrightarrow \quad (\Delta_0^w \underline{\beta}_2 \underline{\beta}_1)^\star \preccurlyeq \underline{\beta}_1 \backslash \underline{\beta}_1 \qquad \text{cf. équation (2.11)}, \\
&\Leftrightarrow \quad (\Delta_0^w \underline{\beta}_2 \underline{\beta}_1)^\star \preccurlyeq (\underline{\beta}_1 \backslash \underline{\beta}_1)^\star \quad \text{cf. équation (2.39)}, \\
&\Leftrightarrow \quad \Delta_0^w \underline{\beta}_2 \underline{\beta}_1 \preccurlyeq \underline{\beta}_1 \backslash \underline{\beta}_1 \qquad \text{cf. équation (2.43) puis (2.39)}, \\
&\Leftrightarrow \quad \Delta_0^w \preccurlyeq \underline{\beta}_1 \backslash \underline{\beta}_1 \not\backslash (\underline{\beta}_2 \underline{\beta}_1) \qquad \text{cf. équation (2.12)}.
\end{aligned} \tag{5.8}$$

4. En effet, $ax^\star = ae \oplus ax \oplus \ldots \succcurlyeq a$.

Cette preuve est la même pour les bornes supérieures et nous obtenons ainsi :

$$\overline{\beta}_1(\Delta_0^w\overline{\beta}_2\overline{\beta}_1)^\star \preccurlyeq \overline{\beta}_1 \quad \Leftrightarrow \quad \Delta_0^w \preccurlyeq \overline{\beta}_1 \oslash \overline{\beta}_1 \not\oslash (\overline{\beta}_2\overline{\beta}_1). \tag{5.9}$$

Enfin, dans le but de satisfaire à la fois les équations (5.8) et (5.9), la fonction optimale $\Delta_0^{\hat{w}}$ est donnée par :

$$\Delta_0^{\hat{w}} \preccurlyeq (\underline{\beta}_1 \oslash \underline{\beta}_1 \not\oslash (\underline{\beta}_2\underline{\beta}_1)) \wedge (\overline{\beta}_1 \oslash \overline{\beta}_1 \not\oslash (\overline{\beta}_2\overline{\beta}_1)),$$

et la taille de la fenêtre $\hat{w}$ est telle que :

$$\hat{w} = \Big((\underline{\beta}_1 \oslash \underline{\beta}_1 \not\oslash (\underline{\beta}_2\underline{\beta}_1)) \wedge (\overline{\beta}_1 \oslash \overline{\beta}_1 \not\oslash (\overline{\beta}_2\overline{\beta}_1))\Big)(0).$$

□

2.3 Exemple d'application

Afin de traiter un exemple d'application de ce problème du Window Flow Control, reprenons le schéma donné dans la figure 5.5. Tout le service fourni pour l'élément S_1 est inclus dans l'intervalle $[\ \underline{\beta}_1\ \ \overline{\beta}_1\ ]$ pour lequel les courbes de services $\underline{\beta}_1$ et $\overline{\beta}_1$ sont des courbes du type latence-pente, avec une latence de $16ms$ pour la première, $20ms$ pour la seconde et une pente de $100Mb/s$ pour les deux courbes de service illustrées figure 5.6a, soit :

$$\underline{\beta}_1(t) = 16ms + 100Mb/s \cdot t \qquad \text{et} \qquad \overline{\beta}_1(t) = 20ms + 100Mb/s \cdot t.$$

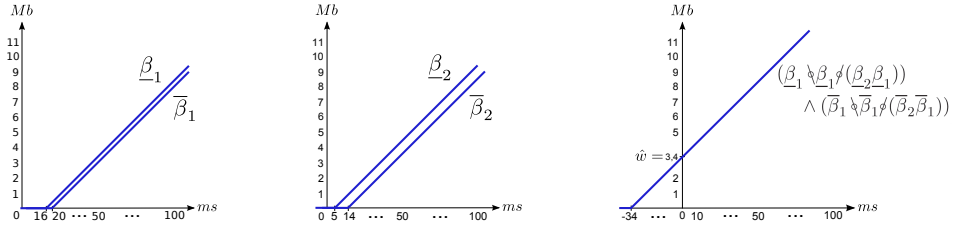
Concernant le service fourni par l'élément S_2, il est inclus dans l'intervalle $[\ \underline{\beta}_2\ \ \overline{\beta}_2\ ]$ avec des fonctions du type pente-latence pour les courbes de service $\underline{\beta}_2$ et $\overline{\beta}_2$ illustrées figure 5.6b et données par :

$$\underline{\beta}_2(t) = 5ms + 100Mb/s \cdot t \qquad \text{et} \qquad \overline{\beta}_2(t) = 14ms + 100Mb/s \cdot t.$$

En appliquant la proposition 5.11 afin de trouver la taille de fenêtre minimale $\hat{w}$ qui permet d'obtenir le même comportement en boucle fermée qu'en boucle ouverte, la fonction $\Delta_0^{\hat{w}}$ est obtenue telle que :

$$\begin{aligned} \hat{w} &= \Big((\underline{\beta}_1 \oslash \underline{\beta}_1 \not\oslash (\underline{\beta}_2\underline{\beta}_1)) \wedge (\overline{\beta}_1 \oslash \overline{\beta}_1 \not\oslash (\overline{\beta}_2\overline{\beta}_1))\Big)(0), \\ &= 3,4Mb. \end{aligned}$$

Cette courbe est représentée sur la figure 5.6c.



(a) Intervalle de service de l'élément S_1 : $[\ \underline{\beta}_1\ \ \overline{\beta}_1\ ]$

(b) Intervalle de service de l'élément S_2 : $[\ \underline{\beta}_2\ \ \overline{\beta}_2\ ]$

(c) Calcul de la taille optimale de fenêtre $\hat{w} = ((\underline{\beta}_1 \backslash \underline{\beta}_1 \not\phi (\underline{\beta}_2 \underline{\beta}_1)) \wedge (\overline{\beta}_1 \backslash \overline{\beta}_1 \not\phi (\overline{\beta}_2 \overline{\beta}_1)))(0)$

Figure 5.6 Exemple d'application du Window Flow Control.

3 Réduction de l'incertitude grâce à un précompensateur

3.1 Objectif

Lorsque la fonction de transfert h d'un système (min,+)-linéaire n'est pas exactement connue mais décrite par un intervalle $[\ \underline{h}\ ,\ \overline{h}\ ]$, alors pour toute entrée u le système a une sortie y également incluse dans un intervalle. En d'autres termes si $h \in [\ \underline{h}\ ,\ \overline{h}\ ]$, alors $\forall u$:

$$y \in [\ \underline{h}u\ ,\ \overline{h}u\ ].$$

La figure 5.7 illustre ce type de contexte.

Ainsi, pour une entrée donnée u, la sortie y est connue uniquement avec une approximation, *i.e.* une incertitude liée à la taille de l'intervalle et définie par les distances Σ_τ pour le domaine temporel, et Σ_ν pour le domaine événementiel.

Cette section met en évidence qu'en ajoutant un correcteur p en amont d'un système h incertain, le système contrôlé est également inclus dans un intervalle noté $[\ \underline{h}p\ ,\ \overline{h}p\ ]$ qui a, dans le pire cas, la même incertitude que h seul. La synthèse d'un tel précompensateur, obtenue par la recherche d'un point fixe d'une application isotone, est donnée ici dans le but de diminuer cette incertitude.

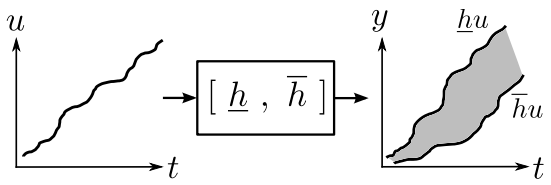


Figure 5.7 Système incertain $[\ \underline{h}\ ,\ \overline{h}\ ]$ avec une entrée exacte u et une sortie y incluse dans un intervalle $[\ \underline{h}u\ ,\ \overline{h}u\ ]$.

3.2 Calcul de l'incertitude avec un précompensateur

L'ajout d'un contrôleur de type précompensateur p ne fait pas augmenter l'incertitude en sortie du système contrôlé.

3.2.1 Incertitude maximale de la sortie

Selon le théorème 3.54, les distances maximales entre les bornes de l'intervalle $[\ \underline{h}\ ,\ \overline{h}\ ]$ peuvent être calculées facilement grâce à la corrélation $\overline{h}\backslash\underline{h}$. Ces distances temporelle et événementielle sont rappelées ici :

$$\begin{aligned}\Sigma_\tau(\underline{h},\overline{h}) &= \inf_{\tau\geq 0}\{\tau \mid (\overline{h}\backslash\underline{h})(-\tau)\leq 0\},\\ \Sigma_\nu(\underline{h},\overline{h}) &= (\overline{h}\backslash\underline{h})(0).\end{aligned}$$

La proposition suivante montre que cette corrélation permet également de trouver l'incertitude maximale de la sortie y, et ce pour n'importe quelle entrée u.

Proposition 5.12 (Incertitude maximale de la sortie y). Soit $[\ \underline{h}\ ,\ \overline{h}\ ]$ un intervalle avec $\underline{h}\preccurlyeq\overline{h}$. Les distances maximales $\Sigma_\tau(\underline{h},\overline{h})$ et $\Sigma_\nu(\underline{h},\overline{h})$ fournies par le calcul de $\overline{h}\backslash\underline{h}$, sont les distances maximales de l'intervalle $[\ \underline{h}u\ ,\ \overline{h}u\ ]$ pour n'importe quelle entrée u. Formellement :

$$\Sigma_\tau(\underline{h},\overline{h}) = \max_{\forall u\in\mathcal{D}}\{\Sigma_\tau(\underline{h}u,\overline{h}u)\} \quad\text{et}\quad \Sigma_\nu(\underline{h},\overline{h}) = \max_{\forall u\in\mathcal{D}}\{\Sigma_\nu(\underline{h}u,\overline{h}u)\}.$$

Preuve. Selon le théorème 3.54, la corrélation $(\overline{h}u)\backslash(\underline{h}u)$ représente les distances temporelle et événementielle maximales entre les bornes de l'intervalle $[\ \underline{h}u\ ,\ \overline{h}u\ ]$. Ensuite, l'inégalité suivante est montrée :

$$\begin{aligned}(\overline{h}u)\backslash(\underline{h}u) &= \overline{h}\backslash(u\backslash(\underline{h}u)) && \text{cf. équation (2.24) et } * \text{ commutatif,}\\ &\succcurlyeq \overline{h}\backslash\underline{h} && \text{cf. équation (2.17) et } * \text{ commutatif.}\end{aligned}$$

Donc, $\forall u$:

$$\Sigma_\tau(\underline{h}u, \overline{h}u) \leq \Sigma_\tau(\underline{h}, \overline{h}) \quad \text{et} \quad \Sigma_\nu(\underline{h}u, \overline{h}u) \leq \Sigma_\nu(\underline{h}, \overline{h}).$$

□

Remarque 5.13 Une entrée particulière u qui mène à l'égalité dans ces équations est donnée par $u = e = \Delta_0^0$.

Remarque 5.14 Les bornes de l'intervalle doivent avoir la même pente asymptotique et $\underline{h}(-\infty) = \overline{h}(-\infty)$, sinon les distances Σ_τ et Σ_ν seraient infinies (cf. remarque 3.58).

Exemple 5.15 Soient $\underline{h} = \Delta_0^1 \oplus \Delta_1^4 * (\Delta_1^2)^\star$ et $\overline{h} = \Delta_0^1 \oplus \Delta_3^2 \oplus \Delta_5^4 * (\Delta_1^2)^\star$ deux fonctions de $(\mathcal{F}_{cp}, \oplus, *)$ composant l'intervalle $[\ \underline{h}\ ,\ \overline{h}\ ]$, et $u = \Delta_0^0 * (\Delta_3^2)^\star$ une entrée donnée. Les calculs des corrélations $(\overline{h}u)\backslash(\underline{h}u)$ et $\overline{h}\backslash\underline{h}$ sont donnés par :

$$\begin{aligned}(\overline{h}u)\backslash(\underline{h}u) &= \mathbf{\Delta^0_{-3}} \oplus \mathbf{\Delta^1_0} * (\Delta_1^2)^\star, \\ \overline{h}\backslash\underline{h} &= \Delta^0_{-4} * (\Delta_1^2)^\star = \mathbf{\Delta^0_{-4}} \oplus \Delta^2_{-3} \oplus \Delta^4_{-2} \oplus \Delta^6_{-1} \oplus \mathbf{\Delta^8_0} * (\Delta_1^2)^\star,\end{aligned}$$

donc $(\overline{h}u)\backslash(\underline{h}u) \succcurlyeq \overline{h}\backslash\underline{h}$. Concernant les distances dans les domaines temporel et événementiel, elles sont données par les fonctions élémentaires Δ_T^K de ces corrélations pour lesquelles respectivement $K = 0$ et $T = 0$ (cf. théorème 3.54) soit :

$$\begin{aligned}\Sigma_\tau(\underline{h}u, \overline{h}u) = 3 \quad &\leq \quad \Sigma_\tau(\underline{h}, \overline{h}) = 4, \\ \Sigma_\nu(\underline{h}u, \overline{h}u) = 1 \quad &\leq \quad \Sigma_\nu(\underline{h}, \overline{h}) = 8.\end{aligned}$$

3.2.2 Ajout du précompensateur

D'après la proposition 5.12, l'incertitude maximale de la sortie y d'un système $h \in [\ \underline{h}\ ,\ \overline{h}\ ]$ peut être calculée pour toutes les entrées u grâce à la corrélation $\overline{h}\backslash\underline{h}$. Or l'utilisation d'un précompensateur (min,+)-linéaire $p \in (\mathcal{F}_{cp}, \oplus, *)$ placé en amont du système peut réduire cette incertitude, voire même la supprimer complètement.

Tout d'abord, si un précompensateur p est placé en amont d'un système incertain $[\ \underline{h}\ ,\ \overline{h}\ ]$, comme illustré dans la figure 5.8, le système contrôlé a un comportement dans l'intervalle $[\ \underline{h}p\ ,\ \overline{h}p\ ]$ et sa sortie y est ainsi incluse dans :

$$y \in [\ \underline{h}pv\ ,\ \overline{h}pv\ ].$$

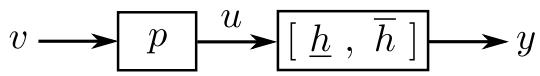


Figure 5.8 Précompensateur p placé en amont d'un système incertain $[\ \underline{h}\ ,\ \overline{h}\ ]$.

Corollaire 5.16 Soit $[\ \underline{hp}\ ,\ \overline{h}p\ ]$ un intervalle avec $\underline{hp} \preccurlyeq \overline{h}p$. Les distances maximales $\Sigma_\tau(\underline{hp}, \overline{h}p)$ et $\Sigma_\nu(\underline{hp}, \overline{h}p)$ fournies par calcul de $\overline{h}p \diagdown \underline{hp}$, sont les distances maximales de l'intervalle $[\ \underline{hpv}\ ,\ \overline{h}pv\ ]$ pour n'importe quelle entrée v. Formellement :

$$\Sigma_\tau(\underline{hp}, \overline{h}p) = \max_{\forall v \in \mathcal{D}} \{\Sigma_\tau(\underline{hpv}, \overline{h}pv)\} \quad \text{et} \quad \Sigma_\nu(\underline{hp}, \overline{h}p) = \max_{\forall v \in \mathcal{D}} \{\Sigma_\nu(\underline{hpv}, \overline{h}pv)\}.$$

Preuve. La preuve est identique à celle de la proposition 5.12.

□

Ensuite, l'incertitude maximale sur la sortie du système contrôlé $[\ \underline{hp}\ ,\ \overline{h}p\ ]$ ne peut être plus grande que celle du système seul $[\ \underline{h}\ ,\ \overline{h}\ ]$.

Proposition 5.17 (Incertitude maximale du système contrôlé). Soit $[\ \underline{hp}\ ,\ \overline{h}p\ ]$ un intervalle avec $\underline{hp} \preccurlyeq \overline{h}p$. L'incertitude sur la sortie y du système contrôlé est plus petite que l'incertitude maximale du système seul $[\ \underline{h}\ ,\ \overline{h}\ ]$:

$$\Sigma_\tau(\underline{hp}, \overline{h}p) \leq \Sigma_\tau(\underline{h}, \overline{h}) \quad \text{et} \quad \Sigma_\nu(\underline{hp}, \overline{h}p) \leq \Sigma_\nu(\underline{h}, \overline{h}).$$

Preuve. La preuve est immédiate en appliquant la proposition 5.12 pour l'intervalle $[\ \underline{hp}\ ,\ \overline{h}p\ ]$ et $\forall p$ donné.

□

Remarque 5.18 Dans le cas où $\Sigma_\tau(\underline{h}, \overline{h})$ et $\Sigma_\nu(\underline{h}, \overline{h})$ sont des distances infinies (cf. remarque 5.14), elles peuvent devenir des distances finies pour certains p en ralentissant suffisamment le taux de production du système contrôlé.

3.3 Réduction de l'incertitude d'un système SISO

Selon la proposition 5.17, puisque l'ajout d'un précompensateur p n'augmente pas l'incertitude sur la sortie y du système contrôlé, il est intéressant de chercher s'il peut faire diminuer cette incertitude en réduisant la taille de l'intervalle $[\ \underline{hp}\ ,\ \overline{h}p\ ]$.

Nous proposons donc ici le calcul d'un précompensateur p permettant d'obtenir des distances $\Sigma_\tau(\underline{h}p, \overline{h}p)$ et $\Sigma_\nu(\underline{h}p, \overline{h}p)$ plus petites qu'un τ_0 fixé pour le premier et un ν_0 fixé pour le second. Parmi les précompensateurs qui permettent de respecter ces contraintes, il est intéressant de calculer le plus grand, *i.e.* le contrôleur optimal selon le critère du juste-à-temps.

Cet objectif peut donc être décrit comme suit :

$$\begin{aligned} \hat{p} &= \bigoplus \{p \mid \Sigma_\tau(\underline{h}p, \overline{h}p) \leq \tau_0 \quad \text{et} \quad \Sigma_\nu(\underline{h}p, \overline{h}p) \leq \nu_0\}, \\ &= \bigoplus \{p \mid (\overline{h}p) \backslash\!\!\backslash (\underline{h}p) \succcurlyeq \Delta^0_{-\tau_0} \oplus \Delta^{\nu_0}_0\}, \end{aligned} \tag{5.10}$$

avec $\Delta^0_{-\tau_0} \oplus \Delta^{\nu_0}_0$ une fonction constante par morceaux finie de $\mathcal{F}_c$ et grâce aux équivalences suivantes :

$$\Sigma_\tau(\underline{h}p, \overline{h}p) \leq \tau_0 \quad \Leftrightarrow \quad (\overline{h}p) \backslash\!\!\backslash (\underline{h}p) \succcurlyeq \Delta^0_{-\tau_0}, \tag{5.11}$$

$$\Sigma_\nu(\underline{h}p, \overline{h}p) \leq \nu_0 \quad \Leftrightarrow \quad (\overline{h}p) \backslash\!\!\backslash (\underline{h}p) \succcurlyeq \Delta^{\nu_0}_0. \tag{5.12}$$

Ce problème peut être traité via l'application de la théorie des équations aux points fixes (cf. section 3.3.1 du chapitre 2) comme le montre la proposition suivante et son corollaire.

Proposition 5.19 Le plus grand précompensateur $\hat{p}$ comme défini dans l'équation (5.10) correspond au plus grand point fixe de l'application suivante :

$$p = p \wedge \overline{h} \backslash\!\!\backslash (\Delta_0^{-\nu_0} \underline{h}p) \wedge \overline{h} \backslash\!\!\backslash (\Delta^0_{\tau_0} \underline{h}p). \tag{5.13}$$

Preuve.

· Dans le domaine temporel, la contrainte est écrite comme suit :

$$\begin{aligned} & (\overline{h}p) \backslash\!\!\backslash (\underline{h}p) \succcurlyeq \Delta^0_{-\tau_0} && \text{cf. équation (5.11)}, \\ \Leftrightarrow \quad & \underline{h}p \succcurlyeq \overline{h}p\Delta^0_{-\tau_0} && \text{cf. équation (2.11)}, \\ \Leftrightarrow \quad & \Delta^0_{\tau_0}\underline{h}p \succcurlyeq \overline{h}p && \text{cf. équation (3.12)}, \\ \Leftrightarrow \quad & \overline{h} \backslash\!\!\backslash (\Delta^0_{\tau_0}\underline{h}p) \succcurlyeq p && \text{cf. équation (2.11)}, \\ \Leftrightarrow \quad & p = p \wedge \overline{h} \backslash\!\!\backslash (\Delta^0_{\tau_0}\underline{h}p) && \text{cf. definition 2.56}. \end{aligned}$$

· Dans le domaine événementiel l'objectif devient :

$$\begin{aligned}
& (\overline{h}p)\backslash(\underline{h}p) \succcurlyeq \Delta_0^{\nu_0} && \text{cf. équation (5.12),}\\
\Leftrightarrow\ & \underline{h}p \succcurlyeq \overline{h}p\Delta_0^{\nu_0} && \text{cf. équation (2.11),}\\
\Leftrightarrow\ & \Delta_0^{-\nu_0}\underline{h}p \succcurlyeq \overline{h}p && \text{cf. équation (3.13),}\\
\Leftrightarrow\ & \overline{h}\backslash(\Delta_0^{-\nu_0}\underline{h}p) \succcurlyeq p && \text{cf. équation (2.11),}\\
\Leftrightarrow\ & p = p \wedge \overline{h}\backslash(\Delta_0^{-\nu_0}\underline{h}p) && \text{cf. definition 2.56.}
\end{aligned}$$

· En considérant ces deux contraintes, le précompensateur p doit satisfaire :

$$p = p \wedge \overline{h}\backslash(\Delta_0^{-\nu_0}\underline{h}p) \wedge \overline{h}\backslash(\Delta_{\tau_0}^{0}\underline{h}p).$$

· Enfin, puisque les opérateurs $*$ et $\wedge$ sont isotones et grâce à la propriété (2.13) du quotient à gauche, la fonction $f(x) = \overline{h}\backslash(\Delta_0^{-\nu_0}\underline{h}x) \wedge \overline{h}\backslash(\Delta_{\tau_0}^{0}\underline{h}x)$ est une fonction isotone.

□

Remarque 5.20 Cette équation fait partie de l'ensemble des points post-fixes d'application (cf. définition 2.56) pour lequel $x = x \wedge f(x)$ avec $f(x) = \overline{h}\backslash(\Delta_0^{-\nu_0}\underline{h}x) \wedge \overline{h}\backslash(\Delta_{\tau_0}^{0}\underline{h}x)$ une fonction isotone.

Le théorème 2.60 présente un algorithme qui, s'il converge, fournit le plus grand point fixe d'une application isotone. Il est donc possible d'appliquer cet algorithme à l'équation (5.13) afin de trouver le plus grand précompensateur $\hat{p}$. Cependant, dans ce théorème, la valeur initiale x_0 étant l'élément $\top$, la première itération va calculer le résultat suivant, puisque $a\backslash\top = \top$ (cf. équation (2.15)) :

$$f(x_0) = f(\top) = \overline{h}\backslash(\Delta_0^{-\nu_0}\underline{h}\top) \wedge \overline{h}\backslash(\Delta_{\tau_0}^{0}\underline{h}\top) = \top.$$

Il est donc nécessaire d'invoquer le corollaire 2.62 qui va rechercher le plus grand point fixe, soit le plus grand précompensateur $\hat{p}$, plus petit qu'une valeur de départ notée val. En outre, puisque le contrôleur p est un système (min,+)-linéaire, il doit être causal (cf. théorème 3.13), *i.e.* il doit vérifier l'égalité $p = \mathsf{Pr}_{\mathsf{caus}}(p)$. Le problème donné dans l'équation (5.10) devient :

$$\hat{p} = \bigoplus\{p \mid (\overline{h}p)\backslash(\underline{h}p) \succcurlyeq \Delta_{-\tau_0}^{0} \oplus \Delta_0^{\nu_0}\ ,\ p = \mathsf{Pr}_{\mathsf{caus}}(p)\ ,\ p \preccurlyeq val\}. \tag{5.14}$$

Corollaire 5.21 Le plus grand précompensateur $\hat{p}$ comme défini dans l'équation (5.14) correspond au plus grand point fixe de l'application suivante :

$$p = p \wedge \overline{h}\backslash(\Delta_0^{-\nu_0}\underline{h}p) \wedge \overline{h}\backslash(\Delta_{\tau_0}^{0}\underline{h}p) \wedge \mathsf{Pr}_{\mathsf{caus}}(p) \wedge val,$$

avec val une fonction causale de $(\mathcal{F}_{cp}, \oplus, *)^{\mathsf{caus}}$.

Preuve. Selon la définition 3.23 de la projection causale $\mathsf{Pr}_{\mathsf{caus}}$ et pour tout p, $\mathsf{Pr}_{\mathsf{caus}}(p) \preccurlyeq p$. Afin d'obtenir $p = \mathsf{Pr}_{\mathsf{caus}}(p)$, le précompensateur p doit satisfaire :

$$\mathsf{Pr}_{\mathsf{caus}}(p) \succcurlyeq p \quad \Leftrightarrow \quad p = p \wedge \mathsf{Pr}_{\mathsf{caus}}(p).$$

□

Pour finir, il est également intéressant de considérer la cas particulier où $\nu_0 = 0$ et $\tau_0 = 0$. Cela signifie que l'objectif est d'obtenir un système contrôlé sans aucune incertitude sur la sortie. Dans ce cas, le précompensateur p doit satisfaire :

$$\underline{h}p = \overline{h}p \;\Leftrightarrow\; \begin{cases} \Sigma_\tau(\underline{h}p, \overline{h}p) \leq 0 \\ \Sigma_\nu(\underline{h}p, \overline{h}p) \leq 0 \end{cases} \quad \Leftrightarrow \quad (\overline{h}p) \backslash (\underline{h}p) \succcurlyeq \Delta_0^0.$$

et le plus grand précompensateur $\hat{p}$ est donc maintenant donné par :

$$\hat{p} = \bigoplus \{p \mid (\overline{h}p) \backslash (\underline{h}p) \succcurlyeq \Delta_0^0 \ , \ p = \mathsf{Pr}_{\mathsf{caus}}(p) \ , \ p \preccurlyeq val\}. \tag{5.15}$$

Proposition 5.22 Le plus grand précompensateur $\hat{p}$ comme défini dans l'équation (5.15) correspond au plus grand point fixe de l'application suivante :

$$p = p \wedge \overline{h} \backslash (\underline{h}p) \wedge \mathsf{Pr}_{\mathsf{caus}}(p) \wedge val. \tag{5.16}$$

Preuve. La preuve reprend celle de la proposition 5.19.

□

Remarque 5.23 (Extension au cas MIMO). Une extension possible au cas MIMO de cette réduction est proposée comme suit. Afin de garantir une incertitude temporelle notée T dans le pire cas de fonctionnement pour tous les éléments de la matrice $H \in [\ \underline{H}\ ,\ \overline{H}\ ]$, le contrôleur $\hat{P}$ optimal doit satisfaire :

$$\begin{aligned} \hat{P} &= \bigoplus \{P \mid (\overline{H}P) \backslash (\underline{H}P) \succcurlyeq \Delta_{-T}^0\}, \\ &= \bigoplus \{P \mid \Delta_T^0 * \underline{H}P \succcurlyeq \overline{H}P\}. \end{aligned}$$

Le principe est le même pour une incertitude événementielle K fixée.

3.4 Exemple d'application

Un exemple d'un précompensateur permettant de supprimer l'incertitude sur la sortie d'un système contrôlé est maintenant proposé. Afin de calculer un tel précompensateur, nous considérons le GET incertain SISO de la figure 5.9. Les délais entre crochets fournissent l'intervalle de temps dans lequel la temporisation de la place évolue.

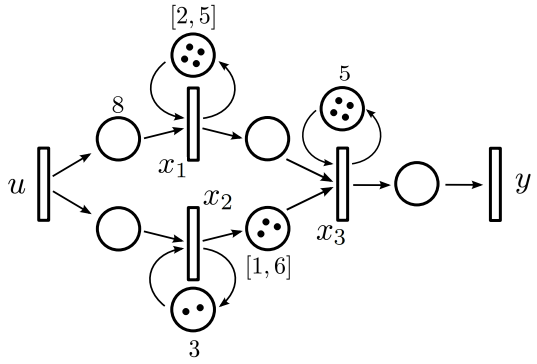


Figure 5.9 Exemple d'un GET incertain SISO.

Le système est soumis à des variations de temps, ce qui signifie que les durées des tâches ne sont pas connues exactement mais à travers des bornes inférieure et supérieure. Ainsi, le système est modélisé par l'intervalle $[\ \underline{h}\ ,\ \overline{h}\ ]$ où $\underline{h} \in (\mathcal{F}_{cp}, \oplus, *)$ représente son comportement le plus rapide (*i.e.* tous les délais minimums sont considérés), et $\overline{h} \in (\mathcal{F}_{cp}, \oplus, *)$ représente son comportement le plus lent (*i.e.* tous les délais maximums sont considérés).

Après calculs, ces comportements extrémaux sont donnés par :

$$\begin{aligned} \underline{h} &= \Delta_8^0 * (\Delta_5^4)^\star \ \oplus\ \Delta_{49}^{35} \oplus \Delta_{64}^{45} * (\Delta_6^4)^\star, \\ \overline{h} &= \Delta_8^0 * (\Delta_5^4)^\star \ \oplus\ \Delta_{24}^{15} \oplus \Delta_{39}^{25} * (\Delta_6^4)^\star. \end{aligned}$$

Selon la proposition 5.12, pour toute entrée u l'incertitude maximale de l'intervalle $[\ \underline{h}\ ,\ \overline{h}\ ]$ est :

$$\Sigma_\tau(\underline{h}, \overline{h}) = 5 \quad \text{et} \quad \Sigma_\nu(\underline{h}, \overline{h}) = 4,$$

puisque le calcul de $\overline{h}\backslash\underline{h}$ est le suivant :

$$\begin{aligned}\overline{h}\backslash\underline{h} &= \Delta^{0}_{-5} \oplus \Delta^{2}_{-2} \oplus \Delta^{4}_{1} \oplus \ldots \oplus \Delta^{49}_{62} * (\Delta^{2}_{3})^{\star},\\ &= \boldsymbol{\Delta^{0}_{-5}} \oplus \Delta^{2}_{-2} \oplus \boldsymbol{\Delta^{4}_{0}} \oplus \Delta^{4}_{1} \oplus \ldots \oplus \Delta^{49}_{62} * (\Delta^{2}_{3})^{\star}.\end{aligned}$$

Afin de trouver le plus grand précompensateur $\hat{p}$ tel que l'incertitude du système contrôlé est supprimée, soit :

$$\hat{p} = \bigoplus\{p \mid (\overline{h}p)\backslash(\underline{h}p) \succcurlyeq \Delta^{0}_{0}\ ,\ p = \mathsf{Pr}_{\mathsf{caus}}(p)\ ,\ p \preccurlyeq \underline{h}\backslash\underline{h}\},$$

l'algorithme du corollaire 2.62 est appliqué à l'équation (5.16) soit à :

$$p = p \wedge \overline{h}\backslash(\underline{h}p) \wedge \mathsf{Pr}_{\mathsf{caus}}(p) \wedge \underline{h}\backslash\underline{h}.$$

La corrélation $\underline{h}\backslash\underline{h}$ qui correspond au précompensateur neutre pour $\underline{h}$ est choisie dans cet exemple pour la fonction *val* (ce choix est dépendant des besoins pour les applications réelles). Donc, le précompensateur trouvé sera le plus grand précompensateur tel que $\hat{p} \preccurlyeq \underline{h}\backslash\underline{h}$.

Le précompensateur $\hat{p}$ obtenu en quatre itérations de l'algorithme est le suivant :

$$\hat{p} = \Delta^{4}_{1} \oplus \ldots \oplus \Delta^{49}_{62} * (\Delta^{2}_{3})^{\star}.$$

Ainsi, les bornes de l'intervalle [$\underline{h}p$, $\overline{h}p$] du système contrôlé sont identiques, l'incertitude en sortie est supprimée :

$$\underline{h}p = \overline{h}p = \Delta^{4}_{9} \oplus \ldots \oplus \Delta^{49}_{70} * (\Delta^{2}_{3})^{\star}.$$

Cette suppression se vérifie lors du calcul des distances dans les domaines événementiel et temporel :

$$\begin{aligned}\Sigma_{\tau}(\underline{h}p, \overline{h}p) = 0 &\leq \Sigma_{\tau}(\underline{h}, \overline{h}) = 5,\\ \Sigma_{\nu}(\underline{h}p, \overline{h}p) = 0 &\leq \Sigma_{\nu}(\underline{h}, \overline{h}) = 4,\end{aligned}$$

puisque (cf. figure 5.10) :

$$(\overline{h}p)\backslash(\underline{h}p) = \boldsymbol{\Delta^{0}_{0}} \oplus \Delta^{5}_{4} \oplus \Delta^{10}_{8} \oplus \ldots \oplus \Delta^{29}_{37} * (\Delta^{2}_{3})^{\star}.$$

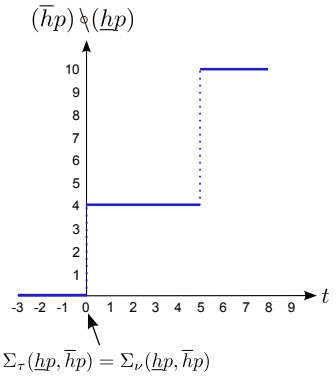


Figure 5.10 Corrélation $(\overline{h}p)\backslash(\underline{h}p)$ dont la valeur en 0 fournit les distances maximales $\Sigma_\tau(\underline{h}p,\overline{h}p) = \Sigma_\tau(\underline{h}p,\overline{h}p) = 0$, soit une incertitude nulle en sortie du système contrôlé $[\ \underline{h}p\ ,\ \overline{h}p\ ]$.

4 Précompensateur neutre optimal

4.1 Objectif

Cette section étudie le problème de recherche d'un précompensateur neutre optimal lorsque le système (min,+)-linéaire $h \in (\mathcal{F}_{cp}, \oplus, *)$ auquel il est appliqué présente des paramètres incertains et appartient à un intervalle $[\ \underline{h}\ ,\ \overline{h}\ ]$ dont seules les bornes sont connues. Plus précisément, il est supposé que le système considéré a un comportement (min,+)-linéaire, mais ce comportement est connu avec des incertitudes. Formellement, $\exists h \in [\ \underline{h}\ ,\ \overline{h}\ ]$ tel que $y = h * u$.

Remarque 5.24 Il est important de noter que notre approche est différente de celle présentée dans [Lhommeau 2005]. En effet, dans [Lhommeau 2005], le système appartient également à un intervalle ($h \in [\ \underline{h}\ ,\ \overline{h}\ ]$) mais est sujet à des fluctuations dans les limites de l'intervalle, c'est-à-dire que h n'est pas nécessairement (min,+)-linéaire. Exprimé différemment, $\nexists h \in [\ \underline{h}\ ,\ \overline{h}\ ]$ tel que $y = h*u$.

L'objectif est de trouver un précompensateur p (cf. figure 5.11) qui soit neutre pour tous les transferts h contenus dans l'intervalle $[\ \underline{h}\ ,\ \overline{h}\ ]$. Afin de respecter cette neutralité pour tout $h \in [\ \underline{h}\ ,\ \overline{h}\ ]$, le précompensateur p doit être choisi dans l'ensemble suivant :

$$\{p \mid \forall h \in [\ \underline{h}\ ,\ \overline{h}\ ],\ h * p = h\} \ = \ \{p \mid \forall h \in [\ \underline{h}\ ,\ \overline{h}\ ],\ e \preccurlyeq p \preccurlyeq h\backslash h\}. \qquad (5.17)$$

Ainsi, quel que soit h dans l'intervalle, le comportement du système contrôlé n'est donc pas modifié par l'ajout du précompensateur.

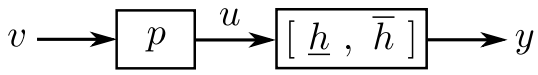


Figure 5.11 Précompensateur p placé en amont d'un système incertain $[\ \underline{h}\ ,\ \overline{h}\ ]$.

En outre, selon le critère du juste-à-temps, p doit être optimal, c'est-à-dire qu'il doit être le plus grand précompensateur respectant cette neutralité pour tout $h \in [\ \underline{h}\ ,\ \overline{h}\ ]$. D'après l'équation (5.3), cela revient à dire que le précompensateur neutre optimal $\hat{p} \in (\mathcal{F}_{cp}, \oplus, *)$ est tel que :

$$\hat{p} = \bigoplus \{p \mid \forall h \in [\ \underline{h}\ ,\ \overline{h}\ ], e \preccurlyeq p \preccurlyeq h \backslash h\} = \bigwedge_{h \in [\ \underline{h}\ ,\ \overline{h}\ ]} h \backslash h.$$

Cependant, l'autocorrélation $h \backslash h$ n'est pas une opération monotone (cf. propriété 3.56). Ce calcul ne peut donc pas se déduire directement du calcul sur les bornes $\underline{h} \backslash \underline{h}$ ou $\overline{h} \backslash \overline{h}$.

Nous montrons donc dans cette section que le précompensateur neutre optimal $\hat{p}$ est donné par l'expression suivante :

$$\hat{p} = \bigwedge_{h \in [\ \underline{h}\ ,\ \overline{h}\ ]} h \backslash h = e \oplus \overline{h} \backslash \underline{h}.$$

Cette expression est valable dans le cas SISO comme dans le cas MIMO. La preuve de ce résultat s'appuie sur une interprétation qui peut être faite de la corrélation $h \backslash h$. En effet, d'après la théorie du second ordre, (cf. théorème 3.54), le signal $(h \backslash h)(t)$ reflète l'écart maximal entre $h(\tau)$ et $h(\tau - t)$ lorsque τ varie. Compte tenu du contexte borné de $h \in [\ \underline{h}\ ,\ \overline{h}\ ]$, $(h \backslash h)(t)$ ne peut pas être supérieur à l'écart entre $\underline{h}$ et $\overline{h}$, mesuré par la corrélation $\overline{h} \backslash \underline{h}$.

4.2 Dans le cas d'un système SISO

Commençons par analyser le terme $\overline{h} \backslash \underline{h}$ par rapport à l'autocorrélation $h \backslash h$.

Lemme 5.25 Soient h, $\underline{h}$ et $\overline{h}$ trois fonctions du dioïde $(\mathcal{F}_{cp}, \oplus, *)^{\mathsf{caus}}$ avec

$\underline{h} \preccurlyeq h \preccurlyeq \overline{h}$:

$$\forall h \in [\ \underline{h}\ ,\ \overline{h}\ ],\ h \backslash h \ \succcurlyeq\ \overline{h} \backslash \underline{h} \quad \Rightarrow \quad \bigwedge_{h \in [\ \underline{h}\ ,\ \overline{h}\ ]} h \backslash h \ \succcurlyeq\ \overline{h} \backslash \underline{h}.$$

Preuve. Selon les propriétés d'isotonie et d'antitonie du quotient à gauche (cf. équation (2.13)) :

$$h \preccurlyeq \overline{h} \quad \Rightarrow \quad h \backslash h \succcurlyeq \overline{h} \backslash h.$$

En outre :

$$h \succcurlyeq \underline{h} \quad \Rightarrow \quad \overline{h} \backslash h \succcurlyeq \overline{h} \backslash \underline{h}.$$

Ainsi, $\forall h_i \in [\ \underline{h}\ ,\ \overline{h}\ ]$:

$$\left.\begin{array}{rcl} h_1 \backslash h_1 & \succcurlyeq & \overline{h} \backslash \underline{h} \\ h_2 \backslash h_2 & \succcurlyeq & \overline{h} \backslash \underline{h} \\ \cdots & & \end{array}\right\} \quad \Rightarrow \quad h_1 \backslash h_1 \wedge h_2 \backslash h_2 \wedge \ldots \succcurlyeq \overline{h} \backslash \underline{h}.$$

En étendant à l'ensemble des $h \in [\ \underline{h}\ ,\ \overline{h}\ ]$, on obtient :

$$\bigwedge_{h \in [\ \underline{h}\ ,\ \overline{h}\ ]} h \backslash h \succcurlyeq \overline{h} \backslash \underline{h}.$$

□

D'après ce lemme, $p = \overline{h} \backslash \underline{h}$ est un minorant pour l'ensemble $\{h \backslash h \mid h \in [\ \underline{h}\ ,\ \overline{h}\ ]\}$. D'un autre côté, nous savons également que $p = e$ est quant à lui neutre pour cet ensemble puisque $h * e = h$. Il s'agit notamment du plus petit neutre puisque si p est neutre pour h, alors $e \preccurlyeq p \preccurlyeq h \backslash h$ (cf. équation (5.17)).

Ainsi, le précompensateur $p = e \oplus \overline{h} \backslash \underline{h}$ est neutre pour tous les h de l'intervalle $[\ \underline{h}\ ,\ \overline{h}\ ]$. En revanche, il reste à montrer qu'il s'agit du plus *grand* neutre.

Remarque 5.26 Notons que l'existence d'une solution optimale est garantie puisque si p_1 et p_2 sont deux précompensateurs neutres $\forall h \in [\ \underline{h}\ ,\ \overline{h}\ ]$, alors $h * p_1 = h$ et $h * p_2 = h$. Ce qui implique que pour tout $h \in [\ \underline{h}\ ,\ \overline{h}\ ]$:

$$\begin{aligned} h(p_1 \oplus p_2) &= hp_1 \oplus hp_2, \\ &= h \oplus h, \\ &= h. \end{aligned}$$

Donc la somme de deux précompensateurs neutres $\forall h \in [\ \underline{h}\ ,\ \overline{h}\ ]$ reste neutre $\forall h \in [\ \underline{h}\ ,\ \overline{h}\ ]$. Par conséquent, la somme de tous les précompensateurs neutres $\forall h \in [\ \underline{h}\ ,\ \overline{h}\ ]$ est le plus grand.

Le lemme et la proposition ci-dessous démontrent l'optimalité du précompensateur $e \oplus \overline{h} \backslash\!\!\backslash \underline{h}$.

Lemme 5.27 Soient h, $\underline{h}$ et $\overline{h}$ trois fonctions du dioïde $(\mathcal{F}_{cp}, \oplus, *)^{\mathsf{caus}}$ avec $\underline{h} \preccurlyeq h \preccurlyeq \overline{h}$:

$$\forall t > 0, \quad \exists h \in [\ \underline{h}\ ,\ \overline{h}\ ] \quad \text{tel que} \quad (h \backslash\!\!\backslash h)(t) = (\overline{h} \backslash\!\!\backslash \underline{h})(t).$$

Preuve. Tout d'abord, le calcul de $\overline{h} \backslash\!\!\backslash \underline{h}$ est donné comme suit (cf. équation (3.15)) :

$$\forall t,\ (\overline{h} \backslash\!\!\backslash \underline{h})(t) = \bigwedge_{\tau \geq 0} \{\overline{h}(\tau - t) \backslash\!\!\backslash \underline{h}(\tau)\} = \sup_{\tau \geq 0} \{\underline{h}(\tau) - \overline{h}(\tau - t)\}.$$

Puisque $\underline{h}$ et $\overline{h}$ sont périodiques, il existe au moins un τ_0 (non nécessairement unique) tel que :

$$\forall t > 0,\ (\overline{h} \backslash\!\!\backslash \underline{h})(t) = \sup_{\tau \geq 0} \{\underline{h}(\tau) - \overline{h}(\tau - t)\} = \underline{h}(\tau_0) - \overline{h}(\tau_0 - t). \tag{5.18}$$

Ce τ_0 appartient à l'ensemble des points pour lesquels le maximum des écarts entre $\underline{h}$ et $\overline{h}$ est atteint (cf. équation (3.26)) :

$$\tau_0 \ \in \ \arg\max_{\tau \geq 0} \{\underline{h}(\tau) - \overline{h}(\tau - t)\}.$$

Ainsi, d'après la figure 5.12 représentant les bornes $\underline{h}$ et $\overline{h}$, lorsque $t = 1$: $\tau_0 \in \{3, 4, 6, 8, 10, \ldots\}$. En prenant par exemple $\tau = 6$, $(\overline{h} \backslash\!\!\backslash \underline{h})(1) = \underline{h}(6) - \overline{h}(5) = 4$.

Ensuite, la fonction $h \in [\ \underline{h}\ ,\ \overline{h}\ ]$ définie comme suit :

$$h(t) \ \triangleq \ \begin{cases} \overline{h}(t) & \text{pour } t < \tau_0, \\ \underline{h}(t) & \text{pour } t \geq \tau_0, \end{cases}$$

permet de satisfaire l'égalité $(h \backslash\!\!\backslash h)(t) = (\overline{h} \backslash\!\!\backslash \underline{h})(t)$ en un $t > 0$ donné. Par exemple, lorsque $t = 1$ et $\tau_0 = 6$, $(h \backslash\!\!\backslash h)(1) = (\overline{h} \backslash\!\!\backslash \underline{h})(1)$. L'expression de l'autocorrélation de h, donnée par :

$$(h \backslash\!\!\backslash h)(t) = \bigwedge_{\tau \geq 0} h(\tau - t) \backslash\!\!\backslash h(\tau),$$

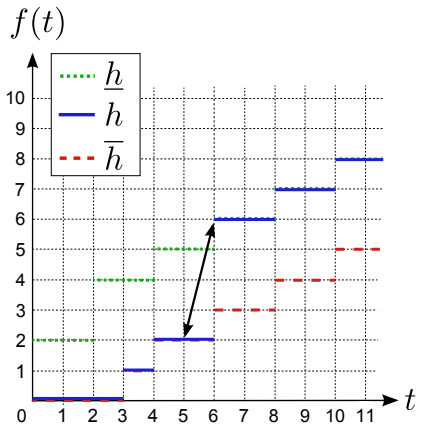


Figure 5.12 Exemple d'une fonction h pour laquelle $t = 1$ et $\tau_0 = 6$. La flèche représente la distance ($= 4$) entre $\underline{h}$ et $\overline{h}$ pour ces valeurs.

peut alors être factorisée comme suit selon les différents instants avant et après τ_0 :

$$(h \backslash h)(t) = \bigwedge_{\tau<\tau_0} \overline{h}(\tau - t) \backslash \overline{h}(\tau) \tag{5.19}$$

$$\wedge \left(\overline{h}(\tau_0 - t) \backslash \underline{h}(\tau_0)\right) \wedge \left(\bigwedge_{\tau_0<\tau<\tau_0+t} \overline{h}(\tau - t) \backslash \underline{h}(\tau)\right) \tag{5.20}$$

$$\wedge \bigwedge_{\tau\geq\tau_0+t} \underline{h}(\tau - t) \backslash \underline{h}(\tau). \tag{5.21}$$

Puisque $\underline{h} \preccurlyeq \overline{h}$ et grâce aux propriétés d'isotonie et d'antitonie de l'opérateur $\backslash$ (cf. équation (2.13)) :

$$(5.19) \Rightarrow \bigwedge_{\tau<\tau_0} \overline{h}(\tau - t) \backslash \overline{h}(\tau) \succcurlyeq \bigwedge_{\tau<\tau_0} \overline{h}(\tau - t) \backslash \underline{h}(\tau) \succcurlyeq \overline{h}(\tau_0 - t) \backslash \underline{h}(\tau_0),$$

$$(5.20) \Rightarrow \left(\overline{h}(\tau_0 - t) \backslash \underline{h}(\tau_0)\right) \wedge \left(\bigwedge_{\tau_0<\tau<\tau_0+t} \overline{h}(\tau - t) \backslash \underline{h}(\tau)\right) = \overline{h}(\tau_0 - t) \backslash \underline{h}(\tau_0),$$

$$(5.21) \Rightarrow \bigwedge_{\tau\geq\tau_0+t} \underline{h}(\tau - t) \backslash \underline{h}(\tau) \succcurlyeq \bigwedge_{\tau\geq\tau_0+t} \overline{h}(\tau - t) \backslash \underline{h}(\tau) \succcurlyeq \overline{h}(\tau_0 - t) \backslash \underline{h}(\tau_0).$$

Soit pour finir, et d'après l'équation (5.18) :

$$\forall t > 0,\ (h \backslash h)(t) = \overline{h}(\tau_0 - t) \backslash \underline{h}(\tau_0) = (\overline{h} \backslash \underline{h})(t).$$

□

Ainsi, pour un t strictement positif donné, une fonction h de l'intervalle $[\ \underline{h}\ ,\ \overline{h}\ ]$ permet de satisfaire l'égalité $(h \backslash h)(t) = (\overline{h} \backslash \underline{h})(t)$. Il existe donc une collection de

fonctions $h \in [\ \underline{h}\ ,\ \overline{h}\ ]$ telles que pour tout $t > 0$:

$$\bigwedge_{h\in[\ \underline{h}\ ,\ \overline{h}\ ]} (h \backslash h)(t) = (\overline{h} \backslash \underline{h})(t). \tag{5.22}$$

Remarque 5.28 Notons que lorsque $t = 0$:

$$(h \backslash h)(0) = 0 \quad \text{tandis que} \quad (\overline{h} \backslash \underline{h})(0) \preccurlyeq 0 \quad (\Leftrightarrow \geq 0).$$

Par exemple, sur la figure 5.12 : $(\overline{h} \backslash \underline{h})(0) = \underline{h}(2) - \overline{h}(2) = 4$.

Le précompensateur $p = \overline{h} \backslash \underline{h}$ est donc le plus grand candidat plus petit que, voire dans ce cas égal à, $\bigwedge_{h\in[\ \underline{h}\ ,\ \overline{h}\ ]} h \backslash h$. Mais cela n'est vrai que pour $t > 0$. La proposition suivante se nourrit des lemmes 5.25 et 5.27, ainsi que de la remarque 5.28, afin de donner le précompensateur neutre optimal $\hat{p}$ pour tout $t \geq 0$.

Proposition 5.29 Soient h, $\underline{h}$ et $\overline{h}$ trois fonctions du dioïde $(\mathcal{F}_{cp}, \oplus, *)^{\mathsf{caus}}$ avec $\underline{h} \preccurlyeq h \preccurlyeq \overline{h}$:

$$\bigwedge_{h\in[\ \underline{h}\ ,\ \overline{h}\ ]} h \backslash h = e \oplus \overline{h} \backslash \underline{h}.$$

Preuve. Selon le lemme 5.27 et l'équation (5.22), pour tout $t > 0$:

$$\bigwedge_{h\in[\ \underline{h}\ ,\ \overline{h}\ ]} (h \backslash h)(t) = (\overline{h} \backslash \underline{h})(t).$$

D'après la remarque 5.28, lorsque $t = 0$:

$$\bigwedge_{h\in[\ \underline{h}\ ,\ \overline{h}\ ]} (h \backslash h)(0) = 0.$$

Ainsi, $\forall t \geq 0$:

$$\bigwedge_{h\in[\ \underline{h}\ ,\ \overline{h}\ ]} h \backslash h = e \oplus \overline{h} \backslash \underline{h}.$$

□

Pour conclure, le précompensateur $\hat{p} = e \oplus \overline{h} \backslash \underline{h}$ est le plus grand précompensateur, donc le précompensateur optimal, qui est neutre pour tout système h de l'intervalle $[\ \underline{h}\ ,\ \overline{h}\ ]$.

Remarque 5.30 Ce précompensateur neutre optimal ne nécessite pas de manipuler des fonctions constantes par morceaux. Il peut s'appliquer à des fonctions affines et ultimement affines, telles que les bornes du conteneur défini dans le chapitre 4.

4.3 Extension au cas MIMO

La proposition 5.29 donnée pour tout système incertain SISO appartenant à un intervalle, peut être étendue aux systèmes MIMO. Pour un système incertain (min,+)-linéaire $H \in (\mathcal{F}_{cp}, \oplus, *)^{q\times p}$ inclus dans un intervalle $[\ \underline{H}\ ,\ \overline{H}\ ]$, le plus grand précompensateur neutre est maintenant défini par :

$$\hat{P} = \bigwedge_{H\in[\ \underline{H}\ ,\ \overline{H}\ ]} H\backslash H = E \oplus \overline{H}\backslash\underline{H},$$

pour lequel E est la matrice identité de $(\mathcal{F}_{cp}, \oplus, *)^{p\times p}$. Rappelons que l'opération $\backslash$ appliquée sur des matrices est exprimée comme suit (cf. équation (2.29)) :

$$(H\backslash H)_{ij} = \bigwedge_{k=1}^{m} H_{ik}\backslash H_{jk}.$$

Lemme 5.31 Soient $h_a, h_b \in (\mathcal{F}_{cp}, \oplus, *)^{\mathsf{caus}}$:

$$\bigwedge_{\substack{h_a\in[\ \underline{h_a}\ ,\ \overline{h_a}\ ]\\ h_b\in[\ \underline{h_b}\ ,\ \overline{h_b}\ ]}} h_a\backslash h_b \ = \ \overline{h_b}\backslash\underline{h_a}.$$

Preuve. La preuve de ce lemme est donnée par l'équation (2.13). □

Proposition 5.32 Soit $[\ \underline{H}\ ,\ \overline{H}\ ]$ un intervalle de matrices avec $\underline{H}, \overline{H} \in (\mathcal{F}_{cp}, \oplus, *)^{q\times p}$ et qui représente le comportement d'un système incertain avec p entrées et q sorties :

$$\bigwedge_{H\in[\ \underline{H}\ ,\ \overline{H}\ ]} H\backslash H = E \oplus \overline{H}\backslash\underline{H}.$$

Preuve. Lorsque $i = j$, $(H\backslash H)_{ii} = \bigwedge_{k=1}^{n} H_{ik}\backslash H_{ik}$, avec $H_{ik} \in [\ \underline{H}_{ik}\ ,\ \overline{H}_{ik}\ ]$. Grâce à la proposition 5.29 :

$$(\bigwedge_{H\in[\ \underline{H}\ ,\ \overline{H}\ ]} H\backslash H)_{ii} = \bigwedge_{k=1}^{n} (e \oplus \overline{H}_{ik}\backslash\underline{H}_{ik})_{ii}.$$

Ensuite, lorsque $i \neq j$, $(H\backslash H)_{ij} = \bigwedge_{k=1}^{n} H_{ik}\backslash H_{jk}$, avec $H_{ik} \in [\ \underline{H}_{ik}\ ,\ \overline{H}_{ik}\ ]$, $H_{jk} \in [\ \underline{H}_{jk}\ ,\ \overline{H}_{jk}\ ]$. Grâce au lemme 5.31 :

$$(\bigwedge_{H\in[\ \underline{H}\ ,\ \overline{H}\ ]} H\backslash H)_{ij} = \bigwedge_{k=1}^{n} (H_{ik}\backslash H_{jk})_{ij} \ = \ \overline{H}_{jk}\backslash\underline{H}_{ik}.$$

Finalement, $\forall i, j$:

$$(\bigwedge_{H\in[\ \underline{H}\ ,\ \overline{H}\ ]} H\backslash H)_{ij} = (E \oplus \overline{H}\backslash\underline{H})_{ij}.$$

□

4.4 Exemple d'application

Considérons un exemple d'un système incertain MIMO et calculons le précompensateur neutre optimal pour ce système. L'exemple choisi est un système (min,+)-linéaire à deux entrées ($p = 2$) et une sortie ($q = 1$), et dont la matrice de transfert appartient à la matrice d'intervalles $\mathbf{H} = ([\ \underline{H}_{11}\ ,\ \overline{H}_{11}\ ]\ \ [\ \underline{H}_{12}\ ,\ \overline{H}_{12}\ ]) \in (\mathcal{F}_{cp}, \oplus, *)^{\mathsf{caus}\ 1\times 2}$, avec $\underline{H}_{11} \preccurlyeq \overline{H}_{11}$ et $\underline{H}_{12} \preccurlyeq \overline{H}_{12}$. Les bornes de cette matrice sont connues et données comme suit :

$$\begin{aligned}[\ \underline{H}_{11}\ ,\ \overline{H}_{11}\ ] &= [\ \Delta_1^0 \oplus \Delta_2^1 * (\Delta_2^3)^\star\ ,\ \Delta_3^0 \oplus \Delta_4^1 \oplus \Delta_5^2 * (\Delta_2^3)^\star\ ],\\ [\ \underline{H}_{12}\ ,\ \overline{H}_{12}\ ] &= [\ \Delta_1^0 \oplus \Delta_2^2 * (\Delta_2^2)^\star\ ,\ \Delta_2^0 * (\Delta_1^1)^\star\ ].\end{aligned}$$

Le but est de calculer le précompensateur neutre optimal $\hat{P} \in (\mathcal{F}_{cp}, \oplus, *)^{2\times 2}$ de ce système. Puisque ce précompensateur est neutre pour tous les h contenus dans les intervalles $[\ \underline{H}_{11}\ ,\ \overline{H}_{11}\ ]$ et $[\ \underline{H}_{12}\ ,\ \overline{H}_{12}\ ]$, il ne modifie pas le comportement du système contrôlé, *i.e.* :

$$\mathbf{H} * \hat{P} = \mathbf{H}.$$

En outre, il s'agit également du plus grand neutre en réponse au critère du juste-à-temps, *i.e.* :

$$\hat{P} = \bigoplus\{P \mid \forall H \in [\ \underline{h}\ ,\ \overline{h}\ ], E \preccurlyeq P \preccurlyeq H \backslash H\} = \bigwedge_{H\in[\ \underline{h}\ ,\ \overline{h}\ ]} H \backslash H.$$

Grâce à la proposition 5.32, la valeur de la matrice $\hat{P}$ s'obtient en réalisant le calcul de $E \oplus \overline{H} \backslash \underline{H}$. Ainsi :

$$\begin{aligned}\hat{P} &= E \oplus \overline{H} \backslash \underline{H},\\ &= \begin{pmatrix} \hat{P}_{11} & \hat{P}_{12} \\ \hat{P}_{21} & \hat{P}_{22} \end{pmatrix} = \begin{pmatrix} \Delta_0^0 \oplus (\Delta_1^6 \oplus \Delta_2^8) * (\Delta_2^3)^\star & \Delta_0^4 * (\Delta_1^1)^\star \\ \varepsilon & \Delta_0^0 \oplus \Delta_1^4 * (\Delta_1^1)^\star \end{pmatrix}.\end{aligned}$$

Dans cet exemple, ce précompensateur est donné causal, soit $\hat{P} \in (\mathcal{F}_{cp}, \oplus, *)^{\mathsf{caus}\ 2\times 2}$.

5 Conclusion

Dans la première section de ce chapitre, nous avons tout d'abord rappelé les structures de contrôle existantes en théorie des systèmes (min,+)-linéaires. Nous avons notamment présenté le précompensateur p (*i.e.* un contrôleur placé en amont du système, commande en boucle ouverte) et le retour de sortie f (*i.e.* un contrôleur placé entre la sortie y et l'entrée u du système, commande en boucle fermée).

Ensuite, dans un contexte de systèmes supposés incertains, soit lorsque leurs dynamiques ne sont connues qu'à travers des intervalles $[\ \underline{h}\ ,\ \overline{h}\ ]$ ou $[\ \underline{\beta}\ ,\ \overline{\beta}\ ]$, et tout en respectant le critère du juste-à-temps, trois problèmes de commandes ont été résolus.

Dans un premier temps, la section 2 a traité le problème connu sous le nom anglais Window Flow Control dont nous avons associé les deux hypothèses de fonctionnement suivantes :

- le flux aller des données S_1 et les acquittements retours S_2 sont supposés distincts,
- les services fournis par les éléments du réseau sont encadrés par des intervalles $[\ \underline{\beta}_1\ ,\ \overline{\beta}_1\ ]$ et $[\ \underline{\beta}_2\ ,\ \overline{\beta}_2\ ]$.

Dans ce contexte, le contrôleur assimilé à une structure de type retour de sortie, est une fenêtre qui filtre les entrées afin de garantir une quantité de données en transit dans le réseau toujours plus petite que sa taille. La proposition 5.11 fournit le calcul suivant de la taille optimale $\hat{w}$ de cette fenêtre :

$$\hat{w} = \Big((\underline{\beta}_1 \backslash\!\!\backslash \underline{\beta}_1 \not\phi (\underline{\beta}_2 \underline{\beta}_1)) \wedge (\overline{\beta}_1 \backslash\!\!\backslash \overline{\beta}_1 \not\phi (\overline{\beta}_2 \overline{\beta}_1))\Big)(0),$$

et permet ainsi d'obtenir une dynamique en boucle fermée (avec S_1 et S_2) identique à celle du système en boucle ouverte (soit S_1 seul). Ce travail est présenté dans les articles [Le Corronc 2010b] et [Le Corronc 2011b].

Dans un second temps, le problème de la réduction de l'incertitude en sortie d'un système incertain $[\ \underline{h}\ ,\ \overline{h}\ ]$ est traité dans la section 3. Nous avons tout d'abord proposé en 5.17 que l'ajout d'un précompensateur n'augmente en aucun cas l'incertitude en sortie du système contrôlé et ce pour n'importe quelle entrée u, *i.e.* :

$$\begin{cases} \Sigma_\tau(\underline{h}p, \overline{h}p) \leq \Sigma_\tau(\underline{h}, \overline{h}) \\ \Sigma_\nu(\underline{h}p, \overline{h}p) \leq \Sigma_\nu(\underline{h}, \overline{h}) \end{cases} \quad \text{puisque} \quad \overline{h}p \backslash\!\!\backslash \underline{h}p \succcurlyeq \overline{h} \backslash\!\!\backslash \underline{h}.$$

Par ailleurs, nous avons montré qu'un tel précompensateur offre la possibilité de diminuer cette incertitude, jusqu'à atteindre une valeur temporelle τ_0 et/ou événementielle ν_0 fixée. Le plus grand précompensateur $\hat{p}$ permettant d'atteindre cet objectif est finalement présenté dans le corollaire 5.21 comme le plus grand point fixe de l'application suivante :

$$p = p \wedge \overline{h} \backslash\!\!\backslash (\Delta_0^{-\nu_0} \underline{h}p) \wedge \overline{h} \backslash\!\!\backslash (\Delta_{\tau_0}^{0} \underline{h}p).$$

La publication [Le Corronc 2010a] a permis de diffuser ce résultat.

Pour finir, la section 4 propose d'appliquer à un système incertain $[\ \underline{h}\ ,\ \overline{h}\ ]$, un précompensateur qui est à la fois neutre pour tous les systèmes inclus dans l'intervalle $[\ \underline{h}\ ,\ \overline{h}\ ]$, et bien sûr optimal selon le critère du juste-à-temps. Ce problème suppose en outre que le système $h \in [\ \underline{h}\ ,\ \overline{h}\ ]$ est un système (min,+)-linéaire, c'est-à-dire que pour toute entrée u et pour tout h dans l'intervalle : $y = h * u$. Une telle restriction n'est pas nécessaire pour les deux contrôleurs précédents. Ce précompensateur neutre $\hat{p}$ qui permet donc d'atteindre l'objectif suivant :

$$\hat{p} = \bigoplus \{p \mid \forall h \in [\ \underline{h}\ ,\ \overline{h}\ ], e \preccurlyeq p \preccurlyeq h \backslash\!\!\backslash h\},$$

est proposé pour les systèmes SISO comme suit (cf. proposition 5.29) :

$$\hat{p} = e \oplus \overline{h} \backslash\!\!\backslash \underline{h}.$$

Son extension aux systèmes MIMO est donnée dans la proposition 5.32 par :

$$\hat{P} = E \oplus \overline{H} \backslash\!\!\backslash \underline{H}.$$

Ce précompensateur neutre optimal est présenté dans la publication [Le Corronc 2009b].

Conclusion générale

Les travaux présentés dans ce mémoire ont pour sujet principal les systèmes (min,+)-linéaires, c'est-à-dire une classe de Systèmes à Événements Discrets pour lesquels apparaissent principalement des phénomènes de délai temporel, de décalage événementiel et de synchronisation. Les domaines d'application sont la modélisation et le contrôle de systèmes manufacturiers, de systèmes de transports, mais aussi de réseaux de communications, ce dernier champ d'application étant connu sous l'appellation Network Calculus.

Dans ce contexte d'étude, nous nous intéressons dans un premier temps à l'élaboration d'outils de calculs pour les systèmes (min,+)-linéaires. Depuis les travaux de l'équipe Max-Plus et de Stéphane Gaubert (cf. [Gaubert 1992]), il est connu que le calcul des relations de transfert de ces systèmes repose sur la manipulation de fonctions (ou séries formelles) qui ont des propriétés de périodicité. Plus précisément, la fonction de transfert d'un système mono-entrée mono-sortie est nécessairement une fonction qui, après un fonctionnement transitoire, est ultimement pseudo-périodique. Cependant, la complexité algorithmique des opérations dépend en grande partie de la longueur du comportement transitoire. En effet, le calcul de la fonction de transfert d'un système peut nécessiter la manipulation et le stockage d'un nombre arbitrairement grand de données, ce qui pénalise fortement les outils informatiques actuels utilisant cette représentation.

Les premières contributions de cette thèse visent à développer un outil de calculs plus efficace, en s'appuyant sur une représentation approchée des fonctions pseudo-périodiques. Dans l'idée de l'approche ensembliste présentée dans [Jaulin 2001], nous encadrons les fonctions manipulées par des ensembles « conteneurs », et nous proposons des fonctions d'inclusion pour les opérations fondamentales à savoir le min, l'inf-convolution et l'étoile de Kleene (clôture sous-additive).

Par construction, les fonctions manipulées sont incluses dans des intervalles (conteneurs), dont les bornes sont le résultat d'approximations convexes. Cette caractéristique permet de relier ce travail à l'analyse convexe (cf. [Rockafellar 1997]). Plus particulièrement, le système de conteneurs proposé dans ce mémoire consiste à manipuler les systèmes (min,+)-linéaires modulo la transformée de Legendre-Fenchel. De ce fait, nous savons que ce système de calcul approché conserve

néanmoins certaines caractéristiques importantes comme la pente ultime des fonctions. Il apparaît en outre que la complexité algorithmique des fonctions d'inclusion est linéaire pour le min et l'inf-convolution, et quasi-linéaire pour l'étoile de Kleene. Tous ces algorithmes ont été mis en oeuvre au sein d'une librairie de calculs développée en C++ et appelée ContainerMinMaxGD.

Dans un second temps, différents problèmes de commande sont abordés. Tous ces problèmes ont en commun qu'ils s'appuient sur une représentation approchée des systèmes. Il s'agit soit de systèmes pour lesquels certains paramètres sont incertains ou sujets à des fluctuations, soit de systèmes dont le comportement n'est pas connu exactement (ce qui apparaît lorsque leur transfert est calculé avec la librairie ContainerMinMaxGD). Dans tous les cas, leur comportement est décrit par un intervalle dont les bornes sont connues.

Ainsi, nous avons transposé quelques structures de contrôle existantes pour les systèmes exacts (précompensateur et retour de sortie) afin de proposer des lois de commande qui nous semblent originales dans ce contexte de systèmes incertains.

Dans le cadre du Network Calculus, nous revisitons notamment le problème du Window Flow Control pour lequel les courbes de service décrivant le traitement des flux de données et les acquittements sont différentes, et toutes deux représentées par des intervalles. D'un point de vue système, la recherche de la fenêtre optimale revient à un problème de synthèse de contrôleur de type retour de sortie. La taille de la fenêtre est calculée de façon à ce que le flux d'acquittement ne freine pas le flux de données.

Dans un second temps, et ce toujours dans le contexte des systèmes décrits par des intervalles, nous nous intéressons à l'impact de la loi de commande sur l'incertitude de la sortie. Nous montrons que l'ajout d'un précompensateur à un système incertain fait nécessairement décroître cette incertitude. Le problème de commande considéré est alors celui de la recherche d'un correcteur tel que l'incertitude soit bornée par une valeur donnée. Dans les cas les plus favorables, une annulation de l'incertitude peut être envisagée grâce au correcteur, c'est-à-dire que le système incertain contrôlé est un système (min,+)-linéaire.

Enfin, le dernier problème considéré est celui de la synthèse d'un contrôleur neutre. Il s'agit alors d'un contrôleur qui retarde l'entrée le plus possible sans pour

autant retarder davantage la sortie par rapport au système seul. Dans le cadre des systèmes manufacturiers ou de traffic, un tel correcteur a l'avantage d'améliorer les flux internes (les tailles des files d'attente), sans dégrader les performances du système. Ce problème, déjà traité dans la littérature, est proposé ici dans le cas des systèmes incertains.

Dans l'ensemble, ce travail fournit donc une méthode de calcul approché pour les systèmes (min,+)-linéaires, ainsi qu'un certain nombre de problèmes de contrôle qui peuvent être traités malgré l'incertitude engendrée par un tel outil de calculs.

Suite à ces travaux, plusieurs perspectives apparaissent. Tout d'abord, il serait intéressant de compléter la librairie de calculs avec l'opération quotient (déconvolution). Mais quelle est la signification de la résiduation sur les conteneurs introduits dans ce mémoire ? D'un autre côté, il reste encore de nombreuses structures et lois de commande à adapter dans ce cadre des systèmes (min,+)-linéaires incertains. Par exemple, est-il possible d'appliquer à un système décrit par un intervalle, un contrôleur de type retour de sortie neutre pour toutes les dynamiques de cet intervalle ?

Annexe A

Dioïde de séries formelles $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$

Sommaire

1 Modélisation dans le dioïde de séries formelles $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ . . 144

1.1 Transformée en γ,δ 144

1.2 Monotonie des trajectoires 145

1.3 Relation de transfert 146

1.4 Périodicité : réalisabilité, causalité et rationalité 148

2 Librairie de calculs MinMaxGD 151

2.1 Formes propre, simple et canonique 151

2.2 Clôture des opérations dans MinMaxGD 153

2.3 Opération produit : $\otimes$ 153

Dans le chapitre 3, la modélisation d'un système (min,+)-linéaire est réalisée grâce aux fonctions compteurs d'événements définies sur le dioïde $\overline{\mathbb{R}}_{min}$, c'est-à-dire en considérant principalement le nombre d'occurrences des événements. Or d'autres points de vue sont également possible. Par exemple, grâce aux fonctions dateurs définies sur le dioïde $\overline{\mathbb{R}}_{max}$, les dates de ces événements peuvent être prises en compte, tandis que le dioïde $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!] = \mathbb{B}[\![\gamma,\delta]\!]_{/\gamma^\star(\delta^{-1})^\star}$ permet de réunir ces deux points de vue (événementiel/temporel) en un seul.

La modélisation des systèmes à travers le dioïde $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ est présentée dans cette annexe. Quelques éléments liés à l'utilisation de la librairie MinMaxGD permettant de manipuler les séries formelles de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ sont également donnés dans un second temps.

1 Modélisation dans le dioïde de séries formelles $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$

Dans la théorie conventionnelle des systèmes, face à un produit de convolution tel que donné dans la représentation entrée/sortie (3.8) et rappelée ici :

$$y = h * u,$$

il est classique d'introduire la transformée de Fourier ou de Laplace des fonctions considérées. Les modèles discrets ont également une représentation fréquentielle : la transformée en z. De la même manière, la classe des systèmes (min,+)-linéaires dispose d'une transformée permettant de prendre en compte à la fois les aspects événementiels et temporels des représentations en compteur ou en dateur. Les décalages événementiels et temporels sont décrits au moyen d'opérateurs représentés formellement par les variables γ et δ.

1.1 Transformée en γ, δ

Soient x un événement et $x(n,t)$ la suite qui lui est associée et dont les coefficients sont dans $\mathbb{B} = \{\varepsilon, e\}$. L'occurrence $x(n_i, t_i)$ de l'événement x peut s'interpréter de deux façons :

- l'occurrence n_i de l'événement x a lieu au plus tôt à la date t_i,
- à la date t_i, il y a eu au moins n_i occurrences de l'événement x.

D'après la définition suivante, une telle suite peut être associée à une série du dioïde $\mathbb{B}[\![\gamma,\delta]\!]$.

Définition A.1 (Transformée en γ, δ). La transformée en γ, δ de la suite $x(n,t)$ est donnée par la série formelle suivante dans $\mathbb{B}[\![\gamma,\delta]\!]$:

$$x(\gamma,\delta) = \bigoplus_{(n,t)\in\mathbb{Z}^2} x(n,t)\gamma^n\delta^t.$$

Une suite $x(n,t)$ ainsi que la série $x(\gamma,\delta)$ qui lui est associée correspond alors à la trajectoire de l'événement x, avec dates t_i et numéro n_i.

1.2 Monotonie des trajectoires

Grâce à cette transformée, les éléments du dioïde $\mathbb{B}[\![\gamma,\delta]\!]$ semblent bien adaptés pour représenter les trajectoires des événements des systèmes étudiés. Cependant, puisqu'un compteur décrit un cumul d'événements, de telles trajectoires sont nécessairement monotones croissantes :

le nombre d'occurrences d'un événement à la date $t-1$
est nécessairement plus petit que celui à l'instant t.

Idem pour les représentations par des dateurs :

la date de l'occurrence de l'événement numéroté $n+1$
est obligatoirement supérieure à celle de l'événement numéroté n.

Autrement dit, pour une occurrence donnée $x(n_1,t_1)$ et représentée par le monôme $\gamma^{n_1}\delta^{t_1}$ dans $\mathbb{B}[\![\gamma,\delta]\!]$, il ne faut pas considérer uniquement cette date t_1 et ce numéro n_1, mais bien toutes les occurrences $x(n_i,t_i)$ telles que $n_i \geq n_1$ et $t_1 \leq t_i$. Dans $\mathbb{B}[\![\gamma,\delta]\!]$, cela revient à considérer tous les monômes situés dans le cône Sud-Est du monôme $\gamma^{n_1}\delta^{t_1}$.

Un tel filtrage des séries de $\mathbb{B}[\![\gamma,\delta]\!]$ peut être effectué de la manière suivante :

$$\begin{aligned}
\gamma^n\delta^t &= (\gamma^n \oplus \gamma^{n+1} \oplus \gamma^{n+2} \oplus \ldots) \otimes (\delta^t \oplus \delta^{t-1} \oplus \delta^{t-2} \oplus \ldots), \\
&\qquad\qquad \text{duplication du point d'origine jusqu'à } \gamma^{+\infty} \text{ sur l'axe horizontal} \\
&\qquad\qquad \text{et jusqu'à } \delta^{-\infty} \text{ sur l'axe vertical,} \\
&= \gamma^n\delta^t\Big(\bigoplus_{i=0}^{+\infty}\gamma^i\Big) \otimes \Big(\bigoplus_{i=0}^{+\infty}\delta^{-i}\Big), \\
&= \gamma^n\delta^t\gamma^\star \otimes (\delta^{-1})^\star \qquad \text{d'après la définition 2.64,} \\
&= \gamma^n\delta^t\gamma^\star(\delta^{-1})^\star.
\end{aligned}$$

Ainsi, seules les séries formelles de $\mathbb{B}[\![\gamma,\delta]\!]$ invariantes à la multiplication par $\gamma^\star(\delta^{-1})^\star$ doivent être considérées afin de prendre en compte cette propriété de monotonie des trajectoires. L'utilisation du dioïde $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ prend en compte cette caractéristique puisque ses éléments sont des classes d'équivalence modulo $\gamma^\star(\delta^{-1})^\star$.

Dans $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$, le produit des séries de $\mathbb{B}[\![\gamma,\delta]\!]$ par $\gamma^\star(\delta^{-1})^\star$ implique les règles de simplification suivantes :

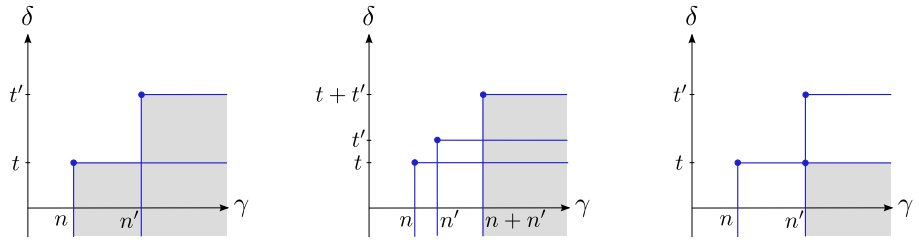
$$\begin{aligned}
\gamma^n\delta^t \oplus \gamma^{n'}\delta^t &= \gamma^{min(n,n')}\delta^t, \\
\gamma^n\delta^t \oplus \gamma^n\delta^{t'} &= \gamma^n\delta^{max(t,t')}.
\end{aligned}$$

De plus, puisque $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ est un dioïde complet, tout couple $a, b \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ admet un plus grand minorant noté $a \wedge b$. Dans le cas où a et b sont des monômes, on obtient la relation suivante permettant, puisque le dioïde est distributif, d'établir la borne inférieure $\wedge$ de deux éléments quelconques de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$:

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

Graphiquement, la correspondance des opérations $\oplus$, $\otimes$ et $\wedge$ entre les monômes de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ et les cônes de $\mathbb{Z}^2$, est donnée comme suit et représentée par la figure A.1 :

- $\gamma^n\delta^t \oplus \gamma^{n'}\delta^{t'}$: union des cônes sud-est de sommets respectifs (n,t) et (n',t').
- $\gamma^n\delta^t \otimes \gamma^{n'}\delta^{t'}$: cône de sommet $(n+n', t+t')$ (soit le cône dont le sommet est la somme vectorielle des sommets (n,t) et (n',t')).
- $\gamma^n\delta^t \wedge \gamma^{n'}\delta^{t'}$: intersection des cônes sud-est de sommets respectifs (n,t) et (n',t').



(a) $\gamma^n\delta^t \oplus \gamma^{n'}\delta^{t'}$ (b) $\gamma^n\delta^t \otimes \gamma^{n'}\delta^{t'} = \gamma^{n+n'}\delta^{t+t'}$ (c) $\gamma^n\delta^t \wedge \gamma^{n'}\delta^{t'} = \gamma^{\max(n,n')}\delta^{\min(t,t')}$

Figure A.1 Représentation graphique des opérations sur $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$..

1.3 Relation de transfert

La modélisation analytique d'un système peut donc être réalisée à travers le dioïde $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$, tout en respectant la condition de trajectoires monotones croissantes des systèmes grâce à la congruence $\gamma^*(\delta^{-1})^\star$.

La représentation d'état donnée en (3.2) devient comme suit avec $A \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^{n\times n}$, $B \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^{n\times p}$ et $C \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^{q\times n}$:

$$\begin{cases} x(\gamma,\delta) = Ax(\gamma,\delta) \oplus Bu(\gamma,\delta), \\ y(\gamma,\delta) = Cx(\gamma,\delta). \end{cases} \tag{A.1}$$

Remarque A.2 Les décalages événementiel et temporel sont réalisés dans ce système d'équations via les opérateurs γ et δ :

$$\begin{aligned}\gamma \otimes x(\gamma,\delta) &= \bigoplus_{n\in\mathbb{Z}} x(n,t)\otimes\gamma^{n+1}\delta^{t} = \bigoplus_{n\in\mathbb{Z}} x(n-1,t)\otimes\gamma^{n}\delta^{t},\\ \delta \otimes x(\gamma,\delta) &= \bigoplus_{t\in\mathbb{Z}} x(n,t)\otimes\gamma^{n}\delta^{t+1} = \bigoplus_{n\in\mathbb{Z}} x(n,t-1)\otimes\gamma^{n}\delta^{t}.\end{aligned}$$

Enfin, d'après le théorème 2.68 de l'étoile de Kleene, le calcul de la matrice de transfert $H \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^{q\times p}$ se ramène au produit de matrices suivant :

$$H(\gamma,\delta) = CA^{\star}B.$$

Exemple A.3 Les trajectoires des événements $x_i(\gamma,\delta)$, soient les tirs des transitions du GET de la figure 3.1 présenté dans le chapitre 3, sont liées dans $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ par les égalités suivantes :

$$\begin{cases} x_1(\gamma,\delta) = \gamma^0\delta^0 u_1(\gamma,\delta) \oplus \gamma^1\delta^2 x_1(\gamma,\delta),\\ x_2(\gamma,\delta) = \gamma^0\delta^0 u_2(\gamma,\delta) \oplus \gamma^1\delta^3 x_2(\gamma,\delta),\\ x_3(\gamma,\delta) = \gamma^1\delta^3 x_1(\gamma,\delta) \oplus \gamma^0\delta^8 x_2(\gamma,\delta),\\ y(\gamma,\delta) = \gamma^0\delta^2 x_3(\gamma,\delta). \end{cases}$$

Dans ces équations, les éléments $x(\gamma,\delta)$ sont représentés sous leur forme minimale. La représentation d'état du système est donnée comme suit :

$$\begin{cases} \begin{pmatrix} x_1\\ x_2\\ x_3 \end{pmatrix} = \begin{pmatrix} \gamma^1\delta^2 & \varepsilon & \varepsilon\\ \varepsilon & \gamma^1\delta^3 & \varepsilon\\ \gamma^1\delta^3 & \gamma^0\delta^8 & \varepsilon \end{pmatrix} \begin{pmatrix} x_1\\ x_2\\ x_3 \end{pmatrix} \begin{pmatrix} e & \varepsilon\\ \varepsilon & e\\ \varepsilon & \varepsilon \end{pmatrix} \begin{pmatrix} u_1\\ u_2 \end{pmatrix},\\ y = \begin{pmatrix} \varepsilon & \varepsilon & \gamma^0\delta^2 \end{pmatrix} \begin{pmatrix} x_1\\ x_2\\ x_3 \end{pmatrix}, \end{cases}$$

et sa matrice de transfert a pour valeur :

$$H(\gamma,\delta) = (\gamma^1\delta^5(\gamma^1\delta^2)^* \quad \gamma^0\delta^{10}(\gamma^1\delta^3)^*).$$

Remarque A.4 (Périodicité des éléments de H). Comme nous l'avions remarqué lors de la modélisation dans le dioïde $\overline{\mathbb{R}}_{min}$, il est à noter que les éléments de cette matrice contiennent tous deux une seule étoile de monôme ($(\gamma^1\delta^2)^*$ et $(\gamma^1\delta^3)^*$) faisant apparaître un caractère ultimement périodique. Cette propriété de périodicité est détaillée dans la section suivante.

1.4 Périodicité : réalisabilité, causalité et rationalité

De manière générale, les éléments de la matrice de transfert H sont des séries formelles périodiques et dans $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$, cette périodicité est exprimée selon la définition suivante.

Définition A.5 (Périodicité). Une série $s \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ est dite périodique s'il existe deux polynômes p et q et un monôme r de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ définis par :

$$p = \bigoplus_{i=0}^{\alpha} \gamma^{n_i}\delta^{t_i}, \quad q = \bigoplus_{j=0}^{\beta} \gamma^{N_j}\gamma^{T_j} \quad et \quad r = \gamma^{\nu}\delta^{\tau}, \tag{A.2}$$

tels que :

$$s = p \oplus qr^{\star}. \tag{A.3}$$

Une matrice est périodique si toutes ses composantes sont périodiques.

Remarque A.6 (Polynôme de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$). Un polynôme de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ est une série périodique dégénérée ($r = \varepsilon$).

Définition A.7 (Pente asymptotique). La pente asymptotique d'une série périodique $s \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ est notée $\sigma(s)$ et est définie comme le ratio : $\sigma(s) = \frac{\nu}{\tau}$.

Sous la forme donnée par les équations (A.2) et (A.3), le polynôme p caractérise la période transitoire de la série alors que le polynôme q représente le motif du régime périodique. Cette périodicité est fournie par le monôme r avec le ratio σ. La figure A.2 illustre cette configuration.

Cette caractéristique principale de périodicité des systèmes (min,+)-linéaire peut être transposée selon plusieurs points de vue à travers des notions de réalisabilité, causalité et rationalité comme le montre le théorème suivant.

Théorème A.8 ([Baccelli 1992, Th. 5.39] et [Cottenceau 1999a, Th. 2.37]). Pour toute matrice de transfert $H \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$, les propositions suivantes sont équivalentes :

(i) H est réalisable,

(ii) H est périodique et causale,

(iii) H est rationnelle.

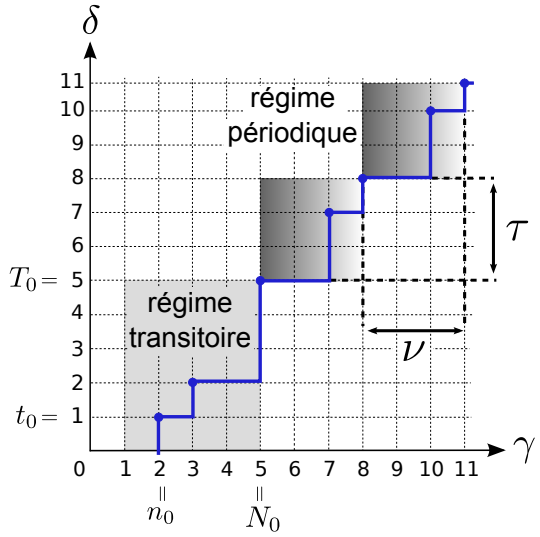


Figure A.2 Série périodique de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$.

Les définitions de réalisabilité, causalité et rationalité sont données ci-dessous.

Définition A.9 (Réalisabilité). Une matrice $H \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^{p\times m}$ est dite réalisable s'il existe quatre matrices A_1, A_2, B et C de tailles respectives $n\times n$, $n\times n$, $n\times p$, et $q\times n$, et à coefficients dans $\mathbb{B}$ telles que :

$$H = C(\gamma A_1 \oplus \delta A_2)^* B.$$

Remarque A.10 Exprimé différemment, H est réalisable s'il existe un système (min,+)-linéaire dont la matrice de transfert est H.

Définition A.11 (Causalité). Une série $s \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ est dite causale si $s = \varepsilon(\gamma,\delta)$ (la série nulle de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$) ou si chacun de ses monômes est à exposant dans $\mathbb{N}$. Une matrice est causale si toutes ses composantes sont causales.

Définition A.12 (Projection causale). Soit $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^{\mathsf{caus}}$ le sous-dioïde des séries causales de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$. L'injection canonique $\mathsf{I}_{\mathsf{caus}}$ définie de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^{\mathsf{caus}}$ dans $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ est résiduable et sa résiduée est notée $\mathsf{Pr}_{\mathsf{caus}} : \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!] \mapsto \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^{\mathsf{caus}}$. Formellement, la série $\mathsf{Pr}_{\mathsf{caus}}(s)$ est la plus grande série causale plus petite ou égale

à s :

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

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 A.13 Graphiquement, la projection causale consiste à ne conserver que les monômes situés dans le cadran Nord-Est du plan $\mathbb{Z}^2$. Si une série x est causale, alors elle est invariante par $\mathsf{Pr}_{\mathsf{caus}}$, soit $\mathsf{Pr}_{\mathsf{caus}}(x) = x$.

Remarque A.14 La définition A.12 est une reformulation de la définition 3.23 dans le cadre des séries de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$.

Définition A.15 (Rationalité). Une série $s \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ est dite rationnelle si son représentant minimal peut s'écrire avec un nombre fini d'opérations $\{\oplus,\otimes,\star\}$ sur l'ensemble $\{\varepsilon,e,\gamma,\delta\}$. Une matrice est rationnelle si toutes ses composantes sont rationnelles.

Remarque A.16 (Équivalence de notations). Grâce aux équivalences apportées par le théorème A.8, la présentation d'un système (min,+)-linéaire peut être introduite de plusieurs manière.

- Un système avec une matrice H rationnelle a nécessairement une réalisation finie de la forme :
$$\begin{cases} x = Ax \oplus Bu, \\ y = Cx. \end{cases}$$
- Si les matrices A, B et C de la représentation d'état A.1 sont positives, alors $H = CA^\star B$ est tel que H_{ij} sont des séries périodiques et causales.
- La matrice de transfert d'un système (min,+)-linéaire peut être exprimée avec une combinaison finie de $\{\gamma,\delta\}$ et $\{\oplus,\otimes,{}^\star\}$ et possède également une représentation périodique et causale. En particulier, il existe une représentation périodique canonique pour les séries rationnelles de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ ce qui a permis le développement de la librairie de calculs MinMaxGD présentée dans la section suivante.

2 Librairie de calculs MinMaxGD

Une librairie de calculs appelée **MinMaxGD** (cf. [Cottenceau 2000]) et disponible à l'adresse suivante : `http://www.istia.univ-angers.fr/~hardouin/`, a été développée afin de manipuler des éléments du dioïde $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$. Les opérations disponibles ainsi que certaines des complexités qui leurs sont associées sont décrites succinctement dans cette section. Pour plus de détails sur les algorithmes se référer à [Cottenceau 1999a] et à [Gaubert 1992].

Remarque A.17 (Manipulation des fonctions de $\mathcal{F}_{cp}$). Cette librairie MinMaxGD peut également être utilisée afin de manipuler des fonctions du dioïde $(\mathcal{F}_{cp},\oplus,*)$, en transformant ces dernières en série de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$. Soit f une fonction de $\mathcal{F}_{cp}$ définie comme suit :

$$f=\bigoplus_{i\geq 0}^{n}\Delta_{t_i}^{k_i}\oplus\Big(\bigoplus_{j\geq 0}^{m}\Delta_{T_j}^{K_j}\Big)*(\Delta_{\tau}^{\nu})^{\star},$$

avec t_i, k_i, T_j et K_j entiers. La série $s\in\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ qui lui est associée est la suivante :

$$s=p\oplus q\otimes r^{\star}=\bigoplus_{i\geq 0}^{n}\gamma^{k_i}\delta^{t_i}\oplus\Big(\bigoplus_{j\geq 0}^{m}\gamma^{K_j}\delta^{T_j}\Big)\otimes(\gamma^{\nu}\delta^{\tau})^{\star}.$$

2.1 Formes propre, simple et canonique

Afin de manipuler aisément les séries périodiques de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ et d'en réaliser des comparaisons, il est avant tout nécessaire d'introduire la notion de représentation canonique de ces séries.

Définition A.18 (Forme canonique d'un polynôme). Un polynôme est dit sous forme canonique lorsqu'il correspond à son représentant minimal dans $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ (cf. propriété 2.41), *i.e.* les sommets de ses cônes dans le plan $\mathbb{Z}^2$. Autrement dit, soit $p\in\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ un polynôme tel que $p=\bigoplus_{i=0}^{\alpha}\gamma^{n_i}\delta^{t_i}$. p est sous forme canonique si $n_0<n_1<\ldots<n_\alpha$ et $t_0<t_1<\ldots<t_\alpha$.

Soit la série $s\in\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ définie telle que (cf. équations (A.2) et (A.3)) :

$$s=p\oplus qr^{\star}\quad\text{avec}\quad p=\bigoplus_{i=0}^{\alpha}\gamma^{n_i}\delta^{t_i},\quad q=\bigoplus_{j=0}^{\beta}\gamma^{N_j}\gamma^{T_j}\quad\text{et}\quad r=\gamma^{\nu}\delta^{\tau}. \tag{A.4}$$

La pente asymptotique de s est donnée par :

$$\sigma(s)=\nu/\tau.$$

Définition A.19 (Formes dégénérée et simple d'une série de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$). La série s donnée par l'équation (A.4) est dite dégénérée lorsque ν ou τ est nul. Elle est dite simple, ou élément simple, lorsque $s = \gamma^n\delta^t(\gamma^\nu\delta^\tau)^\star$.

Définition A.20 (Forme propre d'une série de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$). La série s donnée par l'équation (A.4) est dite sous forme propre si :

- les polynômes p et q sont sous forme canonique,
- $(n_\alpha, t_\beta) < (N_0, T_0)$ (le premier monôme de q doit être situé dans le plan $\mathbb{Z}^2$ à droite et au-dessus du dernier monôme de p),
- $(N_\beta, T_\beta) - (N_0, T_0) < (\nu, \tau)$ (le polynôme q doit représenter un motif inférieur à celui donné par le monôme r).

Définition A.21 (Forme canonique d'une série de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$). La série s donnée par l'équation (A.4) est dite sous forme canonique si elle est sous forme propre et si les couples (n_α, t_β) et (ν, τ) sont minimaux. Autrement dit, la taille du transitoire p et la longueur du motif périodique q sont minimisés.

Exemple A.22 Soit la série $s \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ suivante :

$$s = e \oplus \gamma^2\delta^1 \oplus \gamma^3\delta^3 \oplus (\gamma^7\delta^4 \oplus \gamma^8\delta^5)(\gamma^2\delta^2)^\star.$$

L'ensemble $qr^\star = (\gamma^7\delta^4 \oplus \gamma^8\delta^5)(\gamma^2\delta^2)^\star$ correspond graphiquement à un escalier dont la taille des marches est $(1,1)$. Pour un résultat équivalent, il est possible de réduire la taille du polynôme q en le factorisant par $(e \oplus \gamma^1\delta^1)$. L'égalité suivante est ainsi obtenue :

$$\begin{aligned}
qr^\star &= (\gamma^7\delta^4 \oplus \gamma^8\delta^5)(\gamma^2\delta^2)^\star, \\
&= \gamma^7\delta^4(e \oplus \gamma^1\delta^1)(\gamma^2\delta^2)^\star, \\
&= \gamma^7\delta^4(e \oplus \gamma^1\delta^1)(e \oplus \gamma^2\delta^2 \oplus \gamma^4\delta^4 \oplus \ldots), \\
&= \gamma^7\delta^4(e \oplus \gamma^1\delta^1 \oplus \gamma^2\delta^2 \oplus \gamma^3\delta^3 \oplus \gamma^4\delta^4 \oplus \ldots), \\
&= \gamma^7\delta^4(\gamma^1\delta^1)^\star.
\end{aligned}$$

La forme canonique de la série s est alors donnée par :

$$s = e \oplus \gamma^2\delta^1 \oplus \gamma^3\delta^3 \oplus \gamma^7\delta^4(\gamma^1\delta^1)^\star.$$

2.2 Clôture des opérations dans MinMaxGD

Les opérations somme $\oplus$, produit $\otimes$, inf $\wedge$ et résiduation $\backslash$ sur des séries périodiques de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$, sont implémentées dans la librairie de calculs MinMaxGD. Tout d'abord, les opérations $\oplus$, $\otimes$, $\wedge$ et $\backslash$ sont des opérations closes dans le dioïde $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ comme le montre le théorème suivant.

Théorème A.23 (Clôture des opérations dans $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$) L'ensemble des séries périodiques de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ est stable pour la somme, le produit, l'inf et la résiduation. En outre, pour s_1 et s_2 deux séries périodiques non dégénérées (telles que $\nu_1,\nu_2 \neq 0$ et $\tau_1,\tau_2 \neq 0$), on obtient les résultats suivants :

$$\begin{aligned}\sigma(s_1 \oplus s_2) &= \min(\sigma(s_1),\sigma(s_2)),\\ \sigma(s_1 \otimes s_2) &= \min(\sigma(s_1),\sigma(s_2)),\\ \sigma(s_1 \wedge s_2) &= \max(\sigma(s_1),\sigma(s_2)).\end{aligned}$$

Si $\sigma(s_1) \leq \sigma(s_2)$, alors :

$$\sigma(s_2 \backslash s_1) = \sigma(s_1),$$

sinon, $s_2 \backslash s_1 = \varepsilon$.

Preuve. Les preuves de ce théorème sont données dans [Gaubert 1992, Chap. 7, Annexe A] et [Cottenceau 1999a, Annexe B]. □

Corollaire A.24 ([Cottenceau 1999a, Cor. 2.40]). L'ensemble des séries périodiques et causales de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ est stable pour la somme, le produit et l'étoile de Kleene.

Ensuite, les techniques employées pour exprimer le résultat d'opérations sur les séries périodiques reposent le plus souvent sur une bonne compréhension de la manipulation des éléments simples. A titre d'exemple la méthode de calcul du produit de séries simples est donnée avec la complexité algorithmique permettant de l'obtenir.

2.3 Opération produit : $\otimes$

2.3.1 Produit de polynômes

Théorème A.25 (Calcul et complexité algorithmique du produit de polynômes sous forme canonique). Soient $p = \bigoplus_{i=0}^{m} \gamma^{n_i}\delta^{t_i}$ et $p' = \bigoplus_{j=0}^{m'} \gamma^{n'_j}\delta^{t'_j}$ deux

polynômes de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ donnés sous forme canonique. Le produit $\otimes$ de ces deux polynômes est donné par :

$$p \otimes p' = \bigoplus_{i=0}^{m} \bigoplus_{j=0}^{m'} \gamma^{n_i+n'_j}\delta^{t_i+t'_j},$$

avec $p \otimes p'$ un polynôme également sous forme canonique. La complexité de ce calcul est en $\mathcal{O}(m'm\log(m))$.

Preuve. Ce calcul est tout d'abord composé de $(m \times m')$ produits de monômes. Ensuite, il faut réaliser la somme de ces (mm') monômes en appliquant les règles de simplification. Le coût total de l'opération est donc en $\mathcal{O}(m'm\log(m))$. Il est éventuellement judicieux de commuter les polynômes p et p'.

□

2.3.2 Produit de séries

Lors du calcul du produit de deux séries s et s' de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ dont les pentes asymptotiques sont différentes, il est nécessaire de connaître à partir de quelle valeur entière notée K, la série s est définitivement « au-dessus » de la série s'. Cette valeur est donnée par le lemme suivant.

Lemme A.26 (Domination [Gaubert 1992]). Soient $s = \gamma^n\delta^t(\gamma^\nu\delta^\tau)^\star$ et $s' = \gamma^{n'}\delta^{t'}(\gamma^{\nu'}\delta^{\tau'})^\star$ deux séries simples de pentes asymptotiques différentes. Si $\sigma(s) = \nu/\tau < \sigma(s') = \nu'/\tau'$, alors il existe un entier $K \in \mathbb{N}$ tel que :

$$\gamma^{n'}\delta^{t'} \otimes \gamma^{K\nu'}\delta^{K\tau'}(\gamma^{\nu'}\delta^{\tau'})^\star \preccurlyeq \gamma^n\delta^t(\gamma^\nu\delta^\tau)^\star. \tag{A.5}$$

avec :

$$K = \max\left(\left\lceil\frac{\nu(t'-t)+\tau(n-n')+\nu\tau}{\tau\nu'-\nu\tau'}\right\rceil, \left\lceil\frac{n-n'}{\nu'}\right\rceil, 0\right), \tag{A.6}$$

où $\lceil a\rceil \in \mathbb{Z}$ représente le plus petit entier supérieur à $a \in \mathbb{Q}$.

Autrement dit, en écrivant les séries de (A.5) comme suit :

$$\begin{aligned}\gamma^n\delta^t(\gamma^\nu\delta^\tau)^* &= \bigoplus_{i\geq 0}\gamma^{n+i\nu}\delta^{t+i\tau},\\ \gamma^{n'}\delta^{t'} \otimes \gamma^{K\nu'}\delta^{K\tau'}(\gamma^{\nu'}\delta^{\tau'})^* &= \bigoplus_{j\geq K}\gamma^{n'+j\nu'}\delta^{t'+j\tau'},\end{aligned}$$

il existe un entier K positif tel que l'inégalité (A.5) est vérifiée si, et seulement si :

$$x \in \mathbb{N},\ \forall x \geq K,\ \exists y \in \mathbb{N} \quad \text{tel que} \quad \begin{cases} n' + x\nu' \geq n + y\nu, \\ t' + x\tau' \leq t + y\tau. \end{cases}$$

Ce lemme est illustré par la figure A.3 et sur laquelle la valeur du plus petit K est 3.

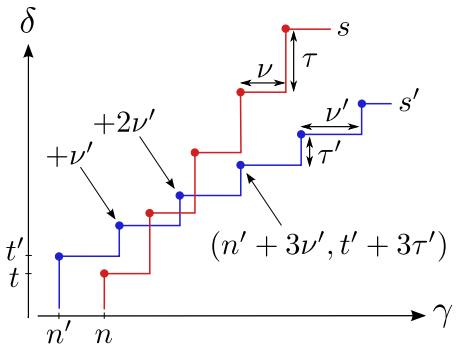


Figure A.3 Domination ultime de deux séries simples $s = \gamma^n \delta^t (\gamma^\nu \delta^\tau)^\star$ et $s' = \gamma^{n'} \delta^{t'} (\gamma^{\nu'} \delta^{\tau'})^\star$ de pentes différentes.

Remarque A.27 Notons que le K donné par l'expression (A.6) n'est pas nécessairement le plus petit entier positif satisfaisant la domination (A.5).

Théorème A.28 (Calcul et complexité algorithmique du produit de séries simples). Soient $s = \gamma^n \delta^t (\gamma^\nu \delta^\tau)^\star$ et $s' = \gamma^{n'} \delta^{t'} (\gamma^{\nu'} \delta^{\tau'})^\star$ deux séries simples de $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$. Le produit $\otimes$ de ces deux séries est une série périodique donnée comme suit selon les valeurs des pentes asymptotiques de s et s'.

· Lorsque $\sigma(s) < \sigma(s')$:

$$s \otimes s' = \gamma^{n+n'} \delta^{t+t'} (e \oplus \gamma^{\nu'} \delta^{\tau'} \oplus \ldots \oplus \gamma^{(K-1)\nu'} \delta^{(K-1)\tau'}) (\gamma^\nu \delta^\tau)^\star,$$

avec K donné par :

$$K = \max\left(\left\lceil \frac{\nu\tau}{\tau\nu' - \nu\tau'} \right\rceil, 0 \right).$$

La complexité de ce calcul est linéaire et dépend de la valeur K, soit une complexité en $\mathcal{O}(K)$.

· Lorsque $\sigma(s) = \sigma(s')$:

$$\begin{aligned} s \otimes s' &= \gamma^{n+n'}\delta^{t+t'}(e \oplus \gamma^{\nu}\delta^{\tau} \oplus \ldots \oplus \gamma^{(k-1)\nu}\delta^{(k-1)\tau}) \\ &\quad \otimes (e \oplus \gamma^{\nu'}\delta^{\tau'} \oplus \ldots \oplus \gamma^{(k'-1)\nu'}\delta^{(k'-1)\tau'})(\gamma^{\nu''}\delta^{\tau''})^{\star}, \end{aligned}$$

avec[1] $\nu'' = ppcm(\nu,\nu') = k\nu = k'\nu'$ et $\tau'' = ppcm(\tau,\tau') = k\tau = k'\tau'$. La complexité de ce calcul est en $\mathcal{O}(k'k\log(k))$.

Preuve.

· Lorsque $\sigma(s) < \sigma(s')$, selon le lemme A.26 et l'équation (A.5), il existe un entier K positif tel que :

$$\gamma^{K\nu'}\delta^{K\tau'}(\gamma^{\nu'}\delta^{\tau'})^{\star} \preccurlyeq (\gamma^{\nu}\delta^{\tau})^{\star} \quad \text{avec} \quad K = \max\left(\left\lceil \frac{\nu\tau}{\tau\nu' - \nu\tau'} \right\rceil, 0\right).$$

Ainsi, le produit de s par s' est donné par :

$$\begin{aligned} s \otimes s' &= \gamma^{n}\delta^{t}(\gamma^{\nu}\delta^{\tau})^{\star} \otimes \gamma^{n'}\delta^{t'}(\gamma^{\nu'}\delta^{\tau'})^{\star}, \\ &= \gamma^{n+n'}\delta^{t+t'} \otimes (\gamma^{\nu'}\delta^{\tau'})^{\star} \otimes (\gamma^{\nu}\delta^{\tau})^{\star}, \\ &= \gamma^{n+n'}\delta^{t+t'} \otimes \Big(e \oplus \gamma^{\nu'}\delta^{\tau'} \oplus \ldots \oplus \gamma^{(K-1)\nu'}\delta^{(K-1)\tau'} \\ &\qquad \oplus \gamma^{K\nu'}\delta^{K\tau'}(\gamma^{\nu'}\delta^{\tau'})^{\star}\Big) \otimes (\gamma^{\nu}\delta^{\tau})^{\star}, \\ &= \gamma^{n+n'}\delta^{t+t'} \otimes \Big(e \oplus \gamma^{\nu'}\delta^{\tau'} \oplus \ldots \oplus \gamma^{(K-1)\nu'}\delta^{(K-1)\tau'}\Big) \otimes (\gamma^{\nu}\delta^{\tau})^{\star}. \end{aligned}$$

La complexité de ce calcul repose sur le développement du polynôme $(e \oplus \gamma^{\nu'}\delta^{\tau'} \oplus \ldots \oplus \gamma^{(K-1)\nu'}\delta^{(K-1)\tau'})$ qui est réalisé en temps linéaire selon la valeur de K.

· Lorsque $\sigma(s) = \sigma(s')$, en posant $\nu'' = ppcm(\nu,\nu') = k\nu = k'\nu'$ et $\tau'' = ppcm(\tau,\tau') = k\tau = k'\tau'$, les étoiles des monômes $\gamma^{\nu}\delta^{\tau}$ et $\gamma^{\nu'}\delta^{\tau'}$ peuvent s'écrire comme suit :

$$\begin{aligned} (\gamma^{\nu}\delta^{\tau})^{\star} &= (e \oplus \gamma^{\nu}\delta^{\tau} \oplus \ldots \oplus \gamma^{(k-1)\nu}\delta^{(k-1)\tau})(\gamma^{\nu''}\delta^{\tau''})^{\star}, \\ (\gamma^{\nu'}\delta^{\tau'})^{\star} &= (e \oplus \gamma^{\nu'}\delta^{\tau'} \oplus \ldots \oplus \gamma^{(k'-1)\nu'}\delta^{(k'-1)\tau'})(\gamma^{\nu''}\delta^{\tau''})^{\star}. \end{aligned}$$

Le produit de séries simples de même pente s'exprime donc :

$$\begin{aligned} s \otimes s' &= \gamma^{n+n'}\delta^{t+t'} \otimes (e \oplus \gamma^{\nu}\delta^{\tau} \oplus \ldots \oplus \gamma^{(k-1)\nu}\delta^{(k-1)\tau}) \\ &\quad \otimes (e \oplus \gamma^{\nu'}\delta^{\tau'} \oplus \ldots \oplus \gamma^{(k'-1)\nu'}\delta^{(k'-1)\tau'}) \otimes (\gamma^{\nu''}\delta^{\tau''})^{\star}. \end{aligned}$$

1. *ppcm* : plus petit commun multiple.

La complexité de ce calcul repose essentiellement sur le produit des deux polynômes de $(\gamma^{\nu}\delta^{\tau})^{\star}$ et $(\gamma^{\nu'}\delta^{\tau'})^{\star}$. Selon le théorème A.25, cette complexité est donnée en $\mathcal{O}(k'k\log(k))$, avec $k = ppcm(\nu,\nu')/\nu$ et $k' = ppcm(\nu,\nu')/\nu'$.

□

Les complexités de ces produits ne sont pas réellement élevées : linéaire lorsque les pentes des séries sont différentes, et quasi-linéaire lorsqu'elles sont identiques. Cependant, elles dépendent de développements de polynômes jusqu'à une certaine valeur (valeur à partir de laquelle une série se retrouve graphiquement au-dessus d'une autre, ou encore résultats de calculs de *ppcm*), développements qui peuvent être arbitrairement longs et dont l'utilisateur n'en maîtrise pas les effets.

Annexe B

Librairie de calculs ContainerMinMaxGD

Sommaire

1 Présentation de la librairie 159
1.1 Description de la librairie 159
1.2 Quelques mot-clés et exemples de scripts 161
2 Expérimentations du conteneur 162
2.1 Première expérimentation 162
2.2 Seconde expérimentation 165
2.3 Complexité expérimentale 166

1 Présentation de la librairie

Le conteneur $\mathbf{f} \in \mathbf{F}$ présenté dans le chapitre 4 (cf. définition 4.21), ainsi que les algorithmes associés aux fonctions d'inclusion $[\oplus]$, $[*]$ et $^{[\star]}$ (cf. propositions 4.30, 4.34 et 4.36), ont été implémentés dans une librairie de calculs appelée **ContainerMinMaxGD**. Cet outil est disponible à l'adresse suivante : `http://www.istia.univ-angers.fr/~euriell.lecorronc/Recherche/softwares.php`, accompagné d'une documentation technique en anglais, et de fichiers d'exemples d'utilisation.

1.1 Description de la librairie

La librairie ContainerMinMaxGD permet de manipuler des conteneurs $\mathbf{f}$ de fonctions $f \in \mathcal{F}_{cp}$ tels que $f \in \mathbf{f} = [\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}}$. La saisie des données peut se faire de deux manières :

· lorsque le système est connu exactement, il est possible de saisir directement ses valeurs afin de créer le conteneur qui l'encadre,

· lorsque le système est incertain et n'est connu qu'à travers un intervalle, il faut alors saisir les valeurs des bornes connues afin que l'outil construise le conteneur.

Dans tous les cas, les approximations convexe $\mathcal{C}_{vx}$ et concave $\mathcal{C}_{cv}$ sont réalisées automatiquement lors de la construction du conteneur. Il est bien sûr nécessaire de respecter la définition 4.21 d'un conteneur $\mathbf{f} = [\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}}$ en proposant à la librairie des bornes $\underline{f}$ et $\overline{f}$ ayant la même pente asymptotique : $\sigma(\underline{f}) = \sigma(\overline{f})$, et les mêmes coordonnées de départ : $\Delta_{\tau_{\underline{f}}}^{\kappa_{\underline{f}}} = \Delta_{\tau}^{\kappa}{}_{\overline{f}}^{\overline{f}}$ (cela correspond à la définition 4.24 de la forme canonique d'un conteneur).

Afin d'obtenir une meilleure précision que celle obtenue avec des nombres réels, les points extrémaux des bornes des conteneurs sont définis comme des nombres rationnels codés grâce à la librairie Multi-Précision GNU (GMP). Ainsi, lorsque les résultats proposés ne sont pas entiers, ils sont affichés sous forme de fraction.

Ensuite, les opérations disponibles dans ContainerMinMaxGD sont les fonctions d'inclusion $[\circ] \in \{[\oplus], [*], {}^{[\star]}\}$ données dans les propositions 4.30, 4.34 et 4.36 telles que pour $\mathbf{f}, \mathbf{g} \in \mathbf{F}$:

$$\begin{cases} \mathbf{f}[\circ]\mathbf{g} \supset \mathbf{f} \circ \mathbf{g}, \\ \mathbf{f}[\circ]\mathbf{g} \in \mathbf{F}. \end{cases}$$

Il est également possible d'obtenir l'incertitude maximale $\Theta_{\mathbf{f}}$ d'un conteneur $\mathbf{f}$ selon la définition 4.26, soient les distances maximales en temps (délai) et en événement (backlog) entre les bornes du conteneur.

Enfin, un lien est réalisé entre la librairie ContainerMinMaxGD et celle appelée MinMaxGD (cf. [Cottenceau 2000] et annexe A.2). Ce lien permet notamment de disposer des opérations de quotient à droite $\not\phi$ et à gauche $\not\backslash$ afin de calculer les corrélations entre des fonctions de $\mathcal{F}_{cp}$ (cf. théorème 3.54). En effet, comme le précise la remarque A.17, puisque la librairie MinMaxGD propose de manipuler les séries formelles du dioïde $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ (cf. définition 2.37), elle peut être utilisée avec des fonctions de $\mathcal{F}_{cp}$ en transformant ces dernières en série de $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$. La librairie ContainerMinMaxGD offre donc la possibilité de créer un intervalle $\mathbf{i} = [\ \underline{i}\ ,\ \overline{i}\ ]$, avec $\underline{i}, \overline{i} \in \mathcal{F}_{cp}$ à partir d'un conteneur $\mathbf{f}$. Ces fonctions de $\mathcal{F}_{cp}$ sont ensuite implicitement transformées en séries de $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$. Afin de ne perdre aucune donnée, les bornes $\underline{i}$

et $\overline{i}$ de cet intervalle sont construites comme suit :

$$\begin{aligned} \underline{i} &\triangleq \min\{\ i \in \mathcal{F}_{cp} \mid i \succcurlyeq \underline{f}\ \}, \\ \overline{i} &\triangleq \max\{\ i \in \mathcal{F}_{cp} \mid i \preccurlyeq \overline{f}\ \}. \end{aligned}$$

Il s'agit donc de prendre la plus grande fonction entière de $\mathcal{F}_{cp}$ inférieure à $\underline{f}$ comme borne $\underline{i}$, et la plus petite fonction entière de $\mathcal{F}_{cp}$ supérieure à $\overline{f}$ comme borne $\overline{i}$.

1.2 Quelques mot-clés et exemples de scripts

Les éléments manipulés dans ContainerMinMaxGD sont donc des conteneurs $\mathbf{f} = [\ \underline{f}\ ,\ \overline{f}\ ]_{\mathcal{L}}$ dont les bornes $\underline{f}$ et $\overline{f}$ appartiennent respectivement aux ensembles $\mathcal{F}_{acv}$ et $\mathcal{F}_{acx}$. Il est également possible de produire des matrices de ces conteneurs. Les mots-clés représentant ces différents objets sont les suivants :

conteneur	`Cserie`
borne supérieure	`Fmaj`
borne inférieure	`Fmin`
matrice	`MCserie`

Voici quelques exemples d'utilisation de ces éléments.

```
// Déclaration des variables
Fmaj maj1, maj2, maj3
Fmin min1, min2, min3
Cserie c1, c2, c3, c4, c5
MCserie mc1, mc2, mc3, mc4, mc5

// Initialisation des bornes
maj1 = (1 0;5 2)[1]
min1 = (1 0;1 3;2 6)[1]
maj2 = (4 2)[2/3]
min2 = (4 2;5 5)[2/3]
maj3 = (2 2;8 5)
min3 = (2 2;5 5)

// Initialisation des conteneurs
c1 = ((0 0;0 3)[1],(0 0;5 4)[1])
c2 = (min1,maj1)
```

```
c3 = (2 2;3 4)
c4 = (0 0)
c5 = (eps)

// Initialisation des matrices de conteneurs
mc1(2,2) = ((0 0;0 3)[2],(0 0;5 4)[2]);
mc2(3,3) = (min3,maj3)
mc3(1,1) = (0 0)
mc4(2,2) = c4
mc5 = mc2

// Opérations disponibles sur les conteneurs et matrices
c2 = c1 + c3
Star(c2)
c4 = (min1+min2,maj1+maj2)
mc1(1,1) = c1 * c3
mc5 = Star(mc2)

// Exécution d'un fichier texte
Exec(nomFichier.txt)
```

Le lien avec la librairie MinMaxGD est quant à lui décrit plus en détail dans la documentation technique.

2 Expérimentations du conteneur

Trois tests de cette librairie ContainerMinMaxGD qui manipule des conteneurs de $\mathbf{F}$ sont maintenant proposés.

2.1 Première expérimentation

Le premier test a pour objectif de vérifier que les résultats obtenus en utilisant les fonctions exactes de $\mathcal{F}_{cp}$ sont toujours inclus dans les résultats des opérations effectuées sur les conteneurs.

Configuration.
Soit A une matrice carrée du dioïde $(\mathcal{F}_{cp}, \oplus, *)^{10\times 10}$ dont la plupart des entrées

sont la fonction ε et certaines autres sont égales à e ou à une simple fonction élémentaire Δ_T^K. Soit $\mathbf{A}$ une matrice carrée de $\mathbf{F}^{10\times 10}$ contenant A de manière garantie.

Remarque B.1 En réalité, les entrées de ces deux matrices A et $\mathbf{A}$ sont égales puisque la construction d'un conteneur $\mathbf{f}$ à partir d'une seule fonction élémentaire Δ_T^K fournit deux bornes identiques $\underline{f} = \overline{f}$ composées d'un seul point extrémal de coordonnées (T, K), et d'une pente asymptotique infinie, *i.e.* $\sigma(\underline{f}) = \sigma(\overline{f}) = +\infty$. Ainsi, $\mathcal{C}_{cv}(\Delta_T^K) = \mathcal{C}_{vx}(\Delta_T^K) = \Delta_T^K$.

Les matrices A et $\mathbf{A}$ sont donc remplies comme suit :

$$A = \mathbf{A} = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \Delta_{58}^1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \Delta_{61}^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \Delta_{81}^1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \Delta_{86}^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \Delta_{58}^1 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \Delta_{61}^1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \Delta_{35}^1 & \varepsilon & \varepsilon & \varepsilon & e \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \Delta_{36}^1 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \Delta_{69}^1 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \Delta_{69}^2 & \varepsilon \end{pmatrix}.$$

Remarque B.2 Cette matrice, issue d'un exemple traité dans [Olsder 1998], est un exemple de nombreuses interconnexions entre ses éléments lorsque son étoile de Kleene est calculée. Il faut entendre par là que même si la matrice A n'a pas beaucoup d'éléments différents de ε, la matrice $A^\star$ est quant à elle remplie de fonctions de $\mathcal{F}_{cp}$ dont les éléments Δ_T^K sont donnés avec $T \in [\ 0\ ,\ 975\ ]$ et $K \in [\ 0\ ,\ 21\ ]$.

Expérimentation.
Les matrices $A^\star$ et $\mathbf{A}^{[\star]}$ sont respectivement calculées avec MinMaxGD et ContainerMinMaxGD. Nous vérifions ensuite que chaque élément de $A^\star$ est effectivement inclus dans son conteneur, soit $(A^\star)_{ij} \in (\mathbf{A}^{[\star]})_{ij}$.

Remarque B.3 Ce calcul de l'étoile de Kleene est intéressant pour tester la librairie puisqu'il en utilise tous ses algorithmes. En effet, l'opération étoile de Kleene $^{[\star]}$ nécessite à elle seule de réaliser des sommes [$\oplus$], des produits [$*$] et le calcul d'étoiles de fonctions élémentaires $\Delta_T^{K\star}$.

Observations.
Tout d'abord, sur la même machine, le calcul de $A^\star$ prend 872 millisecondes tandis que celui de $\mathbf{A}^{[\star]}$ n'en prend que 70 ms, soit 10 fois moins. Ensuite, la vérification se déroule sans problème puisqu'effectivement :

$$(\underline{A}^{[\star]})_{ij} \preccurlyeq (A^\star)_{ij} \preccurlyeq (\overline{A}^{[\star]})_{ij}.$$

Par exemple, pour l'élément $i = 1$ et $j = 1$, la fonction $(A^\star)_{11}$ a pour valeur :

$$\begin{aligned}(A^\star)_{11} \quad = \quad & \Delta_0^0 \oplus \Delta_{286}^6 \oplus \Delta_{424}^9 \oplus \Delta_{476}^{10} \oplus \Delta_{524}^{11} \oplus \Delta_{572}^{12} \oplus \Delta_{614}^{13} \oplus \\ & (\Delta_{666}^{14} \oplus \Delta_{714}^{15} \oplus \Delta_{762}^{16} \oplus \Delta_{810}^{17} \oplus \Delta_{858}^{18} \oplus \Delta_{904}^{19}) * (\Delta_{286}^6)^\star.\end{aligned}$$

Les bornes $(\underline{A}^{[\star]})_{11}$ et $(\overline{A}^{[\star]})_{11}$ qui l'encadrent ont la même première fonction élémentaire que H :

$$\Delta^{\kappa_{(\underline{A}^{[\star]})_{11}}}_{\tau_{(\underline{A}^{[\star]})_{11}}} = \Delta^{\kappa_{(\overline{A}^{[\star]})_{11}}}_{\tau_{(\overline{A}^{[\star]})_{11}}} = \Delta_0^0,$$

et la même pente asymptotique donnée sous forme simplifiée :

$$\sigma((\underline{A}^{[\star]})_{11}) = \sigma((\overline{A}^{[\star]})_{11}) = 3/143.$$

Les points extrémaux de $(\underline{A}^{[\star]})_{11}$ et $(\overline{A}^{[\star]})_{11}$ sont donnés ci-dessous par couple (t, k) :

$$\begin{aligned}(\underline{A}^{[\star]})_{11} \quad &= \quad \{(0,0)\ ;\ (0^+, 6)\}, \\ (\overline{A}^{[\star]})_{11} \quad &= \quad \{(0,0)\}.\end{aligned}$$

D'après ces informations, nous pouvons dire que pour $t = 572$:

$$(\underline{A}^{[\star]})_{11}(572) = 18 \quad \preccurlyeq \quad (A^\star)_{11}(572) = 12 \quad \preccurlyeq \quad (\overline{A}^{[\star]})_{11}(572) = 12.$$

Donc la fonction $(A^\star)_{11}$ est effectivement contenue dans l'intervalle $[\ (\underline{A}^{[\star]})_{11}\ ,\ (\overline{A}^{[\star]})_{11}\ ]$. Enfin, pour cet exemple, l'incertitude maximale du conteneur $(\mathbf{A}^{[\star]})_{11}$ est donnée par :

$$\Theta_{(\mathbf{A}^{[\star]})_{11}} = \{\ D_{\max} = 286\ ,\ B_{\max} = 6\ \}.$$

2.2 Seconde expérimentation

Dans un second temps, il est intéressant de mesurer le gain d'espace mémoire entre l'utilisation de conteneurs de $\mathbf{F}$ et les fonctions exactes de $\mathcal{F}_{cp}$.

Configuration.
Soient A, B et C trois matrices appartenant respectivement à $(\mathcal{F}_{cp}, \oplus, *)^{4\times4}, (\mathcal{F}_{cp}, \oplus, *)^{4\times1}$ et $(\mathcal{F}_{cp}, \oplus, *)^{1\times4}$. Soient $\mathbf{A}$, $\mathbf{B}$ et $\mathbf{C}$ trois matrices appartenant respectivement à $\mathbf{F}^{4\times4}, \mathbf{F}^{4\times1}$ et $\mathbf{F}^{1\times4}$. Les matrices A, B, C et $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ sont données telles que :

$$A = \mathbf{A} = \begin{pmatrix} \Delta_{19}^{39} & \Delta_{29}^{45} & \Delta_{14}^{22} & \Delta_{40}^{20} \\ \Delta_{23}^{1} & \Delta_{5}^{27} & \Delta_{20}^{36} & \Delta_{1}^{12} \\ \Delta_{48}^{9} & \Delta_{46}^{43} & \Delta_{35}^{22} & \Delta_{32}^{39} \\ \Delta_{6}^{10} & \Delta_{27}^{27} & \Delta_{30}^{32} & \Delta_{9}^{32} \end{pmatrix}, \quad B = \mathbf{B} = \begin{pmatrix} e \\ \varepsilon \\ \varepsilon \\ \varepsilon \end{pmatrix} \text{ et } C = \mathbf{C} = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & e \end{pmatrix}.$$

Expérimentation.
Soient $H \in (\mathcal{F}_{cp}, \oplus, *)^{1\times1}$ et $\mathbf{H} \in \mathbf{F}^{1\times1}$ les deux matrices résultats des opérations :

$$\begin{aligned} H = C * A^{\star} * B \quad \text{et} \quad \mathbf{H} &= \mathbf{C}[*]\mathbf{A}^{[\star]}[*]\mathbf{B}, \\ &= [\ \underline{H}\ ,\ \overline{H}\ ]_{\mathcal{L}} = [\ \mathcal{C}_{cv}(\underline{H} \oplus \Omega(\overline{H}))\ ,\ \overline{H}\ ]_{\mathcal{L}}. \end{aligned}$$

La figure B.1 résume ces résultats obtenus avec MinMaxGD et ContainerMinMaxGD.

Observations.
Tout d'abord, notons que le calcul de H produit une matrice dont le seul élément a une combinaison $\oplus$ de 80 fonctions élémentaires Δ_T^K avant sa partie périodique. Voici quelques unes de ces fonctions élémentaires pendant le comportement transitoire et lors du régime périodique, ainsi que sa pente asymptotique :

$$H = \Delta_{6}^{10} \oplus \Delta_{50}^{28} \oplus \Delta_{52}^{40} \oplus \Delta_{78}^{41} \oplus \ldots \oplus (\Delta_{494}^{255} \oplus \ldots \oplus \Delta_{550}^{285}) * (\Delta_{62}^{31})^{\star}, \quad \sigma(H) = 31/62.$$

Concernant les éléments du conteneur $\mathbf{H}$, ses bornes $\underline{H}$ et $\overline{H}$ ont beaucoup moins de points que H. En effet, elles contiennent respectivement 4 et 3 points extrémaux donnés ci-dessous par leurs couples (t, k) :

$$\begin{aligned} \underline{H} &= \{(6, 10)\ ;\ (6^{+}, 28)\ ;\ (78, 2040803/28642)\ ;\ (148, 107)\}, \\ \overline{H} &= \{(6, 10)\ ;\ (50, 28)\ ;\ (78, 41)\}. \end{aligned}$$

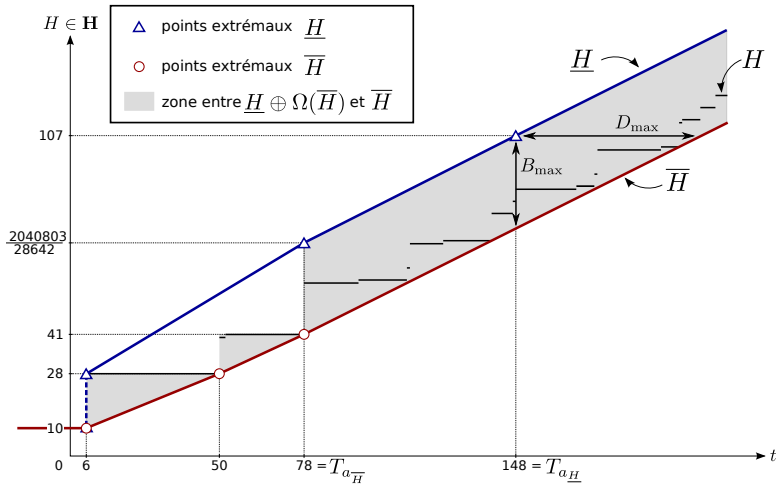


Figure B.1 Matrice $H = C * A^\star * B$ et son conteneur $\mathbf{H} = \mathbf{C}[*]\mathbf{A}^{[\star]}[*]\mathbf{B}$.

Par contre, $\underline{H}$, $\overline{H}$ et H ont la même première fonction élémentaire :

$$\Delta_{\tau_{\underline{H}}}^{\kappa_{\underline{H}}} = \Delta_{\tau_{\overline{H}}}^{\kappa_{\overline{H}}} = \Delta_{6}^{10},$$

et la pente asymptotique de $\mathbf{H}$ correspond à la forme réduite de celle de H :

$$\sigma(\underline{H}) = \sigma(\overline{H}) = 1/2.$$

L'incertitude maximale $\Theta_{\mathbf{H}}$ de $\mathbf{H}$ est également fournie par la librairie, à savoir :

$$\Theta_{\mathbf{H}} = \{\ D_{\max} = 62\ ,\ B_{\max} = 31\ \}.$$

Pour finir, on peut constater que sur la figure B.1, le système H appartient à la zone grise correspondant à la zone encadrée entre $\overline{H}$ et $\underline{H} \oplus \Omega(\overline{H})$, *i.e.* $\underline{H} \oplus \Omega(\overline{H}) \preccurlyeq H \preccurlyeq \overline{H}$.

2.3 Complexité expérimentale

Un dernier test est finalement réalisé afin de calculer la complexité expérimentale du calcul de l'étoile de Kleene d'une matrice carrée, en fonction de sa taille.

Configuration.

Soit $\mathbf{A}$ une matrice carrée du dioïde $\mathbf{F}^{n\times n}$ dont les entrées sont remplies par des fonctions élémentaires Δ_T^K, avec T et K des entiers aléatoirement choisis dans

l'intervalle $[1, 5]$, ou la fonction ε.

Expérimentation.
Le calcul de $\mathbf{A}^{[\star]}$ est réalisé 5 fois avec une taille de matrice évoluant entre 2×2 et 60×60, et les temps d'exécution moyen sont relevés.

Observations.
Les résultats présentant les valeurs moyennes des temps d'exécution de $\mathbf{A}^{[\star]}$ en fonction de la taille de $\mathbf{A}$ sont illustrés dans la figure B.2 (courbe la plus foncée). Ces temps sont compris entre une complexité en $\mathcal{O}(n^3)$ (due au calcul de l'étoile d'une matrice de taille $n \times n$) et une complexité en $\mathcal{O}(n^3 \log n)$.

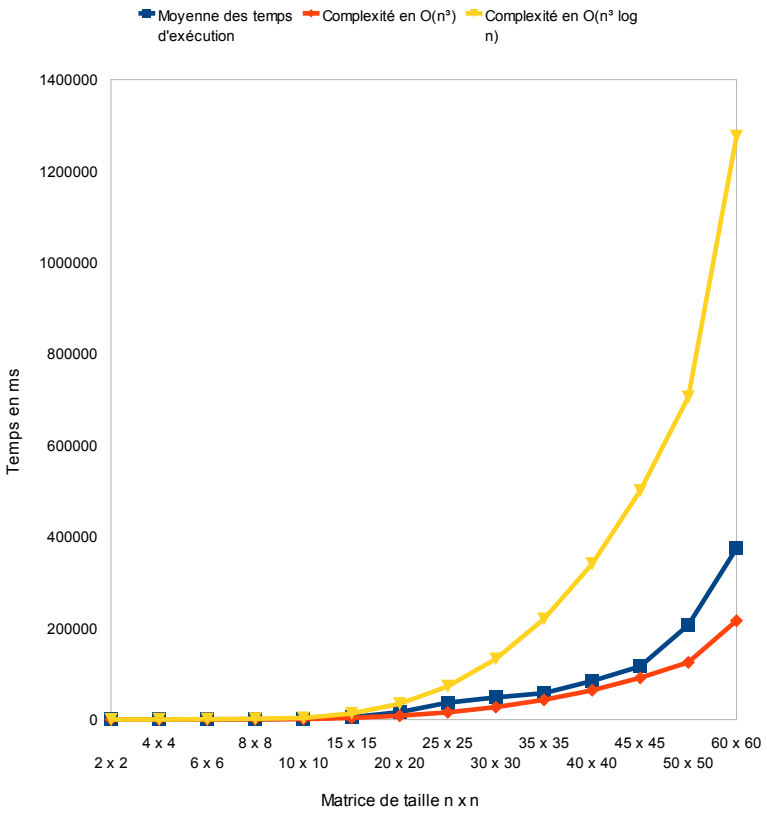


Figure B.2 Temps d'exécution du calcul de $\mathbf{A}^{[\star]}$ en fonction de la taille de $\mathbf{A}$.

Annexe C

Détails des calculs du chapitre 5

Sommaire

1 Réduction de l'incertitude grâce à un précompensateur . . 169
1.1 Calcul de l'incertitude maximale 169
1.2 Suppression de l'incertitude d'un système SISO 170
2 Précompensateur neutre optimal 171

Dans cette dernière annexe, nous proposons les scripts d'exécution qui ont permis d'obtenir les résultats des exemples d'application des sections 3 et 4 du chapitre 5.

1 Réduction de l'incertitude grâce à un précompensateur

Dans le cadre du problème de réduction de l'incertitude d'un système incertain $[\ \underline{h}\ ,\ \overline{h}\ ]$, deux scripts à exécuter avec la librairie MinMaxGD sont proposés.

1.1 Calcul de l'incertitude maximale

Les calculs de $(\overline{h}u)\backslash(\underline{h}u)$ et $\overline{h}\backslash\underline{h}$ sont donnés ci-dessous et permettent de montrer que pour toute entrée u, l'incertitude maximale du système est donnée par la corrélation $\overline{h}\backslash\underline{h}$ (cf. proposition 5.12). Les variables `hi` et `hs` correspondent aux bornes de l'intervalle $[\ \underline{h}\ ,\ \overline{h}\ ]$.

Script C.1 : Calcul des corrélations $(\overline{h}u)\backslash(\underline{h}u)$ et $\overline{h}\backslash\underline{h}$, cf. exemple 5.15.

```
hi = series([1 0],[4 1],[2 1])
hs = series([1 0;2 3],[4 5],[2 1])
u = series(eps,e,[2 3])

hiu = hi * u
```

```
hsu = hs * u

correlation1 = hsu \ hiu
correlation2 = hs \ hi
```

L'exécution de ce script fournit les résultats suivants :

$$\begin{array}{rcccl} \texttt{correlation1} & = & (\overline{h}u)\backslash(\underline{h}u) & = & \mathbf{\Delta^0_{-3}} \oplus \Delta^1_0 * (\Delta^2_1)^\star, \\ \texttt{correlation2} & = & \overline{h}\backslash\underline{h} & = & \Delta^0_{-4} * (\Delta^2_1)^\star \\ & & & = & \mathbf{\Delta^0_{-4}} \oplus \Delta^2_{-3} \oplus \Delta^4_{-2} \oplus \Delta^6_{-1} \oplus \mathbf{\Delta^8_0} * (\Delta^2_1)^\star. \end{array}$$

1.2 Suppression de l'incertitude d'un système SISO

Ensuite, l'algorithme du corollaire 2.62 est repris dans le script ci-dessous afin de calculer la valeur du précompensateur optimal proposé dans la section 3 du chapitre 5. Ce précompensateur permet ainsi de supprimer l'incertitude en sortie du système contrôlé.

SCRIPT C.2 : Calcul du précompensateur optimal $\hat{p}$ permettant de supprimer l'incertitude, cf. section 3.4 du chapitre 5.

```
hi = series(eps,[0 8],[4 5]) + series(eps,[35 49;45 64],[4 6])
hs = series(eps,[0 8],[4 5]) + series(eps,[15 24;25 39],[4 6])
val = prcaus(hi\hi)
p = s_top
pointFixe = %F;

while pointFixe == %F
  pAvant = p;
  p = ((hi*p)/hs) ^ prcaus(p) ^ val
  if pAvant == p
    pointFixe = %T;
  end
end
```

```
pOptimal = p
hip = hi * pOptimal
hsp = hs * pOptimal
correlation = hsp \ hip
```

Les résultats obtenus en quatre itérations sont les suivants :

$$\begin{array}{rcccl} \texttt{pOptimal} & = & \hat{p} & = & \Delta_1^4 \oplus \ldots \oplus \Delta_{62}^{49} * (\Delta_3^2)^\star, \\ \texttt{hip = hsp} & = & \underline{h}\hat{p} = \overline{h}\hat{p} & = & \Delta_9^4 \oplus \ldots \oplus \Delta_{70}^{49} * (\Delta_3^2)^\star, \\ \texttt{correlation} & = & (\underline{h}\hat{p}) \not\oslash (\overline{h}\hat{p}) & = & \mathbf{\Delta_0^0} \oplus \ldots \oplus \Delta_{37}^{29} * (\Delta_3^2)^\star. \end{array}$$

2 Précompensateur neutre optimal

Le calcul du précompensateur neutre optimal donné en tant qu'exemple d'application de la section 4 du chapitre 5 est proposé ci-dessous. Ce script est à exécuter avec la librairie MinMaxGD. Les variables `Hi` et `Hs` sont les matrices contenant respectivement les bornes inférieures et supérieures des éléments de $\mathbf{H}$, ainsi `Hi` $= (\ \underline{H}_{11}\ \ \underline{H}_{12}\ )$ et `Hi` $= (\ \overline{H}_{11}\ \ \overline{H}_{12}\ )$. La matrice `I` correspond quant à elle à la matrice identité E.

Script C.3 : Calcul du précompensateur neutre optimal $\hat{p}$, cf. section 4.4 du chapitre 5.

```
Hi(1,1)=series([0 1],[1 2],[3 2]);
Hi(1,2)=series([0 1],[2 2],[2 2])
Hs(1,1)=series([0 3;1 4],[2 5],[3 2]);
Hs(1,2)=series(e,[0 2],[1 1])
I(1,1)=series(e);
I(2,2)=series(e)

pOpt = I + (Hs \ Hi);
pOptCaus = prcaus(pOpt)
```

L'exécution de ce script fournit le résultat suivant :

$$\begin{aligned}
\texttt{pOptCaus}(1,1) &= \Delta_0^0 \oplus (\Delta_1^6 \oplus \Delta_2^8) * (\Delta_2^3)^\star, \\
\texttt{pOptCaus}(1,2) &= \Delta_0^4 * (\Delta_1^1)^\star, \\
\texttt{pOptCaus}(2,1) &= \varepsilon, \\
\texttt{pOptCaus}(2,2) &= \Delta_0^0 \oplus \Delta_1^4 * (\Delta_1^1)^\star.
\end{aligned}$$

avec $\texttt{pOptCaus} = \hat{p} = \mathsf{Pr}_{\mathsf{caus}}(E \oplus \overline{H} \backslash\!\!\!\backslash \underline{H})$.

Bibliographie personnelle

Revues internationales avec comité de lecture

[Hardouin 2009] L. Hardouin, B. Cottenceau, M. Lhommeau et E. Le Corronc. *Interval systems over idempotent semiring.* Linear Algebra and Its Applications, vol. 431, no. 5-7, pages 855–862, 2009.

[Le Corronc 2011a] E. Le Corronc, B. Cottenceau et L. Hardouin. *Container of (min,+)-linear systems.* Journal of Discrete Event Dynamic Systems, 2011. En cours de révision.

Conférences internationales avec acte et comité de lecture

[Hardouin 2008] L. Hardouin, B. Cottenceau, S. Lagrange et E. Le Corronc. *Performance analysis of linear systems over semiring with additive inputs.* In 9th International Workshop on Discrete Event Systems, WODES'08, pages 43–48, 2008.

[Le Corronc 2009b] E. Le Corronc, B. Cottenceau et L. Hardouin. *Control of uncertain (min,+)-linear systems.* In 3rd International Symposium on Positive Systems : Theory and Applications, POSTA'09, 2009.

[Le Corronc 2010a] E. Le Corronc, B. Cottenceau et L. Hardouin. *Control of uncertain (max,+)-linear systems in order to decrease uncertainty.* In 10th International Workshop on Discrete Event Systems, WODES'10, 2010.

[Le Corronc 2010b] E. Le Corronc, B. Cottenceau et L. Hardouin. *Flow control with (min,+) algebra.* In 4th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation, Special Session Worst Case Traversal Time, ISOLA'10, 2010.

Conférences nationales avec acte et comité de lecture

[Le Corronc 2009a] E. Le Corronc, B. Cottenceau et L. Hardouin. *Approximation convexe de systèmes (max,+)-linéaires.* In Journées Doctorales du GDR MACS, JDMACS'09, 2009.

[Le Corronc 2009c] E. Le Corronc, B. Cottenceau et L. Hardouin. *Encadrement de systèmes (min,+)-linéaires*. In 7ème colloque francophone sur la Modélisation des Systèmes Réactifs, MSR'09, 2009.

[Cottenceau 2009] B. Cottenceau, L. Hardouin et E. Le Corronc. *Représentation tridimensionnelle de la dynamique des graphes d'événements temporisés généralisés*. In 7ème colloque francophone sur la Modélisation des Systèmes Réactifs, MSR'09, 2009.

[Le Corronc 2011b] E. Le Corronc, B. Cottenceau et L. Hardouin. *Contrôle de flux : problème du Window Flow Control*. In Journées Nationales du GDR Génie de la Programmation et du Logiciel, GPL'11, 2011.

Bibliographie

[Agrawal 1999] R. Agrawal, R.L. Cruz, C. Okino et R. Rajan. *Performance bonds for flow control protocols.* IEEE/ACM Transactions on Networking (TON), vol. 7, no. 3, pages 310–323, 1999.

[Baccelli 1992] F. Baccelli, G. Cohen, G.J. Olsder et J.-P. Quadrat. Synchronization and linearity : an algebra for discrete event systems. Wiley and sons, 1992. http://cermics.enpc.fr/~cohen-g/documents/BCOQ-book.pdf.

[Blyth 2005] T.S. Blyth. Lattices and ordered algebraic structures. Springer Verlag, 2005.

[Bouillard 2007] A. Bouillard, B. Gaujal, S. Lagrange et E. Thierry. *Optimal routing for end-to-end guarantees : the price of multiplexing.* In Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools, ValueTools'07, pages 1–10. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007.

[Bouillard 2008a] A. Bouillard, L. Jouhet et E. Thierry. *Computation of a (min,+) multi-dimensional convolution for end-to-end performance analysis.* In Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools, ValueTools'08, pages 1–7. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[Bouillard 2008b] A. Bouillard et E. Thierry. *An algorithmic toolbox for network calculus.* Journal of Discrete Event Dynamic Systems, vol. 18, no. 1, pages 3–49, 2008.

[Bouillard 2009] A. Bouillard, B. Cottenceau, B. Gaujal, L. Hardouin, S. Lagrange et M. Lhommeau. *COINC library : a toolbox for the network calculus.* In Proceedings of the 4th International Conference on Performance Evaluation Methodologies and Tools, ValueTools'09. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

[Boutin 2008] O. Boutin, B. Cottenceau et A. L'Anton. *Dealing with mutual exclusion sections in production systems : from shared resources to parallel TEG's.* In 17th International Federation of Automatic Control World Congress, IFAC'08, 2008.

[Boutin 2009] O. Boutin, B. Cottenceau, A. L'Anton et J.J. Loiseau. *Modelling systems with periodic routing functions in dioid (min,+).* In 13th IFAC

Symposium on Information Control Problems in Manufacturing, INCOM'09, 2009.

[Boyer 2008] M. Boyer et C. Fraboul. *Tightening end to end delay upper bound for AFDX network with rate latency FCFS servers using network calculus.* In 7th IEEE International Workshop on Factory Communication Systems Communication in Automation, WFCS'08, pages 11–20, 2008.

[Chakraborty 2003] S. Chakraborty, S. Künzli et L. Thiele. *A general framework for analysing system properties in platform-based embedded system designs.* In Proceedings of the 6th Conference on Design, Automation and Test in Europe, DATE'03, pages 190–195. Citeseer, 2003.

[Chakraborty 2006] S. Chakraborty, Y. Liu, N. Stoimenov, L. Thiele et E. Wandeler. *Interface-based rate analysis of embedded systems.* In 27th IEEE International Real-Time Systems Symposium, RTSS'06, pages 25–34. IEEE, 2006.

[Chang 2000] C.S. Chang. Performance guarantees in communication networks. Springer, 2000.

[Chang 2002] C.S. Chang, R.L. Cruz, J.-Y. Le Boudec et P. Thiran. *A min,+ system theory for constrained traffic regulation and dynamic service guarantees.* IEEE/ACM Transactions on Networking, vol. 10, no. 6, pages 805–817, 2002.

[Cohen 1989] G. Cohen, P. Moller, J.-P. Quadrat et M. Viot. *Algebraic tools for the performance evaluation of discrete event systems.* Proceedings of the IEEE, vol. 77, no. 1, pages 39–85, 1989.

[Cohen 1993] G. Cohen, S. Gaubert et J.-P. Quadrat. *From first to second-order theory of linear discrete event systems.* In 12th World IFAC Congress, IFAC'93, 1993.

[Cottenceau 1999a] 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. http://www.istia.univ-angers.fr/LISA/THESES/theses.html.

[Cottenceau 1999b] B. Cottenceau, L. Hardouin, J.-L. Boimond et J.-L. Ferrier. *Synthesis of greatest linear feedback for timed-event graphs in dioid.* IEEE Transactions on Automatic Control, vol. 44, no. 6, pages 1258–1262, 1999.

[Cottenceau 2000] B. Cottenceau, M. Lhommeau, L. Hardouin et J.-L. Boimond. *Data processing tool for calculation in dioid.* In 5th International Workshop on Discrete Event Systems, WODES'00, 2000. http://www.istia.univ-angers.fr/~hardouin/.

[Cottenceau 2001] B. Cottenceau, L. Hardouin, J.L. Boimond et J.-L. Ferrier. *Model reference control for timed event graphs in dioids.* Automatica, vol. 37, no. 9, pages 1451–1458, 2001.

[Cottenceau 2006] B. Cottenceau, L. Hardouin et M. Lhommeau. *MinMaxGD, une librairie de calculs dans* $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$. Rapport technique, 1998-2006. `http://www.istia.univ-angers.fr/~hardouin/`.

[Cottenceau 2009] B. Cottenceau, L. Hardouin et E. Le Corronc. *Représentation tridimensionnelle de la dynamique des graphes d'événements temporisés généralisés.* In 7ème colloque francophone sur la Modélisation des Systèmes Réactifs, MSR'09, 2009.

[Cruz 1991a] R.L. Cruz. *A calculus for network delay. I. Network elements in isolation.* IEEE Transactions on Information Theory, vol. 37, no. 1, pages 114–131, 1991.

[Cruz 1991b] R.L. Cruz. *A calculus for network delay. II. Network analysis.* IEEE Transactions on Information Theory, vol. 37, no. 1, pages 132–141, 1991.

[Cuninghame-Green 1979] R.A. Cuninghame-Green. Minimax algebra, volume 166 of *Lecture notes in Economics and Mathematical Systems.* Springer, 1979.

[Di Loreto 2009] M. Di Loreto, S. Gaubert, R.D. Katz et J.J. Loiseau. *Duality between invariant spaces for max-plus linear discrete event systems.* Eprint arXiv, vol. 901, 2009.

[Fidler 2006] M. Fidler et S. Recker. *Conjugate network calculus : a dual approach applying the Legendre transform.* Computer Networks, vol. 50, no. 8, pages 1026–1039, 2006.

[Fidler 2010] M. Fidler. *Survey of deterministic and stochastic service curve models in the network calculus.* IEEE Communications Surveys & Tutorials, vol. 12, no. 1, pages 59–86, 2010.

[Frances 2006] F. Frances, C. Fraboul et J. Grieu. *Using network calculus to optimize the AFDX network.* In 3rd European Congress Embedded Real-Time Software, ERTS'06, 2006.

[Gallot 1997] F. Gallot, J.-L. Boimond et L. Hardouin. *Identification of simple elements in max-algebra : application to SISO discrete event systems modelisation.* In Proceedings of the 4th European Control Conference, ECC'97, 1997.

[Gaubert 1992] S. Gaubert. *Théorie des systèmes linéaires dans les dioïdes*. PhD thesis, INRIA - Ecole des Mines de Paris, 1992. http://amadeus.inria.fr/gaubert/PAPERS/ALL.pdf.

[Gondran 1984] M. Gondran et M. Minoux. *Linear algebra in dioids : a survey of recent results*. In Algebraic and combinatorial methods in operations research : proceedings of the Workshop on Algebraic Structures in Operations Research, page 147. North-Holland, 1984.

[Graham 1972] R.L. Graham. *An efficient algorithm for determining the convex hull of a finite planar set*. Information Processing Letters, vol. 1, no. 4, pages 132–133, 1972.

[Grieu 2004] J. Grieu. *Analyse et évaluation de techniques de commutation Ethernet pour l'interconnexion des systèmes avioniques*. PhD thesis, Institut National Polytechnique de Toulouse, 2004. http://ethesis.inp-toulouse.fr/archive/00000084/01/grieu.pdf.

[Gruet 1995] B. Gruet. Étude d'une structure de commande en boucle fermée. Master's thesis, LISA - Université d'Angers, 1995. http://www.istia.univ-angers.fr/LISA/DEA/dea.html.

[Hardouin 1997] L. Hardouin, E. Menguy, J.-L. Boimond et J.-L. Ferrier. *SISO discrete event systems control in dioid algebra*. Journal européen des systèmes automatisés, vol. 31, no. 3, pages 433–452, 1997.

[Hardouin 2004] L. Hardouin. *Sur la commande linéaire de systèmes à événements discrets dans l'algèbre (max,+)*. Habilitation à Diriger des Recherches, LISA - Université d'Angers, 2004. http://www.istia.univ-angers.fr/~hardouin/publications.html.

[Hardouin 2008] L. Hardouin, B. Cottenceau, S. Lagrange et E. Le Corronc. *Performance analysis of linear systems over semiring with additive inputs*. In 9th International Workshop on Discrete Event Systems, WODES'08, pages 43–48, 2008.

[Hardouin 2009] L. Hardouin, B. Cottenceau, M. Lhommeau et E. Le Corronc. *Interval systems over idempotent semiring*. Linear Algebra and Its Applications, vol. 431, no. 5-7, pages 855–862, 2009.

[Hardouin 2010a] L. Hardouin, C.A. Maia, B. Cottenceau et M. Lhommeau. *Observer design for (max,plus) linear system*. IEEE Transactions on Automatic Control, vol. 55, no. 2, pages 538–543, 2010.

[Hardouin 2010b] L. Hardouin, C.A. Maia, B. Cottenceau et R. Santos-Mendes. *Max-plus linear observer : application to manufacturing systems.* In 10th International Workshop on Discrete Event Systems, WODES'10, 2010.

[Heidergott 2006] B. Heidergott, G.J. Olsder et J. Woude. Max plus at work, modeling and analysis of synchronized systems : a course on max-plus algebra and its applications. Princeton University Press, 2006.

[Houssin 2004] L. Houssin, S. Lahaye et J.-L. Boimond. *Modelling and control of urban bus networks in dioids algebra.* In 9th International Workshop on Discrete Event Systems, WODES'04, 2004.

[Jaulin 2001] L. Jaulin, M. Kieffer, O. Didrit et E. Walter. Applied interval analysis. Springer London, 2001.

[Katz 2007] R.D. Katz. *Max-plus (A,B)-invariant spaces and control of timed discrete-event systems.* IEEE Transactions on Automatic Control, vol. 52, no. 2, pages 229–241, 2007.

[Le Boudec 2001] J.-Y. Le Boudec et P. Thiran. Network calculus : a theory of deterministic queuing systems for the internet. Springer, 2001. http://ica1www.epfl.ch/PS_files/netCalBookv4.pdf.

[Le Corronc 2009a] E. Le Corronc, B. Cottenceau et L. Hardouin. *Approximation convexe de systèmes (max,+)-linéaires.* In Journées Doctorales du GDR MACS, JDMACS'09, 2009.

[Le Corronc 2009b] E. Le Corronc, B. Cottenceau et L. Hardouin. *Control of uncertain (min,+)-linear systems.* In 3rd International Symposium on Positive Systems : Theory and Applications, POSTA'09, 2009.

[Le Corronc 2009c] E. Le Corronc, B. Cottenceau et L. Hardouin. *Encadrement de systèmes (min,+)-linéaires.* In 7ème colloque francophone sur la Modélisation des Systèmes Réactifs, MSR'09, 2009.

[Le Corronc 2010a] E. Le Corronc, B. Cottenceau et L. Hardouin. *Control of uncertain (max,+)-linear systems in order to decrease uncertainty.* In 10th International Workshop on Discrete Event Systems, WODES'10, 2010.

[Le Corronc 2010b] E. Le Corronc, B. Cottenceau et L. Hardouin. *Flow control with (min,+) algebra.* In 4th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation, Special Session Worst Case Traversal Time, ISOLA'10, 2010.

[Le Corronc 2011a] E. Le Corronc, B. Cottenceau et L. Hardouin. *Container of (min,+)-linear systems.* Journal of Discrete Event Dynamic Systems, 2011. En cours de révision.

[Le Corronc 2011b] E. Le Corronc, B. Cottenceau et L. Hardouin. *Contrôle de flux : problème du Window Flow Control.* In Journées Nationales du GDR Génie de la Programmation et du Logiciel, GPL'11, 2011.

[Lenzini 2008] L. Lenzini, E. Mingozzi et G. Stea. *A methodology for computing end-to-end delay bounds in FIFO-multiplexing tandems.* Performance Evaluation, vol. 65, no. 11-12, pages 922–943, 2008.

[Lhommeau 2003a] M. Lhommeau. *Étude de systèmes à événements discrets dans l'algèbre (max,+) : 1. Synthèse de correcteurs robustes dans un dioïde d'intervalles. 2. Synthèse de correcteurs en présence de perturbations.* PhD thesis, LISA - Université d'Angers, 2003. http://www.istia.univ-angers.fr/LISA/THESES/theses.html.

[Lhommeau 2003b] M. Lhommeau, L. Hardouin et Cottenceau B. *Optimal control for (max,+)-linear systems in the presence of disturbances.* In International Symposium on Positive Systems : Theory and Applications, POSTA'03, 2003.

[Lhommeau 2004a] M. Lhommeau, L. Hardouin, B. Cottenceau et L. Jaulin. *Interval analysis and dioid : application to robust controller design for timed event graphs.* Automatica, vol. 40, no. 11, pages 1923–1930, 2004.

[Lhommeau 2004b] M. Lhommeau, L. Hardouin, C.A. Maia et R. Santos-Mendes. *Control and robustness analysis for (max,+)-linear systems.* In 9th International Workshop on Discrete Event Systems, WODES'04, 2004.

[Lhommeau 2005] M. Lhommeau, L. Hardouin, J.-L. Ferrier et I. Ouerghi. *Interval analysis in dioid : application to robust open-loop control for timed event graphs.* In 44th IEEE Conference on Decision and Control and European Control Conference, CDC-ECC'05, pages 7744–7749, 2005.

[Libeaut 1996] L. Libeaut et J.-J. Loiseau. *On the control of timed event graphs.* In 3rd International Workshop on Discrete Event Systems, WODES'96, pages 159–164, 1996.

[Litvinov 2001] G.L. Litvinov et A.N. Sobolevskiĭ. *Idempotent interval analysis and optimization problems.* Reliable Computing, vol. 7, no. 5, pages 353–377, 2001.

[Luders 2002] R. Luders et R. Santos-Mendes. *Generalized multivariable control of discrete event systems in dioids.* In 6th International Workshop on Discrete Event Systems, WODES'02, pages 197–202. IEEE, 2002.

[Maia 2003a] C.A. Maia, L. Hardouin, R. Santos-Mendes et B. Cottenceau. *Optimal closed-loop control of timed event graphs in dioids.* IEEE Transactions on Automatic Control, vol. 48, no. 12, pages 2284–2287, 2003.

[Maia 2003b] C.A. Maia, R. Santos-Mendes et L. Hardouin. *Some Results on Identification of Timed Event Graphs in Dioid.* In 11th IEEE Mediterranean Conference on Control and Automation, MED'03, 2003.

[Maia 2011] C.A. Maia, L. Hardouin, R. Santos-Mendes et J.-J.s Loiseau. *A super-eigenvector approach to control constrained max-plus linear systems.* In 50th IEEE Conference on Decision and Control and European Control Conference, CDC-ECC'11, 2011.

[MaxPlus 1991] MaxPlus. *Second order theory of min-linear systems and its application to discrete event systems.* In Proceedings of the 30th IEEE Conference on Decision and Control, CDC'91, pages 1511–1516, December 1991.

[Moore 1979] R.E. Moore. Methods and applications of interval analysis. SIAM, 1979.

[Olsder 1998] G.J. Olsder, Subiono et M. Mac Gettrick. *On large scale max-plus algebra model in railway systems.* In Algèbres Max-Plus et applications en informatique et automatique, École de printemps d'informatique théorique, Noirmoutier, 1998.

[Ouerghi 2006a] I. Ouerghi et L. Hardouin. *Control synthesis for P-temporal event graphs.* In 8th International Workshop on on Discrete Event Systems, WODES'06, pages 229–234, 2006.

[Ouerghi 2006b] I. Ouerghi et L. Hardouin. *A precompensator synthesis for P-temporal event graphs.* In International Symposium on Positive Systems : Theory and Applications, POSTA'06, pages 391–398, 2006.

[Pandit 2006] K. Pandit, J. Schmitt, C. Kirchner et R. Steinmetz. *A transform for network calculus and its application to multimedia networking.* In Proceedings of SPIE on Multimedia Computing and Networking, MMCN'06, volume 6071, pages 103–114, 2006.

[Parekh 1993] A.K. Parekh et R.G. Gallager. *A generalized processor sharing approach to flow control in integrated services networks : the single-node case.* IEEE/ACM Transactions on Networking (TON), vol. 1, no. 3, pages 344–357, 1993.

[Parekh 1994] A.K. Parekh et R.G. Gallager. *A generalized processor sharing approach to flow control in integrated services networks : the multiple node case.* IEEE/ACM Transactions on Networking, vol. 2, no. 2, pages 137–150, 1994.

[Rockafellar 1997] R.T. Rockafellar. Convex analysis. Princeton University Press, 1997.

[Schioler 2007] H. Schioler, H.P. Schwefel et M.B. Hansen. *CyNC : a MATLAB/SimuLink toolbox for network calculus.* In Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools, ValueTools'07, pages 1–10. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007.

[Schmitt 2005] J.B. Schmitt et U. Roedig. *Sensor network calculus–A framework for worst case analysis.* In Proceedings of IEEE/ACM International Conference on Distributed Computing in Sensor Systems, DCOSS'05, pages 141–154. Springer, 2005.

[Schmitt 2006] J.B. Schmitt et F.A. Zdarsky. *The disco network calculator : a toolbox for worst case analysis.* In Proceedings of the 1st international conference on Performance Evaluation Methodologies and Tools, ValueTools'06. ACM, 2006.

[Shang 2011] Y. Shang et M.K. Sain. *Fixed zeros in the model matching problem for systems over semirings.* Linear Algebra and its Applications, vol. 434-1, pages 18–43, 2011.

[Tarski 1955] A. Tarski. *A lattice-theoretical fixpoint theorem and its applications.* Pacific journal of Mathematics, vol. 5, no. 2, pages 285–309, 1955.

[Wandeler 2006] E. Wandeler et L. Thiele. *Real-Time Calculus (RTC) Toolbox.* 2006.

[Wonham 1987] W.M. Wonham et P.J. Ramadge. *On the supremal controllable sublanguage of a given language.* SIAM Journal on Control and Optimization, vol. 25, page 637, 1987.

Index

Application résiduable et sa résiduée, **p.31**
Approximation concave, **p.88**, **p.94**
Approximation convexe, **p.86**, **p.94**

Backlog, **p.72**

Causalité, **p.50**
Clôture sous-additive, **p.37**
Conteneur de systèmes (min,+)-linéaire, **p.89**
Contrainte sur la sortie, **p.66**
Corrélation et autocorrélation, **p.68**
Courbe d'arrivée, **p.62**
Courbe de contrainte, **p.60**
Courbe de service, **p.64**

Déconvolution, **p.57**
Délai, **p.72**
Dioïde, **p.16**
Dioïde complet, **p.20**
Dioïde de fonctions, **p.22**, **p.48**, **p.51**, **p.56**
Dioïde de matrices, **p.22**
Dioïde de séries formelles, **p.24**
Dioïde quotient, **p.26**, **p.82**

Ensemble des points fixes d'application, **p.35**
Etoile de Kleene, **p.37**

Fonction élémentaire, **p.52**
Fonction compteur, **p.45**
Fonction croissante, **p.56**
Fonction croissante et constante par morceaux $\mathcal{F}_c$, **p.46**
Fonction croissante, affine et ultimement affine, **p.83**
Fonction croissante, constante par morceaux et ultimement pseudo-périodique, **p.50**
Fonction d'inclusion, **p.93**, **p.95, 96**, **p.98**
Fonction de transfert, **p.49**
Fonction sous-additive, **p.85**
Forme canonique, **p.53**, **p.91**, **p.151, 152**

Graphe d'Événements Temporisés (GET), **p.3**, **p.44**

Injection canonique, **p.34**
Isotonie, antitonie, monotonie, **p.29**

Pente asymptotique, **p.50**, **p.83**
Point extrémal, **p.84**
Précompensateur, **p.108**, **p.120**, **p.127**
Projecteur, **p.34**
Projection causale, **p.54**
Pseudo-périodicité, **p.50**

Quotient à gauche et à droite, **p.32**

Relation d'ordre, **p.17**
Relation d'ordre d'un dioïde, **p.19**
Relation entrée/sortie, **p.3**, **p.48**
Représentation d'état, **p.3**, **p.47**
Retour de sortie, **p.110**

Semi-continuité inférieure, semi-continuité supérieure, continuité, **p.30**
Stocks et temps de séjour, **p.70**
Système (min,+)-linéaire, **p.49**
Système rebouclé, **p.51**
Systèmes mis en parallèle, **p.51**
Systèmes mis en série, **p.51**

Transformée de Legendre-Fenchel, **p.81**

Window Flow Control, **p.113, 114**

Modèles et calculs garantis pour les systèmes (min,+)-linéaires

Résumé : Cette thèse se situe dans le contexte des systèmes (min,+)-linéaires incertains, contenus dans des intervalles dont seules les bornes sont connues.

Dans un premier temps, puisque les calculs exacts – somme, inf-convolution, étoile de Kleene/clôture sous-additive – sont souvent coûteux en espace mémoire et en temps de calculs, des calculs approchés sont proposés à travers des fonctions d'inclusion. Les résultats obtenus sont des intervalles particuliers appelés conteneurs. La borne supérieure d'un conteneur est l'élément le plus grand de la classe d'équivalence du système approché selon la transformée de Legendre-Fenchel. La borne inférieure représente un minorant de cette classe d'équivalence et ainsi du système exact. Les caractéristiques de convexité de ces bornes permettent d'obtenir des algorithmes de calculs de complexité linéaire et quasi-linéaire.

Dans un second temps, la commande des systèmes (min,+)-linéaires incertains est étudiée. Ces incertitudes peuvent apparaître suite aux calculs effectués sur les conteneurs proposés ci-dessus, mais également lorsque des paramètres incertains ou variables sont présents lors des modélisations. Des structures de contrôles existantes (précompensateur, retour de sortie) sont appliquées à ces systèmes incertains et les problèmes suivants sont ainsi traités : problème du Window Flow Control, réduction de l'incertitude en sortie du système contrôlé par un précompensateur, calcul d'un précompensateur neutre ralentissant les entrées sans dégrader la dynamique du système seul.

Mots-clés : Systèmes à Événements Discrets, Algèbres (min,+) et (max,+), Network Calculus, Contrôle de systèmes (min,+)-linéaires, Window Flow Control

Guaranteed models and computations for (min,+)-linear systems

Abstract : This thesis deals with uncertain (min,+)-linear systems contained in intervals, the bounds of which are known.

First, as exact computations – sum, inf-convolution, Kleene star/subadditive closure – are often memory consuming and time costly, some approached computations are introduced through inclusion functions. The obtained results are particular intervals called container. The upper bound of a container is the greatest element of the equivalence class of the approached system modulo the Legendre-Fenchel transform. The lower bound of the container is a lower bound of this class, and so of the approached system. The convex characteristics of these bounds provide algorithms of linear or quasi-linear complexity for the computations.

Second, the control of uncertain (min,+)-linear systems is studied. These uncertainties can appear after computations realized over the containers introduced above, or when uncertain or variable parameters are present during the modelings. Some existing control structures (precompensator, feedback) are applied over these uncertain systems and the following problems are treated: Window Flow Control problem, reduction of the uncertainty at the output of the system controlled by a precompensator, computation of a neutral precompensator which delays the input without damage the transfer of the system alone.

Keywords : Discrete Event Systems, (Min,+) and (max,+) algebras, Network Calculus, Control of (min,+)-linear systems, Window Flow Control