

Sur l'estimation d'état des systèmes max-plus

Guilherme Espindola-Winck¹, Laurent Hardouin¹, and Mehdi Lhommeau¹

Laboratoire Angevin de Recherche en Ingénierie des Systèmes, Angers, France
{guilherme.espindolawinck, laurent.hardouin, mehdi.lhommeau}@univ-angers.fr

Abstract

Le filtrage stochastique des systèmes max-plus linéaires incertains peut être réalisé grâce à l'algorithme proposé dans [21]. Dans la pratique, les systèmes max-plus linéaires incertains mettent en jeu des dépendances entre les états du système. Cette dépendance dégrade la qualité du filtrage proposé dans [21] à cause de la violation de l'hypothèse d'indépendance des variables aléatoires qui modélisent le système. L'objectif de ce travail est d'étendre cet algorithme de filtrage aux systèmes max-plus linéaires incertains dans lesquels les états peuvent générer des dépendances. Le résultat principal du papier s'appuie sur la propriété de vivacité des Graphes d'Événements Temporisés. Cette propriété de vivacité permet d'obtenir une représentation triangulaire par bloc pour la matrice d'état. À partir de cette nouvelle forme de la matrice d'état, nous proposons un nouvel algorithme de filtrage permettant de garder la propriété d'indépendance des variables stochastiques. L'efficacité de ce nouvel algorithme de filtrage stochastique est illustrée sur un exemple.

Contents

1	Introduction	1
2	Systèmes max-plus incertains	3
2.1	Algèbre max-plus et arithmétique par intervalles	3
2.2	Systèmes max-plus linéaires	3
2.3	Systèmes max-plus linéaires incertains	5
3	Filtrage stochastique des systèmes max-plus linéaires	6
3.1	Introduction	6
3.2	Filtrage bayésien	7
3.3	Le filtrage bayésien pour les systèmes max-plus	7
3.4	Calcul de l'espérance conditionnelle $\mathbb{E}[\mathbf{z} \mathbf{x}]$	7
3.5	Inversion de $\mathbb{E}[\mathbf{z} \mathbf{x}]$	8
4	Filtrage stochastique pour les systèmes max-plus sous forme implicite	10
4.1	Algorithme de filtrage pour les formes implicites	11
5	Exemple d'application	12
6	Conclusion	13

1 Introduction

Les systèmes max-plus linéaires. Les Systèmes dynamiques à Événements Discrets (SED) sont des systèmes dont la dynamique est pilotée par des événements, c'est-à-dire que l'évolution de l'état dépend entièrement de l'occurrence d'événements discrets asynchrones dans le temps. Les systèmes de fabrication, les réseaux de télécommunications, les réseaux de transport, sont des exemples de SED [1]. Pour décrire le comportement de ces systèmes, les équations

différentielles ordinaires ou partielles ne conviennent pas, c'est pour cette raison que de nouveaux formalismes ont été développés, parmi lesquels on peut citer, les chaînes de Markov, les réseaux de Petri, le lecteur est invité à consulter [4] pour une vue d'ensemble. Parmi les SED, une classe particulière impliquant des phénomènes de synchronisation et de retard a fait l'objet d'un développement algébrique dédié, généralement appelé algèbre max-plus linéaire. Cette classe de SED peut être représentée graphiquement par des Graphes d'Événements Temporisés (GET). Un GET est un réseau de Petri temporisé dont chaque place n'admet qu'une transition amont et qu'une transition aval.

La commande des systèmes max-plus linéaires. Le cadre de l'algèbre max-plus, un semi-anneau idempotent, est approprié pour décrire le comportement dynamique des GET, notamment grâce à des équations d'état linéaires (au sens de l'algèbre max-plus). Ces équations d'états sont analogues à celles que l'on trouve dans la théorie classique des systèmes linéaires. Cette représentation linéaire des GET permet de traiter des problèmes de contrôle. Parmi les problèmes résolus nous pouvons citer la commande optimale [6], [20], la commande prédictive [26], [23], la conception de contrôleurs robustes [17], [24]. Citons également, des problèmes d'estimation d'état [12], [19], [3], [9]. L'estimation d'état est un problème fondamental pour aborder des applications telles que la détection et le diagnostic de défauts [25] et la commande par retour d'état [11].

L'observation d'état des systèmes max-plus linéaires. L'approche observateur d'état proposée par [12] permet d'obtenir une estimation de l'état, par le dessous, aussi proche que possible du véritable état. Cette approche est efficace pour les systèmes déterministes. Elle permet de synthétiser un contrôleur basé sur la sortie d'un observateur [13]. Dans les systèmes dynamiques réels, les signaux des capteurs sont généralement corrompus par le bruit et les transitions des états du processus réel sont dans une certaine mesure perturbés par des événements exogènes. Si les propriétés stochastiques de ces sources de bruit sont disponibles, l'estimation de l'état peut être effectuée plus efficacement en tenant compte de ces connaissances. On peut alors développer une approche plus sophistiquée de l'estimation d'état inspirée du filtrage stochastique comme celle proposée dans [21].

Le filtrage de systèmes max-plus stochastiques. Les systèmes max-plus stochastiques sont des systèmes max-plus linéaires dans lesquels les entrées des matrices des équations d'état sont définies par leur densité de probabilité. Dans ce travail, nous allons nous focaliser sur les systèmes max-plus incertains, une sous-classe des systèmes max-plus stochastiques dans lesquels les entrées des matrices de la représentation d'état sont des variables aléatoires bornées dont le support est un intervalle réel. Les variables aléatoires sont également supposées statistiquement indépendantes les unes des autres. La méthode de filtrage proposée dans [21] pour ce type de systèmes peut se décomposer en deux étapes : une étape de prédiction basée sur le calcul de l'espérance conditionnelle $E[\text{observation}|\text{état}]$ et une étape, délicate dans le cas des systèmes max-plus, de mise à jour réalisée avec un algorithme de propagation de contraintes [15] qui permet d'inverser cette espérance conditionnelle pour obtenir une estimation de l'état.

Notre contribution. La méthode de filtrage proposée suppose que les entrées stochastiques des matrices système sont indépendantes. Malheureusement, dans la plupart des applications, les entrées des matrices du système correspondent à une somme de temps de traitement, dans laquelle divers temps de traitement apparaissent dans plusieurs entrées. Par conséquent, l'hypothèse que les entrées des matrices sont indépendantes n'est pas vérifiée.

Dans ce papier, nous proposons une extension de cette méthode de filtrage dans laquelle nous exploitons la propriété de vivacité d'un Graphe d'Événement Temporisé (GET). En effet, si le GET est vivant alors on peut exprimer sa dynamique sous forme implicite. Ainsi nous obtenons un nouvel algorithme de filtrage basé sur cette propriété de vivacité. Ce nouvel algorithme

permet d'améliorer les résultats proposés dans [21] notamment en assurant l'indépendance statistique des variables aléatoires lors des calculs.

Organisation. La section 2 contient les rappels sur les systèmes max-plus linéaires, l'analyse par intervalle et le filtrage stochastique des systèmes max-plus incertains. Dans la section 3 nous rappelons l'algorithme stochastique de filtrage présenté dans [21] pour les systèmes max-plus stochastiques. Nous présentons, notre contribution, l'algorithme de filtrage des systèmes max-plus stochastiques sous forme implicite dans la section 4. Dans la section 5, nous donnons un exemple numérique dans lequel nous comparons le filtre original [21] avec le nouvel algorithme que nous proposons. Enfin nous concluons et proposons des perspectives à notre travail dans la section 6.

2 Systèmes max-plus incertains

2.1 Algèbre max-plus et arithmétique par intervalles

Dans cette section, nous rappelons quelques éléments de base sur l'algèbre max-plus [1] et de l'arithmétique par intervalles.

Définition 1. Le symbole $\overline{\mathbb{R}}_{\max}$ désigne l'ensemble $\mathbb{R} \cup \{-\infty, +\infty\}$ muni des deux opérations binaires \max et $+$ (avec $\infty - \infty = -\infty$) notées \oplus et \otimes respectivement. On a $\varepsilon = -\infty$ et $e = 0$. On appelle cette structure l'algèbre max-plus.

Définition 2. Soit $A, B \in \overline{\mathbb{R}}_{\max}^{n \times p}$ et $C \in \overline{\mathbb{R}}_{\max}^{p \times q}$, on a $[A \oplus B]_{ij} = (a_{ij} \oplus b_{ij})$ et $[A \otimes C]_{ij} = (\bigoplus_{k=1}^p a_{ik} \otimes c_{kj})$.

Les premiers travaux sur l'analyse par intervalles dans l'algèbre max-plus sont issus de [18]. Depuis, de nombreux auteurs se sont intéressés à l'utilisation des intervalles dans l'algèbre max-plus [14], [5], [2].

Définition 3 (Intervalle max-plus). Un intervalle (fermé) $[x]$ dans l'algèbre max-plus est un sous-ensemble de $\overline{\mathbb{R}}_{\max}$ de la forme $[x] = [\underline{x}, \overline{x}] = \{x \in \overline{\mathbb{R}}_{\max} \mid \underline{x} \leq x \leq \overline{x}\}$. On notera $\mathbb{I}\overline{\mathbb{R}}_{\max}$ l'ensemble des intervalles de $\overline{\mathbb{R}}_{\max}$.

Un intervalle $[x] \subseteq [y]$ si et seulement si $\underline{y} \leq \underline{x} \leq \overline{x} \leq \overline{y}$. De même, $[x] = [y]$ si et seulement si $\underline{x} = \underline{y}$ et $\overline{x} = \overline{y}$. Un nombre $x \in \overline{\mathbb{R}}_{\max}$ peut être représenté par l'intervalle $[x, x]$.

Les opérations \oplus et \otimes seront étendues aux intervalles : $[\underline{x}, \overline{x}] \oplus [\underline{y}, \overline{y}] = [\underline{x} \oplus \underline{y}, \overline{x} \oplus \overline{y}]$ et $[\underline{x}, \overline{x}] \otimes [\underline{y}, \overline{y}] = [\underline{x} \otimes \underline{y}, \overline{x} \otimes \overline{y}]$.

Une matrice intervalle dans l'algèbre max-plus est une matrice dont les éléments sont des intervalles. On peut étendre les opérations \oplus et \otimes aux matrices intervalles. Soient les matrices intervalles $[A] = [\underline{A}, \overline{A}]$, $[B] = [\underline{B}, \overline{B}]$ et $[C] = [\underline{C}, \overline{C}]$ de dimensions $(n \times p)$, $(n \times p)$ et $(p \times q)$, alors $([A] \oplus [B])_{ij} = [a_{ij} \oplus b_{ij}]$ et $([A] \otimes [C])_{ij} = \bigoplus_{k=1}^p ([a_{ik}] \otimes [c_{kj}])$.

2.2 Systèmes max-plus linéaires

Les Graphes d'Événements Temporisés correspondent à la sous-classe des Réseaux de Petri temporisés dont chaque place a une seule transition amont et une seule transition aval. Le nombre de jetons dans une place s'interprète comme le nombre de ressources disponibles. Par exemple, le nombre de jetons dans la place p_2 (Fig. 1) correspond au nombre de pièces en attente d'être traitées par la machine M_1 , c'est-à-dire le nombre de "ressources" qui vont être consommées. Un jeton dans la place p_3 indique qu'une pièce est en train d'être traitée par la

machine. De plus, à chaque place peut-être associée une temporisation permettant de modéliser le temps de séjour que doit passer un jeton avant de pouvoir participer au franchissement de la transition aval. Il existe un autre modèle de GET où on associe une durée à chaque franchissement de transitions. Il a été montré qu'il est toujours possible de se ramener au cas où seules les places sont temporisées [22].

Dans cette section, nous allons rappeler la modélisation d'un GET dans l'algèbre max-plus. Pour ce faire, considérons l'exemple ci-dessous :

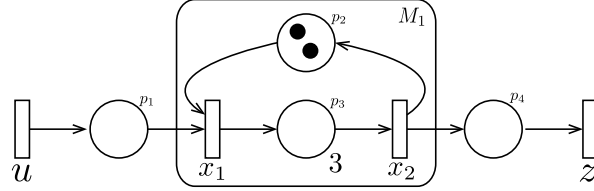


Figure 1: Modélisation d'un système de production composé d'une machine

Dans la suite, nous supposons que seules les places sont temporisées. Tout d'abord nous associons à chaque transition une fonction *dateur* $x_i(k) : \mathbb{N} \rightarrow \mathbb{R}$, où $x_i(k)$ désigne la date à laquelle se produit la $k^{\text{ième}}$ activation de la transition i . Ensuite, on considère que le tir des transitions se fait à vitesse maximale (au plus tôt). On obtient alors les équations suivantes :

$$\begin{cases} x_1(k) &= \max(u(k), x_2(k-2)) \\ x_2(k) &= x_1(k) + 3 \\ x_3(k) &= x_2(k) \end{cases}$$

En introduisant une variable $x_3(k) = x_2(k-1)$, nous obtenons une forme étendue du système précédent :

$$\begin{cases} x_1(k) &= \max(u(k), x_3(k-1)) \\ x_2(k) &= x_1(k) + 3 \\ x_3(k) &= x_2(k-1) \\ z(k) &= x_2(k) \end{cases}$$

Ces équations peuvent s'écrire sous forme linéaire au sens de l'algèbre max-plus :

$$x_1(k) = u(k) \oplus x_3(k-1) \quad (1a)$$

$$x_2(k) = 3 \otimes x_1(k) \quad (1b)$$

$$x_3(k) = x_2(k-1) \quad (1c)$$

$$z(k) = x_2(k) \quad (1d)$$

Et plus généralement, on obtient la représentation d'état qui suit :

$$\mathbf{x}(k) = A_0 \mathbf{x}(k) \oplus A_1 \mathbf{x}(k-1) \oplus B_0 \mathbf{u}(k), \quad (2a)$$

$$\mathbf{z}(k) = C \mathbf{x}(k) \quad (2b)$$

Les vecteurs \mathbf{x} , \mathbf{u} et \mathbf{z} ont des dimensions égales aux nombres de transitions : internes, d'entrée (commande) et de sortie (mesure).

Dans un GET, le nombre de jetons dans un circuit est invariant au cours de son évolution, quelle que soit la séquence des transitions franchies. Un GET est vivant si et seulement si le marquage de tout circuit est strictement positif. En effet, si le GET a un circuit sans jetons, il résulte que les transitions de ce circuit ne pourront jamais être tirées, il est alors non vivant. La matrice A_0 correspond aux places entre deux transitions internes nommées x_i , et dont les places ne portent pas de jetons dans le marquage initial.

Par ailleurs, on sait que la solution de l'équation $\mathbf{x} = A\mathbf{x} \oplus \mathbf{b}$ est donnée par $\mathbf{x} = A^*\mathbf{b}$ ¹. Finalement, on obtient la forme générale suivante :

$$\begin{cases} \mathbf{x}(k) &= A\mathbf{x}(k-1) \oplus B\mathbf{u}(k), \\ \mathbf{z}(k) &= C\mathbf{x}(k), \end{cases} \quad (3)$$

avec $A = A_0^*A_1$ et $B = A_0^*B_0$. Pour le GET cela signifie que :

1. toute place située entre deux transitions internes doit contenir exactement un jeton ;
2. toutes les places situées entre une transition d'entrée et une transition interne doivent être sans jeton ;
3. toutes les places situées entre une transition interne et une transition de sortie doivent être sans jeton.

Par des manipulations combinatoires tout à fait classiques (en augmentant l'état), on peut toujours faire en sorte qu'un GET vivant vérifie les conditions énoncées précédemment.

2.3 Systèmes max-plus linéaires incertains

Les GET permettent, par exemple, de modéliser des systèmes de production, de transports, etc. Dans ce cadre, les éléments des matrices de la représentation d'état (3) peuvent par exemple représenter des temps d'usinage et/ou de convoyage. Afin de prendre en compte les incertitudes dans la modélisation, nous allons considérer que ces durées sont aléatoires, uniformément distribuées et qu'elles évoluent sur un support borné. Par conséquent, nous avons considéré la représentation d'état qui suit [8][7] :

$$\mathbf{x}(k) = A_0(k)\mathbf{x}(k) \oplus A_1(k)\mathbf{x}(k-1) \oplus B_0(k)\mathbf{u}(k) \quad (4a)$$

$$\mathbf{z}(k) = C(k)\mathbf{x}(k), \quad (4b)$$

avec $A_0(k) \in [\underline{A}_0, \overline{A}_0]$, $A_1(k) \in [\underline{A}_1, \overline{A}_1]$, $B_0(k) \in [\underline{B}_0, \overline{B}_0]$ et $C(k) \in [\underline{C}, \overline{C}]$. La matrice \underline{A}_0 (respectivement \overline{A}_0) est la borne inférieure (respectivement supérieure) de $[A_0]$ avec $\underline{a}_0^{ij} \in \underline{A}_0$ (resp. $\overline{a}_0^{ij} \in \overline{A}_0$). La matrice $A_0(k)$ est telle que $a_0^{ij}(k) \in [\underline{a}_0^{ij}, \overline{a}_0^{ij}]$.

Remarque 1. *Précédemment, nous avons montré qu'il était possible de passer de la représentation (2) à la représentation d'état (3) en calculant la solution de l'équation $\mathbf{x} = A\mathbf{x} \oplus \mathbf{b}$. Dans le cas du système (4) nous ne pouvons pas appliquer la même transformation. En effet, même si les entrées des matrices A_0 , A_1 et B_0 sont supposées indépendantes, les produits $A_0^*A_1$ et $A_0^*B_0$ vont générer des dépendances entre les variables aléatoires des matrices résultantes [27][1].*

¹L'opérateur $(\bullet)^*$ est appelé l'étoile de Kleene et il est défini par $A^* = E \oplus A \oplus A^{\otimes 2} \oplus \dots$ avec E la matrice identité, i.e. une matrice carrée de même dimension que A avec la valeur e sur la diagonale principale et ε ailleurs.

Remarque 2. Si le GET décrit par (4) est vivant² alors il existe une matrice de permutation P telle que

$$P' A_0(k) P = \mathcal{A}_0(k) = \begin{pmatrix} B & \varepsilon \\ C & D \end{pmatrix} \quad (5)$$

où B et D sont des matrices carrées. Autrement dit, la matrice résultante $\mathcal{A}(k) = P' A_0(k) P$ est bloc-triangulaire inférieure. Ce résultat implique que si P existe alors la matrice $A_0(k)$ est non irréductible. Pour un GET cela signifie qu'il est possible de renuméroter les nœuds du graphe afin de mettre la matrice $A_0(k)$ sous forme bloc-triangulaire inférieure.

Comme nous l'avons précisé dans la remarque 1, nous ne pouvons pas exprimer le système (4) sous forme explicite. En revanche, en s'appuyant sur la remarque 2, si le GET est vivant il est possible d'exprimer la dynamique par

$$\mathbf{x}(k) = \mathcal{A}_0(k) \mathbf{x}(k) \oplus \mathbf{y}(k), \quad (6)$$

où $\mathcal{A}_0(k) \mathbf{x}(k)$ représente la partie implicite et $\mathbf{y} = A_1(k) \mathbf{x}(k-1) \oplus B_0(k) \mathbf{u}(k)$ correspond à la partie explicite. Par conséquent (4) s'écrit, composante par composante, comme le système d'équations :

$$\begin{aligned} x_1(k) &= y_1(k), \\ x_2(k) &= a_0^{21}(k) x_1(k) \oplus y_2(k), \\ x_3(k) &= a_0^{31}(k) x_1(k) \oplus a_0^{32}(k) x_2(k) \oplus y_3(k), \\ &\vdots \\ x_n(k) &= \left(\bigoplus_{l=1}^{n-1} a_0^{nl}(k) x_l(k) \right) \oplus y_n(k). \end{aligned} \quad (7)$$

où chaque y_i est soit $\max_{j=1}^n \{a_1^{ij}(k) + x_j(k-1)\}$ ou $\max_{j=1}^p \{b_0^{ij}(k) + u_j(k)\}$, avec $a_1^{ij}(k)$ et $b_0^{ij}(k)$ des variables indépendantes, $x_j(k-1)$ et $u_j(k)$ sont fixées. Ainsi les entrées du vecteur $\mathbf{y}(k)$ sont supposées indépendantes. De plus, la partie droite de (7) est $\max_{l=1}^{i-1} \{a_0^{il}(k) + x_l(k)\}$ pour tout $i \in [1, n]$ et dépend de $x_l(k)$ pour $l = 1$ à $i-1$. En supposant que le calcul de (7) se fait de $i = 1$ jusqu'à n et que les termes $x_l(k)$ sont fixés et connus, nous assurons l'indépendance des composantes du vecteur $\mathbf{x}(k)$. C'est une des contributions de cette nouvelle approche que nous proposons dans ce papier.

3 Filtrage stochastique des systèmes max-plus linéaires

3.1 Introduction

Les méthodes bayésiennes fournissent un cadre général et rigoureux pour les problèmes d'estimation d'état dynamique ([10]). Avant de détailler la contribution principale du papier, nous allons présenter les notions de base du filtrage stochastique dans le cadre des systèmes autonomes³ dont la dynamique est donnée par :

$$\begin{cases} \mathbf{x}(k) = f_k(\mathbf{x}(k-1)), \\ \mathbf{z}(k) = g_k(\mathbf{x}(k)), \end{cases} \quad (8)$$

²Nous dirons qu'un GET est vivant si, en présence d'entrées infinies (i.e. $u = \varepsilon$ non contraignante), chaque dateur au plus tôt $x(k)$ vérifie $x(k) < +\infty$ (tout événement arrive en un temps fini, ou encore, chaque transition est tirée un nombre infini de fois.)

³Tout système décrit par le système d'équations (3) peut être transformé en un système autonome $\mathbf{x}(k) = \mathcal{M} \mathbf{x}'(k-1)$ avec $\mathcal{M} = (AB)$ et l'état étendu $\mathbf{x}'(k-1) = (\mathbf{x}^T(k-1) \mathbf{u}^T(k))^T$

où $\mathbf{x} \in \mathbb{R}^n$ est le vecteur d'état, $\mathbf{z} \in \mathbb{R}^q$ l'observation/mesure à l'instant k , $f_k : \mathbb{R}^n \rightarrow \mathbb{R}^n$ une fonction non linéaire de transition de l'état $k - 1$ à l'état k et $g_k : \mathbb{R}^n \rightarrow \mathbb{R}^q$ la fonction d'observation.

3.2 Filtrage bayésien

Le schéma de filtrage stochastique pour les systèmes max-plus que nous développons dans ce papier est inspiré de travaux sur les systèmes continus classiques [16]. Ce problème de filtrage peut-être formulé comme suit. Tout d'abord, on suppose donnés les paramètres du modèle (A, B, C) ainsi que les caractéristiques statistiques de l'état initial $\mathbf{x}(0)$, c'est-à-dire son espérance et sa covariance. À l'instant k on dispose d'un ensemble de mesure/observation : $\mathcal{Z} = \{\mathbf{z}(1), \dots, \mathbf{z}(k)\}$ et le problème de filtrage consiste alors à estimer le vecteur aléatoire $\mathbf{x}(k)$ à partir de ces observations/mesures. Selon le critère de minimum de variance, l'estimateur optimal pour ce problème est l'espérance conditionnelle $\mathbb{E}[\mathbf{x}(k) | \mathbf{z}(1), \dots, \mathbf{z}(k)]$. Il est possible de calculer cette moyenne conditionnelle, en utilisant le filtre bayésien optimal [16]. Son équation est facile à écrire, mais il est en général impossible de la résoudre de façon analytique, sauf dans le cas particulier des systèmes linéaires gaussiens, où elle se ramène aux équations connues du filtre de Kalman-Bucy. Les équations considérées dans le cas des systèmes max-plus rendent les calculs nécessaires à l'obtention du filtre bayésien impossible à réaliser.

3.3 Le filtrage bayésien pour les systèmes max-plus

Dans cette section, nous présentons une méthode de filtrage bayésien alternative adaptée aux systèmes max-plus linéaire [21]. Afin de calculer l'estimée $\hat{\mathbf{x}}(k|k)$ de l'état $\mathbf{x}(k)$ on procède en deux étapes :

- Dans l'étape de *prédiction* on déduit une estimation, notée $\hat{\mathbf{x}}(k | k - 1)$, de l'état à l'instant k de la prévision à l'état $k - 1$:

$$\hat{\mathbf{x}}(k | k - 1) = \mathbb{E}[\mathbf{x}(k) | \hat{\mathbf{x}}(k - 1 | k - 1)]. \quad (9)$$

Le calcul de $\hat{\mathbf{x}}(k | k - 1)$ peut-être obtenu directement à partir de (4a).

- Dans l'étape de correction, on utilise la nouvelle observation $\mathbf{z}(k)$ afin de corriger la prédiction :

$$\hat{\mathbf{x}}(k | k) = \arg \min_{\mathbf{x}} \|\mathbf{x} - \hat{\mathbf{x}}(k | k - 1)\|_{\infty}, \quad (10a)$$

$$\text{s.t. } \mathbf{z}^{\#}(k) = \mathbb{E}[\mathbf{z}(k) | \mathbf{x}]. \quad (10b)$$

Dans la section suivante, nous détaillons les outils nécessaires au calcul des équations (9) et (10).

3.4 Calcul de l'espérance conditionnelle $\mathbb{E}[\mathbf{z} | \mathbf{x}]$

L'obtention des équations (9) et (10b) nécessite le calcul de l'espérance conditionnelle d'expression max-plus, c'est-à-dire d'expression du type max-affine

$$z_i = \bigoplus_{j=1}^n \{m_{ij}(k) \otimes x_j\} = \max_{j=1}^n \{m_{ij}(k) + x_j\} \quad (11)$$

où $m_{ij}(k) \in [\underline{m}_{ij}, \overline{m}_{ij}]$ avec $i \in [1, q]$, $j \in [1, n]$ et $\mathbf{x} \in \overline{\mathbb{R}}_{\max}$. Dans la suite, $m_{ij}(k)$ sont considérées comme des variables aléatoires indépendantes et uniformément distribuées. Par indépendance, nous entendons :

$$P\{m_{i1}(k) \in [m_{i1}], \dots, m_{in}(k) \in [m_{in}]\} = P\{m_{i1}(k) \in [m_{i1}]\} \times \dots \times P\{m_{in}(k) \in [m_{in}]\}.$$

Rappelons que si X est une variable, on notera F_X sa fonction de répartition, où $F_X(x) \equiv 0$ pour tout $x \leq x_0$ et $\mathbb{E}[X] = x_0 + \int_{x_0}^{\infty} (1 - F_X(x))dx$. Ce résultat implique que pour tout x si $F_{X_1}(x) \leq F_{X_2}(x)$ alors $\mathbb{E}[X_1] \geq \mathbb{E}[X_2]$. De plus, si $Z = \max_{j=1}^n \{X_j\}$, avec X_j des variables aléatoires indépendantes, alors $F_Z(z) = P[Z \leq z] = P[X_1 \leq z, X_2 \leq z, \dots, X_n \leq z] = \prod_{j=1}^n P[X_j \leq z] = \prod_{j=1}^n F_{X_j}(z)$. À présent, nous allons détailler le calcul de (11). Dans la suite, on note $F_{m_{ij}}(t)$ la fonction de répartition de la variable aléatoire $m_{ij}(k) \in [\underline{m}_{ij}, \overline{m}_{ij}]$. Le décalage induit par la variable d'état $m_{ij}(k) + x_j$ se répercute sur la fonction de répartition : $F_{m_{ij}}(t - x_j)$. Dans notre cas où les variables sont uniformément distribuées, la fonction de répartition est donnée par

$$F_{m_{ij}}(t - x_j) = \begin{cases} 0 & \text{if } t \leq \underline{m}_{ij} + x_j \\ \frac{t - x_j - \underline{m}_{ij}}{\overline{m}_{ij} - \underline{m}_{ij}} & \text{if } \underline{m}_{ij} + x_j < t \leq \overline{m}_{ij} + x_j \\ 1 & \text{sinon,} \end{cases} \quad (12)$$

Le calcul de la fonction de répartition de chaque $z_i, \forall i \in [1, q]$, noté $F_{z_i}(t)$, est donné par

$$F_{z_i}(t) = \prod_{j=1}^n F_{m_{ij}}(t - x_j). \quad (13)$$

L'équation (13) est une fonction polynomiale par morceau d'ordre n avec pour support $[\underline{z}_i, \overline{z}_i] = [\max_{j=1}^n \{\underline{m}_{ij} + x_j\}, \max_{j=1}^n \{\overline{m}_{ij} + x_j\}]$. Finalement le calcul de $\mathbb{E}[z_i | \mathbf{x}], \forall i \in [1, q]$ de l'équation (10b) est obtenu par⁴

$$\mathbb{E}[z_i | \mathbf{x}] = \overline{z}_i - \int_{\underline{z}_i}^{\overline{z}_i} F_{z_i}(t) dt \quad (14)$$

3.5 Inversion de $\mathbb{E}[\mathbf{z} | \mathbf{x}]$

Dans cette section, on s'intéresse au problème qui suit : étant donné une observation (ou mesure) \mathbf{z}^\sharp , on cherche un vecteur \mathbf{x}^\sharp , tel que $\mathbf{z}^\sharp = \mathbb{E}[\mathbf{z} | \mathbf{x}^\sharp]$. Plus formellement, on cherche à caractériser l'ensemble $\chi = \{\mathbf{x}^\sharp \in \mathbb{R}^n \mid \mathbb{E}[\mathbf{z} | \mathbf{x}^\sharp]\}$. La propagation de contraintes [15] est un outil efficace pour caractériser l'ensemble $\chi \subset \mathbb{R}$. Tout d'abord, nous rappelons quelques résultats sur les contracteurs. L'opérateur \mathcal{C}_χ est un contracteur pour l'ensemble χ , si pour tout $[\mathbf{x}] \in \mathbb{R}^n$, il vérifie la propriété de "contractance" $\mathcal{C}_\chi([\mathbf{x}]) \subset [\mathbf{x}]$ et de "consistance" $\mathcal{C}_\chi([\mathbf{x}]) \cap \chi = [\mathbf{x}] \cap \chi$. De plus, un contracteur est dit minimal si $\mathbf{x} \cap \chi = \mathcal{C}_\chi([\mathbf{x}])$. Dans la suite, nous supposons que : H1) $\chi \neq \emptyset$, H2) $\chi \subset [\mathbf{x}]$, H3) $\mathbb{E}[\mathbf{z} | \mathbf{x}] \leq \mathbf{z}^\sharp$, H4) $\forall j \in [1, n], \mathbb{E}[\mathbf{z} | (\underline{x}_1, \underline{x}_2, \dots, \overline{x}_j, \dots, \underline{x}_n)^T] \geq \mathbf{z}^\sharp$. Le contracteur \mathcal{C}_χ est détaillé dans l'algorithme 1.

⁴Une procédure de calcul de l'espérance a été développé exclusivement pour la loi uniforme. Néanmoins, toutes les densités polynomiales peuvent bénéficier d'une procédure similaire de calcul de l'équation (13). En revanche, il faut remarquer que la complexité est associée au nombre de morceaux.

<p>Algorithme 1 : Contracteur Intervalle : $\mathcal{I} = \Omega_\chi([\mathbf{x}])$</p> <p>Données : $F(A(k) \in [A])$ (fonc. répartition de la mat. du syst.), $[\mathbf{x}], \mathbf{z}^\sharp$</p> <p>Résultat : $\mathcal{I} = \Omega_\chi([\mathbf{x}])$</p> <pre> 1 tant que $\underline{\mathbf{x}}' \neq \underline{\mathbf{x}}$ et $\bar{\mathbf{x}}' \neq \bar{\mathbf{x}}$ faire 2 $\underline{\mathbf{x}}' \leftarrow \underline{\mathbf{x}}; \bar{\mathbf{x}}' \leftarrow \bar{\mathbf{x}}$; 3 pour chaque $i \in [1, q]$ faire 4 pour chaque $j \in [1, n]$ faire 5 /* Dichotomie inférieure */ 6 $\mathbf{x} \leftarrow \bar{\mathbf{x}}; x_j \leftarrow \underline{x}_j;$ 7 si $\mathbb{E}[z_i \mathbf{x}] < z_i^\sharp$ alors 8 $\underline{\mathbf{x}} \leftarrow \Delta_{ij}^L([\underline{\mathbf{x}}, \bar{\mathbf{x}}]);$ 9 /* Dichotomie supérieure */ 10 $\mathbf{x} \leftarrow \underline{\mathbf{x}}; x_j \leftarrow \bar{x}_j;$ 11 si $\mathbb{E}[z_i \mathbf{x}] > z_i^\sharp$ alors 12 $\bar{\mathbf{x}} \leftarrow \Delta_{ij}^U([\underline{\mathbf{x}}, \bar{\mathbf{x}}]);$ 13 14 11 return $[\underline{\mathbf{x}}, \bar{\mathbf{x}}];$ </pre>	<p>Algorithme 2 : recherche unidimensionnelle : Δ_{ij}^L ou $U([\mathbf{x}])$</p> <p>Données : $F(M(k) \in [M])$ (fonc. répartition de la mat. du syst.), $[\mathbf{x}], (i, j), z_i^\sharp, \text{type L or U}$</p> <p>Résultat : $\mathbf{w} = \Delta_{ij}^L([\mathbf{x}])$ ou $\Delta_{ij}^U([\mathbf{x}])$</p> <pre> 1 $(\mathbf{y}, \mathbf{w}) \leftarrow \begin{cases} (\bar{\mathbf{x}}, \underline{\mathbf{x}}) & \text{si L} \\ (\underline{\mathbf{x}}, \bar{\mathbf{x}}) & \text{si U} \end{cases};$ 2 tant que $\underline{\mathbf{x}} \neq \bar{\mathbf{x}}$ faire 3 $y_j \leftarrow (\underline{x}_j + \bar{x}_j)/2;$ 4 si $\mathbb{E}[z_i \mathbf{y}] > z_i^\sharp$ alors 5 $\bar{x}_j \leftarrow y_j;$ 6 sinon 7 $\underline{x}_j \leftarrow y_j;$ 8 $w_j \leftarrow y_j;$ 9 return $\mathbf{w};$ </pre>
---	--

Le calcul de Δ_{ij}^L et Δ_{ij}^U (Algorithme 2) est une recherche unidimensionnelle réalisée à l'aide d'un algorithme basé sur une méthode de dichotomie. À chaque étape, l'intervalle de recherche est initialisé avec $[\underline{x}_j, \bar{x}_j]$ et il est ensuite divisé en deux parties égales. La partie contenant une solution est utilisée comme entrée de l'algorithme pour l'étape suivante. L'algorithme s'arrête quand l'intervalle solution est suffisamment petit. Finalement, le contracteur Ω_χ correspond à la composition de $2qn$ opérateurs $\Delta : \Omega_\chi([\mathbf{x}]) = (\Delta_{11}^L \circ \Delta_{11}^U \circ \dots \circ \Delta_{1n}^L \circ \Delta_{1n}^U \circ \dots \circ \Delta_{qn}^L \circ \Delta_{qn}^U)([\mathbf{x}])$. Les lignes 6 et 9 de l'algorithme 1 et la ligne 4 de l'algorithme 2 permettent de calculer l'équation (14). Dans le cas où les conditions ne sont pas satisfaites, les opérateurs correspondent à l'opérateur identité : $\Delta_{ij}^L([\mathbf{x}]) = \text{Id}^L([\mathbf{x}]) = \underline{\mathbf{x}}$ et $\Delta_{ij}^U([\mathbf{x}]) = \text{Id}^U([\mathbf{x}]) = \bar{\mathbf{x}}$.

Remarque 3. Le contracteur $\Omega_\chi(\bullet)$ est monotone et il vérifie les propriétés de contractance et de consistance (cf. [21, Lemme 2] pour la preuve).

Remarque 4. $\Omega_\chi(\mathcal{I}) = \mathcal{I}$ est un point fixe et c'est le plus petit intervalle tel que $\chi \subset \mathcal{I}$ (cf. [21, Lemme 3] pour la preuve).

À l'issue de la procédure de contraction, nous obtenons un intervalle $\mathcal{I} = [\underline{\mathbf{x}}^{opt}, \bar{\mathbf{x}}^{opt}]$ tel que pour une mesure \mathbf{z}^\sharp , $\exists \mathbf{x} \in [\underline{\mathbf{x}}^{opt}, \bar{\mathbf{x}}^{opt}], \mathbf{z}^\sharp = \mathbb{E}[\mathbf{z} | \mathbf{x}]$. Cet intervalle n'est généralement pas dégénéré (c.-à-d. $\underline{\mathbf{x}} \neq \bar{\mathbf{x}}$).

Comme évoqué précédemment, parmi toutes les solutions possibles $\mathbf{x} \in \chi$, nous cherchons la plus proche d'un point \mathbf{x}^0 donné, c'est-à-dire $\mathbf{x}^\sharp = \arg \min_{\mathbf{x} \in \chi} \|\mathbf{x} - \mathbf{x}^0\|_\infty$ (cf. Eq. (10b)). Tout d'abord, remarquons que nous obtenons une sur-approximation \mathcal{I} de l'ensemble χ (Rem. 4). Ensuite, il est évident qu'il peut exister plusieurs solutions \mathbf{x}^\sharp pour lesquelles l'équation (10b) atteint sa valeur minimale. Par conséquent, pour résoudre ce problème, nous modifions légèrement le problème d'optimisation par $\mathbf{x}' = \arg \min_{\mathbf{x} \in \mathcal{I}} \|\mathbf{x} - \mathbf{x}^0\|_\infty$. L'algorithme 3 (Ligne 7) permet de trouver une solution à ce problème. Finalement, il est aisé de montrer que la solution de $\min_{\mathbf{x} \in \mathcal{I}} \|\mathbf{x} - \mathbf{x}^0\|_\infty$ est unique. De même, on ne peut pas garantir que la solution trouvée appartienne à χ . Cette solution est sous-optimale et les bornes d'erreurs ne sont pas connues

étant donné que le critère repose uniquement sur une valeur de \mathbf{x} tel que $\mathbf{z}^\# = \mathbb{E}[\mathbf{z}|\mathbf{x}]$ avec la minimisation au sein d'un intervalle qui contient χ . Pour plus de détails sur le fonctionnement de cet algorithme, nous renvoyons le lecteur à [21].

Algorithme 3 : Sélection d'une solution dans \mathcal{I}

Données : $F(M(k) \in [M])$ (fonc. répartition de la matrice du système), $\mathbf{z}^\#, \mathbf{x}^0$
Résultat : $\mathbf{x}^\# = \text{Inv}(\mathbf{z}^\#, \mathbf{x}^0)$

- 1 générer $[\underline{\mathbf{x}}, \bar{\mathbf{x}}]$; $bool \leftarrow \text{vrai}$;
- 2 **tant que** $bool$ **faire**
- 3 $[\underline{\mathbf{x}}, \bar{\mathbf{x}}] \leftarrow \Omega_\chi([\underline{\mathbf{x}}, \bar{\mathbf{x}}])$;
- 4 $bool \leftarrow \underline{\mathbf{x}} \neq \bar{\mathbf{x}}$;
- 5 **si** $bool$ **alors**
- 6 **pour chaque** $j \in [1, q]$ **faire**
- 7 $x'_j \leftarrow \begin{cases} \underline{x}_j & \text{si } x_j^0 < \underline{x}_j \\ x_j^0 & \text{si } \underline{x}_j \leq x_j^0 \leq \bar{x}_j \\ \bar{x}_j & \text{sinon.} \end{cases}$ // $\mathbf{x}' = \arg \min_{\mathbf{x} \in [\underline{\mathbf{x}}, \bar{\mathbf{x}}]} \|\mathbf{x} - \mathbf{x}^0\|_\infty$
- 8 $j^\# \leftarrow \arg \max_{j \in [1, n]} |x'_j - x_j^0|$;
- 9 /* Dégénération de l'intervalle */
- 9 $\underline{x}_{j^\#} \leftarrow x'_{j^\#}$; $\bar{x}_{j^\#} \leftarrow x'_{j^\#}$
- 10 **return** $\underline{\mathbf{x}}$ ou $\bar{\mathbf{x}}$; // $\underline{\mathbf{x}} = \bar{\mathbf{x}}$

3.5.1 Algorithme de filtrage stochastique

À présent, nous avons toutes les briques de base pour réaliser le filtrage stochastique de systèmes max-plus linéaires. Soit le système max-plus linéaire incertain :

$$\begin{aligned} \mathbf{x}(k) &= A(k)\mathbf{x}(k-1), \\ \mathbf{z}(k) &= C(k)\mathbf{x}(k), \end{aligned}$$

avec $\mathbf{x} \in \overline{\mathbb{R}}_{\max}^n$, $\mathbf{z} \in \overline{\mathbb{R}}_{\max}^q$, $A(k) \in [A] \subset \overline{\mathbb{R}}_{\max}^{n \times n}$ et $C(k) \in [C] \subset \overline{\mathbb{R}}_{\max}^{q \times n}$. Les entrées des matrices $[A]$ et $[C]$ sont des variables aléatoires indépendantes uniformément distribuées. En résumé, la méthode de filtrage stochastique d'un système incertain max-plus linéaire est donnée dans l'algorithme 4.

Algorithme 4 : Algorithme de Filtrage pour les systèmes max-plus linéaires sous forme explicite

Données : $F(A)$, $F(C)$ et $\mathbf{z}^\#(k)$
Résultat : $\hat{\mathbf{x}}(k|k) = \text{Filtre}(\mathbf{z}^\#(k), \hat{\mathbf{x}}(k-1|k-1))$

- 1 $\hat{\mathbf{x}}(k|k-1) = \mathbb{E}[\mathbf{x}(k)|\hat{\mathbf{x}}(k-1|k-1)]$; // Eq. (14)
- 2 $\hat{\mathbf{x}}(k|k) = \text{Inv}(\mathbf{z}^\#(k), \hat{\mathbf{x}}(k|k-1))$; // Eq. (10), Algo. 3
- 3 **return** $\hat{\mathbf{x}}(k|k)$

4 Filtrage stochastique pour les systèmes max-plus sous forme implicite

Le système stochastique décrit par l'équation (7) est implicitement dépendant composante par composante. Par conséquent, pour calculer correctement l'espérance de la i -ème composante

de \mathbf{x} , il faut être capable de calculer la distribution conjointe d'une composante par rapport aux autres dont elle dépend. Ce calcul, compte tenu de la forte non-linéarité des systèmes considérés, est généralement impossible et donc le filtrage proposé dans la section précédente ne peut plus être utilisé directement. Dans la suite, nous proposons une modification du schéma de filtrage pour lever cette difficulté. Cette nouvelle méthode de filtrage repose sur la forme implicite des systèmes max-plus linéaires.

4.1 Algorithme de filtrage pour les formes implicites

Soit le système max-plus incertain :

$$\begin{cases} \mathbf{x}(k) &= A_0(k)\mathbf{x}(k) \oplus A_1(k)\mathbf{x}(k-1), \\ \mathbf{z}(k) &= C(k)\mathbf{x}(k), \end{cases} \quad (15)$$

où $\mathbf{x} \in \overline{\mathbb{R}}_{\max}^n$, $\mathbf{z} \in \overline{\mathbb{R}}_{\max}^q$, $A_0(k) \in [A_0] \subset \overline{\mathbb{R}}_{\max}^{n \times n}$, $A_1(k) \in [A_1] \subset \overline{\mathbb{R}}_{\max}^{n \times n}$ et $C(k) \in [C] \subset \overline{\mathbb{R}}_{\max}^{q \times n}$. Les entrées des matrices $[A_0]$, $[A_1]$ et $[C]$ sont des variables aléatoires indépendantes uniformément distribuées.

D'après la remarque 2, on sait que la matrice A_0 peut s'écrire sous forme triangulaire :

$$x_i(k) = \left(\bigoplus_{j=1}^{i-1} a_0^{ij}(k)x_j(k) \right) \oplus y_i(\mathbf{x}(k-1)), \quad (16)$$

avec $y_i(\mathbf{x}(k-1))$ la i -ème entrée de $\mathbf{y}(\mathbf{x}(k-1)) = A_1(k)\mathbf{x}(k-1)$ (on suppose que la composante de $\mathbf{x}(k-1)$ est obtenue à partir de l'itération précédente).

En considérant l'équation (14), la prédiction pour tout $i \in [1, n]$ est donnée par

$$\hat{x}_i(k|k-1) = \mathbb{E} \left[\left(\bigoplus_{j=1}^{i-1} a_0^{ij}(k)\hat{x}_j(k|k) \right) \oplus y_i(\mathbf{x}(k-1)) \right]. \quad (17)$$

Afin de calculer $\hat{x}(k|k)$ il faut appeler n -fois la procédure **Inv** (Algo 3). Clairement, il ne s'agit pas d'un schéma de filtrage classique en deux étapes. En effet, nous mettons à jour séparément la prédiction pour chaque composante i du vecteur d'état. De plus, pour traiter les formes implicites, la fonction $f_k(\mathbf{x}(k), \mathbf{x}(k-1))$ est remplacée par la fonction $f_k(\mathbf{x}(k-1))$. En résumé, l'algorithme de filtrage est donné par l'algorithme 5.

Algorithme 5 : Algorithme de filtrage des formes implicites

Données : $F(A_0)$, $F(A_1)$, $F(C)$ et $\mathbf{z}^\#(k)$

Résultat : $\hat{\mathbf{x}}(k|k) = \mathbf{Filtre}(\mathbf{z}^\#(k), \hat{\mathbf{x}}(k-1|k-1))$

- 1 $\mathbf{y} = A_1(k)\hat{\mathbf{x}}(k-1|k-1)$;
 - 2 $\hat{\mathbf{x}}(k|k-1) \leftarrow (\varepsilon, \dots, \varepsilon)^T$;
 - 3 **pour chaque** $i \in [1, n]$ **faire**
 - 4 $\hat{x}_i(k|k-1) \leftarrow \mathbb{E} \left[\left(\bigoplus_{j=1}^{i-1} a_0^{ij}(k)\hat{x}_j(k|k) \right) \oplus y_i \right]$; // Eq. (17)
 - 5 $\hat{\mathbf{x}}(k|k) \leftarrow \mathbf{Inv}(\mathbf{z}^\#(k), \hat{\mathbf{x}}(k|k-1))$; // Eq. (10)
 - 6 **return** $\hat{\mathbf{x}}(k|k)$;
-

5 Exemple d'application

Dans cette section, nous allons présenter un exemple d'application de l'algorithme de filtrage des formes implicites. Tout d'abord, considérons le GET de la Figure 2. Ce GET représente un système de production composé de deux machines.

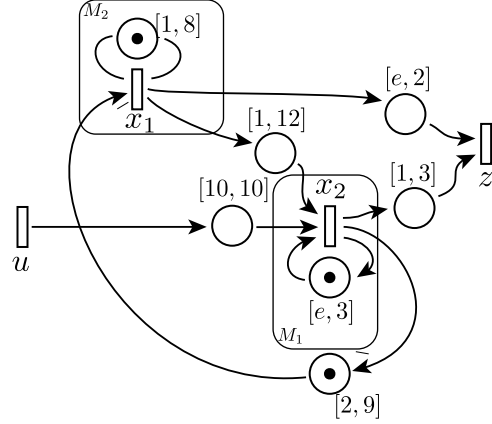


Figure 2: TEG de l'Exemple.

Le comportement dynamique de ce GET peut être décrit (Section 2.2) par la représentation d'état qui suit :

$$\mathbf{x}(k) = A_0(k)\mathbf{x}(k) \oplus A_1(k)\mathbf{x}(k-1) \oplus Bu(k), \quad (18)$$

avec $A_0(k) \in \begin{pmatrix} \varepsilon & \varepsilon \\ [1, 12] & \varepsilon \end{pmatrix}$, $A_1(k) \in \begin{pmatrix} [1, 8] & [2, 9] \\ \varepsilon & [e, 3] \end{pmatrix}$ et $B = \begin{pmatrix} \varepsilon \\ 10 \end{pmatrix}$. La commande du système $u(k) \in [2, 4]$ où l'intervalle $[2, 4]$ représente le support d'une distribution uniforme.

L'équation d'observation de ce système est donnée ci-dessous :

$$\mathbf{z}(k) = C(k)\mathbf{x}(k), \quad (19)$$

avec $C(k) \in ([e, 2] \ [1, 3])$. On suppose que l'état initial du système est connu et donné par $\mathbf{x}(0) = (e \ e)^T$. On suppose également que $\hat{\mathbf{x}}(k-1|k-1) \equiv \mathbf{x}(k-1) = \mathbf{x}(0)$.

Nous allons comparer deux approches de filtrage. La première, notée F_1 correspond à la nouvelle approche que nous proposons dans ce papier. La seconde approche est celle proposée dans le papier [21] et notamment dans l'annexe où une extension des résultats est proposée pour les systèmes périodiques. Ce filtre sera noté F_2 dans la suite. Résumons :

- *Filtre* F_1 utilise la méthode de filtrage proposée par l'algorithme 5. Afin de prendre en compte l'action de l'entrée $u(k)$, la ligne 1 est remplacée par $\mathbf{y} = A_1(k)\hat{\mathbf{x}}(k-1|k-1) \oplus Bu(k)$;
- *Filtre* F_2 utilise l'algorithme 4 en remplaçant la ligne 1 par :

$$\begin{aligned} \hat{x}_i(k|k-1) &= \mathbb{E}[x_i(k) | (\hat{\mathbf{x}}(k-1|k-1)^T \ u(k))^T], \\ &= \mathbb{E} \left[\left(\bigoplus_{j=1}^{i-1} a_0^{ij}(k) \hat{x}_j(k|k-1) \right) \oplus \left(\bigoplus_{j=1}^n a_1^{ij}(k) \hat{x}_j(k-1|k-1) \right) \oplus b^i u(k) \right], \end{aligned}$$

pour tout $i \in \{1, \dots, n\}$. Cette méthode correspond à la stratégie donnée dans l'annexe de [21] pour les systèmes à comportement périodique.

Le tableau 1 montre les résultats obtenus pour les simulations avec 4000 occurrences de tirs. Chaque valeur dans le tableau correspond à l'erreur quadratique moyenne (root-mean-square-error) $\text{RMSE}(x_i(k), \hat{x}_i(k|k))$ ⁵ en fonction du filtre considéré.

État i	Filtre F_1	Filtre F_2
1	2.0205	2.2338
2	1.1530	1.1530

Table 1: Tableau des résultats de l'Exemple.

L'analyse du tableau 1 montre une faible différence sur la qualité de l'estimation de l'état x_2 entre les deux filtres. Les valeurs de l'erreur sont proches. En revanche, on s'aperçoit que pour l'état x_1 la valeur de l'erreur est plus faible avec le nouvel algorithme de filtrage F_1 que nous proposons dans ce papier. On voit que dans le cas où la prédiction est très bruitée (paire $(A_0(k), A(k))$), l'algorithme proposé permet d'améliorer de façon significative l'estimation.

Interprétation des résultats de simulation :

- En termes de complexité (temps d'exécution) le Filtre F_2 est beaucoup plus rapide que F_1 étant donné qu'un seul appel de `Inv` est nécessaire ;
- Le calcul du filtre F_2 repose sur une hypothèse d'indépendance des $\hat{x}_j(k|k-1)$. Cette hypothèse n'est plus vérifiée lorsque des entrées exogènes agissent sur le système.

6 Conclusion

Dans ce papier, nous avons développé une extension de l'algorithme de filtrage stochastique proposé dans [21]. Ce nouvel algorithme exploite la forme implicite des systèmes max-plus linéaires. La représentation d'état de ces systèmes peut se mettre sous forme bloc-triangulaire, ce qui permet de conserver les résultats sur l'indépendance des variables aléatoires du système. Deux challenges intéressants sont en cours d'exploration. Tout d'abord, l'introduction d'un critère permettant d'évaluer l'erreur d'estimation afin de réaliser un compromis entre celle-ci et la valeur de la prédiction. Ensuite, il serait intéressant de propager l'estimation obtenue avec l'algorithme présenté dans ce papier aux états sans jetons.

References

- [1] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat. *Synchronization and Linearity : An Algebra for Discrete Event Systems*. Wiley and Sons, 1992.
- [2] Thomas Brunsch, Laurent Hardouin, Carlos Andrey Maia, and Joerg Raisch. Duality and interval analysis over idempotent semirings. *Linear Algebra and its Applications*, 437:2436–2454, 2012.

⁵ $\text{RMSE}(\mathbf{a}, \mathbf{b}) = \sqrt{\frac{1}{N} \sum_{i=1}^N (\mathbf{a}_i - \mathbf{b}_i)^2}$.

- [3] R. M. Ferreira Cândido, R. Santos-Mendes, L. Hardouin, and C.A. Maia. Particle filter for max-plus systems. *European Control Conference, ECC 2013*, 2013.
- [4] C. G. Cassandras and S. Laforune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.
- [5] Katarína Cechlárová and R.A. Cuninghame-Green. Interval systems of max-separable linear equations. *Linear Algebra and its Applications*, 340(1):215–224, 2002.
- [6] Guy Cohen, Stéphane Gaubert, and Jean-Pierre Quadrat. Max-plus algebra and system theory: Where we are and where to go now. *Annual Reviews in Control*, 23:207 – 219, 1999.
- [7] Renato Markele Ferreira Cândido, Laurent Hardouin, Mehdi Lhommeau, and Rafael Santos Mendes. An algorithm to compute the inverse image of a point with respect to a nondeterministic max-plus linear system. *IEEE Transactions on Automatic Control*, 66(4):1618–1629, 2021.
- [8] Renato Markele Ferreira Cândido, Laurent Hardouin, Mehdi Lhommeau, and Rafael Santos Mendes. Conditional reachability of uncertain max plus linear systems. *Automatica*, 94:426–435, 2018.
- [9] S.S. Farahani, T. van den Boom, and B. De Schutter. On optimization of stochastic max-min-plus-scaling systems – An approximation approach. *Automatica*, 83:20–27, September 2017.
- [10] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEEE Proceedings F, Radar and Signal Processing*, 140(2):107–113, 1993.
- [11] L. Hardouin, B. Cottenceau, Y. Shang, and J. Raisch. *Control and State Estimation for Max-Plus Linear Systems*. Now Foundations and Trends, 2018.
- [12] L. Hardouin, C. A. Maia, B. Cottenceau, and M. Lhommeau. Observer design for (max,+) linear systems. *IEEE Trans. on Automatic Control*, 55 - 2:538 – 543, 2010.
- [13] L. Hardouin, Y. Shang, C. A. Maia, and B. Cottenceau. Observer-based controllers for max-plus linear systems. *IEEE Transactions on Automatic Control*, 62(5):2153–2165, 2017.
- [14] Laurent Hardouin, Bertrand Cottenceau, Mehdi Lhommeau, and Euriell Le Corrnc. Interval systems over idempotent semiring. *Linear Algebra and its Applications*, 431(5):855–862, 2009.
- [15] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer-Verlag, London, 2001.
- [16] Andrew H. Jazwinski. *Stochastic processes and filtering theory*. Number 64 in Mathematics in science and engineering. Acad. Press, New York, NY [u.a.], 1970.
- [17] M. Lhommeau, L. Hardouin, J.-L. Ferrier, and I. Ouerghi. Interval analysis in dioid: Application to robust open-loop control for timed event graphs. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pages 7744–7749, 2005.
- [18] Grigori L. Litvinov and Andrei N. Sobolevskii. Idempotent interval analysis and optimization problems. *Reliable Computing*, 7(5):353–377, 2001.
- [19] M. D. Loreto, S. Gaubert, R. D. Katz, and J. Loiseau. Duality between invariant spaces for max-plus linear discrete event systems. *SIAM J. on Control and Optimaztion*, 2010.
- [20] C. A. Maia, L. Hardouin, R. Santos Mendes, and B. Cottenceau. Optimal Closed-Loop Control for Timed Event Graphs in Dioid. *IEEE-TAC*, 48(12):2284–2287, 2003.
- [21] Rafael Santos Mendes, Laurent Hardouin, and Mehdi Lhommeau. Stochastic filtering of max-plus linear systems with bounded disturbances. *IEEE Transactions on Automatic Control*, 64(9):3706–3715, 2019.
- [22] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.
- [23] I. Necoara, B. De Schutter, T.J.J. van den Boom, and H. Hellendoorn. Stable model predictive control for constrained max-plus-linear systems. *Discrete Event Dynamic Systems: Theory and Applications*, 17(3):329–354, sep 2007.
- [24] I. Necoara, B. De Schutter, T.J.J. van den Boom, and H. Hellendoorn. Robust control of con-

- strained max-plus-linear systems. *Int. J. of Robust and Nonlinear Control*, 19(2):218–242, January 2009.
- [25] Claire Paya, Euriell Le Corrond, Yannick Pencolé, and Philippe Vialletelle. Observer-based detection of time shift failures in $(\max,+)$ -linear systems. In *The 31st International Workshop on Principles of Diagnosis (DX-2020)*, Nashville, United States, September 2020.
- [26] T.J.J. van den Boom, J. Xu, and B. De Schutter. Corrections to “Model predictive control for stochastic max-plus linear systems with chance constraints”, [IEEE Trans. on Aut. Control, 64(1): 337–342, 2019]. *IEEE Transactions on Automatic Control*, 65(2):905–906, February 2020.
- [27] J. Xu, T. van den Boom, and B. De Schutter. Model predictive control for stochastic max-plus linear systems with chance constraints. *IEEE Transactions on Automatic Control*, 64(1):337–342, 2019.