



HAL
open science

Contribution à la commande prédictive des systèmes dynamiques modélisés par réseaux de Petri

Marwa Taleb

► **To cite this version:**

Marwa Taleb. Contribution à la commande prédictive des systèmes dynamiques modélisés par réseaux de Petri. Automatique. Université du Havre, 2016. Français. NNT : 2016LEHA0026 . tel-01668510

HAL Id: tel-01668510

<https://theses.hal.science/tel-01668510>

Submitted on 20 Dec 2017

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.



Normandie Université

THESE

Pour obtenir le diplôme de doctorat

Spécialité : Génie informatique, automatique et traitement du signal

Préparée au sein du Laboratoire GREAH, Université du Havre

Contribution à la commande prédictive des systèmes dynamiques modélisés par réseaux de Petri

Présentée et soutenue par
Marwa TALEB

Thèse soutenue publiquement le 23 Novembre 2016
devant le jury composé de

Mme. Audine SUBIAS	Maître de conférences (HDR), INSA Toulouse	Rapporteur
M. Noureddine ZEROUNI	Professeur des universités, ENSMM Besançon	Rapporteur
M. Francesco BASILE	Professeur des universités, Université de Salerne	Examineur
M. Eric SANLAVILLE	Professeur des universités, Université du Havre	Examineur
M. Edouard LECLERCQ	Maître de conférences, Université du Havre	Encadrant
M. Dimitri LEFEBVRE	Professeur des universités, Université du Havre	Directeur de thèse

Thèse dirigée par M. Dimitri LEFEBVRE et M. Edouard LECLERCQ, Laboratoire GREAH



DÉDICACE

À mes parents...

REMERCIEMENTS

Les travaux de recherche présentés dans ce manuscrit de thèse ont été effectués au sein du Groupe de Recherche en Électrotechnique et automatique du Havre, GREAH de l'Université du Havre, France, sous le financement du Ministère de l'Enseignement Supérieur et de la Recherche (MESR). Avant d'entamer la présentation de ces travaux, je tiens à adresser, de manière générale, mes sincères remerciements à toutes les personnes qui ont participé, de près ou de loin, à l'accomplissement et à la réussite de ce travail de thèse.

First of all, I would like to thank Mr. Francesco BASILE, Professor at the University of Salerno, for having honored me to take part in the evaluation of this work. Mes vifs remerciements s'adressent aussi à Mme. Audine SUBIAS, maître de conférences (HDR) à l'INSA de Toulouse, et à M. Noureddine ZERHOUNI, Professeur à ENSMM Besançon, d'avoir eux aussi accepté d'être rapporteurs de ce travail. Je remercie également Monsieur Eric SANLAVILLE, Professeur à l'université du Havre, pour l'intérêt qu'il a bien voulu accorder à ce travail en acceptant de l'examiner.

Je voudrais exprimer toute ma gratitude et ma profonde reconnaissance à mon Directeur de thèse M. Dimitri LEFEBVRE, professeur à l'Université du Havre, et à mon encadrant M. Edouard LECLERCQ, maître de conférences à l'université du Havre, pour m'avoir confié ce travail, pour la qualité de leurs encadrements, pour leur exceptionnelle disponibilité, leur rigueur, la richesse de leurs remarques et conseils toujours constructifs, leur infinie patience ainsi que leurs encouragements. De plus, leur esprit visionnaire et leurs expériences m'ont été fort utiles pour développer mon goût de recherche.

Toute ma reconnaissance va également à tous les membres du département Génie Productique et Mécanique, GMP de l'IUT du Havre avec qui ce fût et restera un plaisir de travailler.

Je n'oublierai évidemment pas toute l'équipe administrative et technique du GREAH pour l'ambiance sympathique et cordiale qu'ils ont su faire régner au sein du laboratoire. Je tiens particulièrement à remercier tous les membres de l'équipe CSFS, spécialement Anas HIJAZI et Rabah AMMOUR, et mon collègue de bureau Abdou Rahmen ADEN DRIET.

Un grand Merci va également à tous mes ami(e)s, notamment Habiba BALI, Nesrine BAHRI, Sonia ENNASSIRI, Khadija CHRAITIA, Mouheb DHIFLI, Adel BELLARA, Mohamed AMAMI, Mohamed BEN HAMIDA, Hamza ENNASSIRI et Nabih DEBBAH non seulement pour leur gentillesse mais également pour les moments inoubliables que nous avons partagés ensemble.

Je tiens à témoigner ma profonde gratitude et mes très sincères remerciements à mes chers parents, mes deux chers frères, mes tantes ainsi que toute ma famille pour la confiance qu'ils m'ont accordée, pour leurs encouragements et leur soutien humain et moral durant toutes mes années d'étude.

Articles publiés dans des revues internationales et européennes avec comités de lecture

A1 - Marwa Taleb, Edouard Leclercq & Dimitri Lefebvre. "Model Predictive Control for Discrete and Continuous Timed Petri Nets". *International Journal of Automation and Computing (IJAC)*. Accepted (under press).

A2 - Marwa Taleb, Edouard Leclercq & Dimitri Lefebvre. "Commande prédictive des réseaux de Petri hybrides élémentaires". *Journal Européen des Systèmes Automatisés (JESA)*. Accepted (under press) .

Articles soumis dans des revues internationales avec comités de lecture

A3 - Marwa Taleb, Edouard Leclercq & Dimitri Lefebvre. "Control design of timed continuous Petri nets via Model Predictive Constant Control". *International Journal of Control (IJC)*.

A4 - Marwa Taleb, Edouard Leclercq & Dimitri Lefebvre. "Hybrid Predictive Control for Timed Elementary Hybrid Petri Nets". *IEEE Transactions on Systems, Man, and Cybernetics : Systems*.

Articles présentés dans des conférences nationales et internationales avec comités de lecture

- C1** - Marwa Taleb, Edouard Leclercq & Dimitri Lefebvre. "Control design of timed Petri nets via Model Predictive Control with continuous Petri nets". *12th IFAC - IEEE International Workshop on Discrete Event Systems - WODES'14*, pages 149-154, Paris, France, May 2014.
- C2** - Marwa Taleb, Edouard Leclercq & Dimitri Lefebvre. "Limitation of flow variation of continuous Petri nets via Model Predictive Control". *American Control Conference - ACC'14*, pages 4919-4924, Portland, United States, June 2014.
- C3** - Marwa Taleb, Edouard Leclercq & Dimitri Lefebvre. "Limitation of flow variation of continuous Petri nets via Model Predictive Control and Lyapunov criterion". *13th European Control Conference - ECC'14*, pages 1825-1830, Strasbourg, France, June 2014.
- C4** - Marwa Taleb, Edouard Leclercq & Dimitri Lefebvre. "Commande prédictive des réseaux de Petri hybrides élémentaires". *6^{ème} Journées Doctorales / Journées Nationales MACS - JDMACS 2015*, Bourges, France, Juin 2015.
- C5** - Marwa Taleb, Edouard Leclercq & Dimitri Lefebvre. "Control design of Elementary Hybrid Petri nets via Model Predictive Control". *5th Analysis and Design of Hybrid Systems - ADHS'15*, pages 41-46, Atlanta, United States, October 2015.

TABLE DES MATIÈRES

Remerciements	v
Liste de Publications	vii
Table des matières	ix
Table des figures	xiii
Liste des tableaux	xvii
Notations	xix
Introduction générale	1
I Modélisation par réseaux de Petri	7
Introduction	8
I.1 Modélisation par réseaux de Petri discrets	8
I.1.1 Présentation du réseau de Petri discret autonome	8
I.1.1.1 Définition du réseau de Petri discret autonome	9
I.1.1.2 Principe d'évolution du réseau de Petri discret autonome .	10
I.1.1.3 Modélisation des comportements des systèmes par réseaux de Petri discrets autonomes	11
I.1.1.4 Propriétés des réseaux de Petri discrets autonomes	14
I.1.2 Présentation du réseau de Petri discret T-temporisé	16
I.1.3 Domaines d'application du réseau de Petri discret	18
I.1.4 Problème d'explosion combinatoire du nombre d'états	18
I.2 Modélisation par réseaux de Petri continus	19
I.2.1 Principe de la fluidification	19

I.2.2	Présentation du réseau de Petri continu autonome	20
I.2.2.1	Définition du réseau de Petri continu autonome	20
I.2.2.2	Principe d'évolution du réseau de Petri continu autonome	21
I.2.3	Présentation du réseau de Petri continu temporisé	22
I.2.4	Domaines d'application du réseau de Petri continu	28
I.2.5	Limitation du réseau de Petri continu	28
I.3	Modélisation par réseaux de Petri hybrides	29
I.3.1	Présentation du réseau de Petri Hybride autonome	29
I.3.1.1	Définition du réseau de Petri hybride autonome	29
I.3.1.2	Principe d'évolution du Réseau de Petri hybride élémen- taire autonome	33
I.3.2	Présentation du réseau de Petri Hybride élémentaire temporisé . . .	34
I.3.3	Domaines d'application du réseau de Petri hybride	37
	Conclusion	38
II	État de l'art : Commande des réseaux de Petri	41
	Introduction	42
II.1	Commande des réseaux de Petri discrets	42
II.1.1	Commande supervisée	43
II.1.2	Commande d'ordonnancement	44
II.2	Commande des réseaux de Petri continus	47
II.2.1	Commande des réseaux de Petri continus sous la sémantique ser- veurs infinis	48
II.2.1.1	Commande multiplicative des réseaux de Petri continus . .	48
II.2.1.2	Commande additive des réseaux de Petri continus	49
II.2.2	Éléments d'état de l'art sur la commande des réseaux de Petri conti- nus sous la sémantique serveurs finis	53
II.2.3	Éléments d'état de l'art sur la commande des réseaux de Petri conti- nus sous la sémantique serveurs infinis	54
II.2.3.1	Commande optimale en régime permanent	55
II.2.3.2	Commande "ON/OFF"	55
II.2.3.3	Commande pour les trajectoires linéaires et linéaires par morceaux	57
II.2.3.4	Commandes "ON/OFF+" et "B-ON/OFF"	58
II.2.3.5	Commande proportionnelle	59
II.2.3.6	Commande prédictive	60
II.3	Commande des réseaux de Petri hybrides	66
II.3.1	Commande des réseaux de Petri hybrides élémentaires	66

II.3.2	Éléments d'état de l'art sur la commande des réseaux de Petri hybrides	68
II.4	Synthèse de l'analyse bibliographique et choix de la commande	70
	Conclusion	72
III	Commande prédictive des réseaux de Petri continus	75
	Introduction	76
III.1	Commande prédictive constante sur l'horizon de prédiction	76
III.1.1	Principe de la commande prédictive	76
III.1.2	Problème du "hill-climbing"	80
III.1.3	Étude de la complexité de calcul	84
III.1.4	Détermination de l'horizon de prédiction maximal	87
III.1.5	Exemple illustratif 1	87
III.2	Limitation de la variation du flux	93
III.2.1	Principe de la commande prédictive avec limitation de la variation du flux	93
III.2.2	Exemple illustratif 2	95
III.3	Étude de stabilité	98
III.3.1	Modification de la fonction coût	98
III.3.2	Étude de la stabilité asymptotique	99
III.3.3	Exemple illustratif 3	102
III.4	Application de la commande prédictive aux réseaux de Petri discrets T- temporisés	105
III.4.1	Principe de la commande prédictive du RdPD T-temporisé	105
III.4.2	Exemple illustratif 4	109
	Conclusion	112
IV	Commande prédictive des réseaux de Petri hybrides élémentaires	115
	Introduction	116
IV.1	Commande prédictive hybride des RdPHE avec des transitions discrètes immédiates	117
IV.1.1	Principe général	117
IV.1.2	Commande prédictive discrète	118
IV.1.3	Commande prédictive continue	120
IV.1.4	Exemple illustratif 1	124
IV.1.5	Commande prédictive hybride avec limitation de la variation du flux	129
IV.1.6	Exemple illustratif 2	131
IV.2	Commande prédictive hybride des RdPHE avec des transitions discrètes temporisées	134

IV.2.1 Principe de la commande prédictive hybride simplifiée	135
IV.2.2 Exemple illustratif 3	137
IV.2.3 Principe de la commande prédictive hybride complète	141
IV.2.4 Exemple illustratif 4	147
Conclusion	149
Conclusion et perspectives	151
Bibliographie	155
Résumé	163
Abstract	165

TABLE DES FIGURES

I.1	Exemple d'un RdPD généralisé	9
I.2	Exemple d'un RdPD avec un conflit structurel	12
I.3	Exemple d'un RdPD avec un conflit effectif	12
I.4	Exemple d'un RdPD avec synchronisation	12
I.5	Structure d'un RdPD avec parallélisme	13
I.6	Structure d'un RdPD avec partage de ressources	13
I.7	Exemple d'un RdP non-borné	14
I.8	Exemple d'un RdP sauf	14
I.9	Exemple d'un RdP avec blocage	15
I.10	Exemple d'un RdP conservatif	16
I.11	Franchissement d'une transition discrète temporisée	17
I.12	Graphe d'atteignabilité du RdP de la figure I.10	19
I.13	Schéma explicatif de la fluidification	20
I.14	Exemple d'un RdPC généralisé	21
I.15	Exemple d'un réseau continu avec plusieurs régions	25
I.16	Projection des régions dans le plan (m_1, m_4)	25
I.17	Exemple d'un RdPHE (à gauche) et d'un RdPHDE (à droite) [DA01]	31
I.18	Exemple d'un RdPHnE	32
II.1	Principe de la supervision	43
II.2	Exemple d'un RdPD	46
II.3	Processus d'exploration partielle du graphe d'atteignabilité pour le RdPD de la figure II.2	47
II.4	Principe de la commande prédictive appliquée au RdPC	63
II.5	Exemple d'un RdPC [MGR ⁺ 08]	64
II.6	Trajectoire du marquage dans le plan (m_2, m_3)	64

III.1	Principe de la commande prédictive constante appliquée au RdPC . . .	78
III.2	Exemple 1 de RdPC	88
III.3	Trajectoire du marquage dans le plan de projection (m_1, m_2) pour $N_c = 1$	88
III.4	Évolution temporelle du critère de performance J pour $N_c = 1$	89
III.5	Trajectoires du marquage, dans le plan de projection (m_1, m_2) , des ap- proches Ap_1, Ap_2 et Ap_3 pour $N_c = 16$	89
III.6	Évolution temporelle du critère de performance J pour N_c adaptatif . .	90
III.7	Évolution temporelle du flux de franchissement X_c pour N_c adaptatif . .	91
III.8	Évolution temporelle du flux de franchissement X_c en présence d'une perturbation	91
III.9	Trajectoire du marquage dans le plan de projection (m_1, m_2) en présence d'une perturbation	92
III.10	Évolution temporelle du critère de performance J en présence d'une perturbation	92
III.11	Exemple 2 de RdPC	95
III.12	Norme de l'accélération et la valeur de A_C lorsque $r_{dX_c} = 0$ (gauche) et $r_{dX_c} = 1$ (droite)	96
III.13	Norme de l'accélération et la valeur de A_C lorsque $r_{dX_c} = 2$	96
III.14	Variation du flux de la transition t_3 pour différentes valeurs de r_{dX_c} . . .	97
III.15	Évolution temporelle du marquage M_c pour $r_{dX_c} = 0$ (gauche) et $r_{dX_c} =$ 2 (droite)	97
III.16	Évolution temporelle de r_{dX_c} adapté avec (III.32)	103
III.17	Évolution temporelle de l'horizon de prédiction N_c	103
III.18	Évolution de l'erreur quadratique du marquage lorsque $r_{dX_c} = 1$ (gauche) et r_{dX_c} est adaptatif (droite)	103
III.19	Évolution du marquage lorsque r_{dX_c} est adaptatif	104
III.20	Trajectoire du marquage continu dans le plan de projection (m_1, m_2) lorsque r_{dX_c} est adaptatif	104
III.21	Exemple d'un RdPD T-temporisé sans blocage	110
III.22	Évolution temporelle du flux cumulé X_{cum}	110
III.23	Évolution temporelle de l'erreur quadratique du marquage continu J_M .	110
III.24	Trajectoire du marquage continu et discret dans le plan de projection (m_1, m_4)	111
IV.1	Principe de la commande prédictive hybride	117
IV.2	Principe de la commande prédictive continue constante	121
IV.3	Système de trois réservoirs connectés	124
IV.4	RdPHD qui modélise le système de trois réservoirs connectés	126

IV.5	Commande continue du RdPHDE	127
IV.6	Évolution temporelle du marquage continu M_c du RdPHDE	128
IV.7	Évolution temporelle du flux continu X_c du RdPHDE	128
IV.8	Commande continue du RdPHDE lorsque $r_{dX_c} = 0.5$	132
IV.9	Évolution temporelle du marquage continu M_c du RdPHDE lorsque $r_{dX_c} = 0.5$	133
IV.10	Norme de l'accélération du RdPHDE lorsque $r_{dX_c} = 0$ (gauche) et $r_{dX_c} =$ 0.5 (droite)	134
IV.11	Commande continue du RdPHDE lorsque $r_{dX_c} = 0.5$	139
IV.12	Évolution temporelle du marquage continu du RdPHDE lorsque $r_{dX_c} = 0.5$	140
IV.13	Norme de l'accélération lorsque $r_{dX_c} = 0$ (gauche) et $r_{dX_c} = 0.5$ (droite)	140
IV.14	Décomposition de l'intervalle de la prédiction continue	141
IV.15	Commande continue constante par sous-intervalle	143
IV.16	Commande continue du RdPHDE lorsque $r_{dX_c} = 0.5$	147
IV.17	Évolution temporelle du marquage continu du RdPHDE lorsque $r_{dX_c} = 0.5$	148
IV.18	Norme de l'accélération lorsque $r_{dX_c} = 0$ (gauche) et $r_{dX_c} = 0.5$ (droite)	148

LISTE DES TABLEAUX

I.1	Domaines d'application des RdPH	38
II.1	Comparaison entre les différentes techniques de commande [SJM11]	70
III.1	Complexité de la commande prédictive qui utilise le critère (III.1) sans tenir compte du changement de région	85
III.2	Complexité de la commande prédictive qui utilise le critère (III.1) en tenant compte du changement de région	85
III.3	Complexité de la commande prédictive qui utilise le critère (II.17) sans tenir compte du changement de région	85
III.4	Complexité de la commande prédictive qui utilise le critère (II.17) en tenant compte du changement de région	86
III.5	Les performances des approches Ap_1 , Ap_2 et Ap_3	90

Symboles mathématiques

\mathbb{N}	Ensemble des nombres entiers.
\mathbb{R}	Ensemble des nombres réels.
\mathbb{Z}	Ensemble des entiers relatifs.
*	Étoile de Kleene.
<i>diag</i>	Matrice diagonale.
V^t	Transposé du vecteur (ou de la matrice) V .
$ E $	Cardinal de l'ensemble E .
\dot{X}	Dérivée de X .
$\lfloor X \rfloor$	Partie entière par défaut de X .
$\lceil X \rceil$	Partie entière par excès de X .
$\{X\}$	Partie décimale de X .

Abréviations

B-ON/OFF	Balanced ON/OFF.
DLSS	Dynamic Look-Ahead Stage Search.
IB-state	Invariant Behavior state.
MPC	Model Predictive Control.
ON/OFF+	ON/OFF plus.
RdP	Réseau de Petri.
RdPC	Réseau de Petri continu.
RdPD	Réseau de Petri discret.
RdPH	Réseau de Petri hybride.

RdPHDE	Réseau de Petri hybride D-Élémentaire.
RdPHE	Réseau de Petri hybride Élémentaire.
RdPHnE	Réseau de Petri hybride non-Élémentaire.
RTA*	Real Time A*.
SDH	Systèmes dynamiques hybrides.
SED	Systèmes à événements discrets.
TOR	Tout ou rien.
U.T.	Unité de temps.

Variables

$0_{n \times q}$	Matrice nulle de dimension $n \times q$.
$\eta_d (\eta_c)$	Degré de validation d'une transition discrète (continue).
$\Gamma_d (\Gamma_c, \Gamma_h)$	Vecteur de franchissements des transitions discrètes (continues, hybrides).
λ	Vecteur des vitesses maximales de franchissement des transitions continues.
Λ	Matrice diagonale de λ .
$\mathcal{N}_d (\mathcal{N}_c, \mathcal{N}_h)$	Structure du réseau de Petri discret (continu, hybride).
σ_d	Séquence de commandes discrètes.
$\sigma_d(s)$	$s^{\text{ème}}$ transition de la séquence σ_d .
τ	Temps continu.
τ^+	Instant juste après le franchissement des transitions discrètes à la date τ .
τ_k	Temps échantillonné.
θ	Période d'échantillonnage.
${}^\circ t_j ({}^\circ p_j)$	Places (transitions) en amont de t_j (de p_i).
$t_j^\circ (p_j^\circ)$	Places (transitions) en aval de t_j (de p_i).
A_k	Matrice de contraintes de la région R_k .
A_C	Accélération cumulée multipliée par θ .
A_{acc}	Vecteur d'accélération.
D_{min}	Vecteur des durées minimales de franchissement des transitions discrètes.
F_c	Vecteur flux maximal de franchissement.
H	Longueur d'une séquence σ_d .
$I_{n \times q}$	Matrice identité de dimension $n \times q$.
J	Critère de performance global.
J_M	Erreur quadratique sur le marquage continu.
J_X	Erreur quadratique sur le flux continu.
J_{dX_c}	Terme qui prend en compte la variation du flux.
J_{MU_c}	Partie du critère J_M qui dépend du vecteur de commandes continues U_c .

J_{U_c}	Partie du critère global J qui dépend du vecteur de commandes continues U_c .
$J_{X_{U_c}}$	Partie du critère J_X qui dépend du vecteur de commandes continues U_c .
K	Nombre total de régions.
$M_d^{tot} (M_c^{tot})$	Séquence de marquages discrets (continus).
$M_d (M_c, M_h)$	Marquage des places discrètes (continues, hybrides).
$M_{0d} (M_{0c}, M_{0h})$	Marquage initial des places discrètes (continues, hybrides).
$M_{refd} (M_{refc})$	Marquage discret (continu) désiré.
n	Nombre de places.
$N_d (N_c)$	Horizon de prédiction discret (continu).
$n_d (n_c)$	Nombre de places discrètes (continues).
P	Ensemble de places.
$P_d (P_c)$	Ensemble des places discrètes (continues).
q	Nombre de transitions.
$q_d (q_c)$	Nombre de transitions discrètes (continues).
r_{dX_c}	Facteur de pondération du terme J_{dX_c} .
$R(\tau) (R_k)$	Région du marquage $M_c(\tau) (M_c(\tau_k))$.
$R_c(\mathcal{N}_c, M_{0c})$	Espace d'atteignabilité du réseau de Petri continu autonome.
$R_d(\mathcal{N}_d, M_{0d})$	Espace d'atteignabilité du réseau de Petri discret autonome.
$R_h(\mathcal{N}_h, M_{0h})$	Espace d'atteignabilité du réseau de Petri hybride autonome.
T	Ensemble de transitions.
T^{comm}	Ensemble de transitions commandables.
$T_d (T_c)$	Ensemble des transitions discrètes (continues).
$T_d^{comm} (T_c^{comm})$	Ensemble de transitions discrètes (continues) commandables.
U_c	Vecteur de commandes continues.
W_{po}	Matrice d'incidence arrière.
$W_{po}(i, j)$	Poids de l'arc de la transition t_j vers la place p_i .
W_{pr}	Matrice d'incidence avant.
$W_{pr}(i, j)$	Poids de l'arc de la place p_i vers la transition t_j .
X_c	Flux en régime forcé des transitions continues.
X_d	Vecteur de commandes discrètes.
X_{cum}	Flux cumulé multiplié par θ .

INTRODUCTION GÉNÉRALE

Les systèmes à événements discrets (SED) représentent une sous-classe des systèmes dynamiques pour lesquels les variables d'état évoluent de manière discrète au cours du temps. Cette évolution est conditionnée par l'occurrence d'événements ponctuels. Ces systèmes couvrent aujourd'hui une classe importante d'applications réelles, notamment les processus manufacturiers, les systèmes informatiques, logistiques et de transport ainsi que les réseaux de télécommunications.

La mise en œuvre d'une commande permettant à ces systèmes de fonctionner de manière efficace et satisfaisante, vis-à-vis des spécifications établies par le cahier des charges, est un des principaux défis du monde industriel. Récemment, plusieurs facteurs tels que la progression rapide de la technologie, l'augmentation des demandes et des exigences du marché ainsi que la concurrence accrue entre les entreprises, ont directement contribué à l'accroissement de la complexité des SED, de plus, il est impossible de réaliser l'ensemble des tests de commande sur les systèmes réels. La mise en disposition d'un outil de modélisation formel qui permet d'appréhender le comportement du système réel dans le but de l'analyser, le commander et le simuler, s'avère être un enjeu crucial. Cet axe de recherche a attiré l'attention des communautés qui s'intéressent à la théorie des systèmes. Divers outils de modélisation des SED ont été développés, les plus connus sont les automates à états finis et les réseaux de Petri (RdP). Dans le cadre de cette thèse, nous nous intéressons à la modélisation et à la commande des SED par l'outil RdP.

Le RdP a été proposé en 1962 par le mathématicien allemand Carl Adam Petri pendant sa thèse de doctorat. Cet outil de modélisation représente un formalisme mathématique puissant qui bénéficie d'une représentation graphique intuitive et modulaire [Sil13]. Il possède également la capacité de représenter explicitement certains comportements du système dont le parallélisme, les structures de choix, la synchronisation et le par-

tage de ressources. Actuellement, le RdP joue un rôle important dans la modélisation du comportement dynamique, l'analyse de performance, la synthèse et l'implémentation de la commande des SED [Sil93, Sil13], spécialement des systèmes de production [MYK02, YRCL03b, KSN07], et de transport [LC04, DBM⁺14].

L'un des objectifs de commande des SED, modélisés par RdP, consiste à déterminer une séquence de franchissements représentant la trajectoire qui permet d'atteindre un état désiré à partir d'un état initial. Généralement, les techniques de commande proposées dans la littérature utilisent des fonctions heuristiques afin de générer la portion du graphe d'atteignabilité qui contient cette trajectoire [LD94, MYK02]. Ces méthodes ne sont malheureusement pas applicables aux systèmes possédant un grand nombre d'états, car le graphe d'atteignabilité devient extrêmement volumineux. Elles peuvent mener au problème d'explosion combinatoire du nombre d'états. Afin de contourner ce problème, l'équipe de David et Alla ainsi que celle de Silva ont proposé en 1987 une technique de relaxation appelée fluidification. Elle permet d'obtenir un réseau de Petri continu (RdPC) dont le comportement approche celui du réseau de Petri discret (RdPD), et ainsi de pouvoir bénéficier des outils de l'automatique continue [SJM^V11].

Le but de la commande des RdPC est très fortement inspiré des contributions de la commande des systèmes à temps continu. Il s'agit de piloter le système jusqu'à atteindre une consigne [SR04]. Plusieurs méthodes ont été proposées dans la littérature afin de garantir la stabilité du système et sa convergence vers l'état final, toutefois la plupart d'entre elles sont contraignantes. En effet, certaines ne sont applicables que pour une classe particulière de modèles, dans le but de réduire la complexité de calcul, tels que les RdPC sans choix [WMJS10] ou les RdPC sans synchronisation [XRS06, XRS08a]. D'autres sont motivées par un problème précis de commande, qui est la minimisation du temps de convergence [XRS08b, AOJMS11, WMJS12], mais qui entraînent de fortes variations de flux. Ces fortes variations de flux sont traduites, dans la réalité, par une sollicitation excessive des actionneurs qui provoque un vieillissement précoce des composants. Par ailleurs, en considérant des sous-classes particulières de RdPC et en imposant souvent des trajectoires linéaires (ou linéaires par morceaux), le champ d'application de la commande développée reste limité et certains comportements particuliers ne peuvent être traités comme les comportements de "hill-climbing", correspondant à une trajectoire devant nécessairement s'éloigner de la cible avant de pouvoir y converger.

Le modèle RdPC ne permet pas de représenter les conditions logiques ni les actions discrètes, comme l'ouverture ou la fermeture d'une vanne, l'occurrence d'une panne ou le partage de ressources [GA07]. Le réseau de Petri Hybride (RdPH) a été proposé dans [DA01]

pour prendre en compte ces actions discrètes et modéliser les systèmes dynamiques hybrides qui font intervenir explicitement et simultanément des dynamiques continues et des dynamiques événementielles. Mais, les lois de commande précédentes ne sont pas adaptées pour prendre en compte les actions discrètes et aucune d'entre elles n'exploite la commande continue pour déterminer une séquence de commandes discrètes.

C'est dans ce contexte que nous proposons, à travers ce mémoire, une méthode de commande inspirée de la commande prédictive. Grâce à sa robustesse face aux perturbations, sa formulation du problème d'optimisation qui anticipe et empêche toute violation des contraintes et ses performances, la commande prédictive est largement utilisée en industrie; elle intervient dans divers domaines tels que l'automobile, la pétrochimie et les applications aérospatiales [BMDP02, QB03]. Dans ce travail, nous souhaitons tirer parti de ces avantages. De plus, elle optimise une fonction quadratique qui peut combiner facilement plusieurs objectifs de commande ce qui permet de traiter le problème de la sollicitation excessive des actionneurs. Cette méthodologie a d'ailleurs déjà été utilisée pour la commande des systèmes dynamiques modélisés par RdPC [GMR⁺06, MGR⁺08], RdPD [LL15] ainsi que par RdPH [JCBM13]. En revanche, la complexité et le temps de calcul sont deux questions clés qui limitent l'utilisation de la commande prédictive. Ils sont directement liés à la taille du RdPC et à la largeur de l'horizon de prédiction, ce qui a conduit la plupart des contributions à utiliser un nombre réduit de pas de prédiction [GMR⁺06]. Mais parfois, il est primordial de prédire suffisamment loin pour détecter le changement de comportement du système commandé, notamment lors de situations de type "hill-climbing".

Une première contribution consiste à apporter des simplifications indispensables dans l'algorithme de calcul de la commande prédictive continue et à proposer une adaptation de l'horizon de prédiction en fonction de la nature de la trajectoire, afin d'utiliser un horizon de prédiction large tout en limitant l'effort de calcul. Une étude de complexité est proposée, en se basant sur cette étude, nous évoquons une méthode qui détermine la borne supérieure de l'horizon de prédiction afin que la solution proposée soit réalisable en fonction de la période d'échantillonnage du calculateur.

La deuxième contribution est motivée par la nécessité de réduire la sollicitation excessive des actionneurs. Elle propose une modification de la fonction de performance en ajoutant un terme pondéré qui tient compte de la variation de flux. Une détermination systématique du facteur de pondération qui assure la stabilité du système autour de l'état désiré est proposée.

La troisième contribution consiste à proposer un algorithme qui exploite la commande prédictive développée pour commander les RdPC afin de déterminer la séquence de commandes discrètes des RdPD, sans établir le graphe d'atteignabilité et faire de recherche exhaustive dans l'espace discret des marquages atteignables.

La quatrième contribution vise à développer une approche hybride qui commande les RdPH, munis de transitions discrètes immédiates et temporisées et qui limite la sollicitation excessive des actionneurs continus. La commande suggérée est inspirée du principe de la commande prédictive, elle adapte la commande prédictive continue à la structure du RdPH en tenant compte de l'impact des actions discrètes sur la partie continue, puis la combine avec la commande prédictive discrète proposée dans [LL15].

Cette thèse est composée de quatre chapitres.

Le premier chapitre présente les définitions et les concepts fondamentaux relatifs à l'outil de modélisation et d'analyse RdP. Dans une première partie de ce chapitre, nous présentons le RdPD et quelques domaines d'application. Nous expliquons également le problème d'explosion combinatoire du nombre d'états inhérent au RdPD et nous introduisons ensuite la fluidification. Dans la seconde partie, nous présentons en détails les RdPC et les RdPH. Nous passons en revue leur définition, leur représentation ainsi que leur principe d'évolution sous la sémantique retenue dans nos travaux de recherche. L'intérêt et l'efficacité de la modélisation par ces réseaux sont également mis en évidence par un ensemble d'applications réelles.

Le deuxième chapitre établit un état de l'art sur la commande des trois extensions du RdP introduites dans le premier chapitre et situe le contexte général en précisant la problématique de nos travaux de recherche. Dans la première partie, nous exposons brièvement quelques commandes de RdPD en insistant sur leurs limites de mise en œuvre. La deuxième partie de ce chapitre étudie la commande des RdPC à vitesses variables. Nous faisons d'abord une description générale de la problématique de commande des RdPC. Nous présentons ensuite les approches de commande les plus performantes parmi celles proposées dans la littérature. Nous exposons leur principe, leurs avantages ainsi que leurs principaux inconvénients, dans ce descriptif, nous nous focalisons sur la commande prédictive. Finalement, nous proposons quelques éléments bibliographiques sur la commande des RdPH, notamment sur les approches basées sur la commande prédictive. Afin de mettre en évidence les avantages de cette commande, une étude comparative est présentée à la fin de cette partie. Dans cette étude, nous détaillons aussi ses inconvénients et les problèmes rencontrés dans la résolution de la problématique posée.

Le troisième chapitre est dédié à l'amélioration de la commande prédictive pour répondre à la problématique posée dans le chapitre précédent. Une première partie de ce chapitre concerne la commande prédictive des RdPC. Elle détaille l'ensemble des simplifications apportées à cette commande. En outre, elle présente la procédure d'adaptation de l'horizon de prédiction, une étude de complexité est ensuite menée. Une nouvelle méthode qui limite la sollicitation excessive des actionneurs est aussi proposée. Elle est suivie d'une étude de stabilité autour de l'état souhaité. Nous montrerons, dans une deuxième partie, de quelle façon la commande prédictive proposée peut être utilisée pour commander les RdPD. Des simulations numériques sont présentées dans chaque partie pour illustrer l'utilité des différentes contributions de ce chapitre.

Dans le quatrième et dernier chapitre, nous introduisons la commande prédictive hybride pour commander les RdPH et limiter la sollicitation excessive des actionneurs. Nous simplifions le problème de commande en supposant que toutes les transitions discrètes sont immédiates. Nous présentons la commande développée sous la forme d'une combinaison d'une commande prédictive discrète et de la commande prédictive continue développée dans le chapitre précédent. Nous détaillons le principe de cette approche ainsi que l'algorithme global de commande. Dans un second temps, nous présentons les modifications subies par cet algorithme afin de considérer les contraintes temporelles discrètes. L'efficacité de l'approche proposée est mise en relief par un ensemble de simulations numériques.

Finalement, ce mémoire de thèse se termine avec une conclusion générale qui synthétise les différentes contributions mises en place. Des perspectives de poursuite de ces travaux de recherche sont également identifiées.

Sommaire

Introduction	8
I.1 Modélisation par réseaux de Petri discrets	8
I.1.1 Présentation du réseau de Petri discret autonome	8
I.1.1.1 Définition du réseau de Petri discret autonome	9
I.1.1.2 Principe d'évolution du réseau de Petri discret autonome	10
I.1.1.3 Modélisation des comportements des systèmes par ré-	
seaux de Petri discrets autonomes	11
I.1.1.4 Propriétés des réseaux de Petri discrets autonomes . .	14
I.1.2 Présentation du réseau de Petri discret T-temporisé	16
I.1.3 Domaines d'application du réseau de Petri discret	18
I.1.4 Problème d'explosion combinatoire du nombre d'états	18
I.2 Modélisation par réseaux de Petri continus	19
I.2.1 Principe de la fluidification	19
I.2.2 Présentation du réseau de Petri continu autonome	20
I.2.2.1 Définition du réseau de Petri continu autonome	20
I.2.2.2 Principe d'évolution du réseau de Petri continu autonome	21
I.2.3 Présentation du réseau de Petri continu temporisé	22
I.2.4 Domaines d'application du réseau de Petri continu	28
I.2.5 Limitation du réseau de Petri continu	28
I.3 Modélisation par réseaux de Petri hybrides	29
I.3.1 Présentation du réseau de Petri Hybride autonome	29
I.3.1.1 Définition du réseau de Petri hybride autonome	29
I.3.1.2 Principe d'évolution du Réseau de Petri hybride élé-	
mentaire autonome	33
I.3.2 Présentation du réseau de Petri Hybride élémentaire temporisé	34
I.3.3 Domaines d'application du réseau de Petri hybride	37
Conclusion	38

Introduction

EN 1962, le réseau de Petri (RdP) a vu le jour dans la thèse de doctorat du mathématicien allemand Carl Adam Petri (1926-2010) [Sil13]. Cet outil représente un formalisme mathématique puissant qui bénéficie d'une représentation graphique intuitive et modulaire. Il a été initialement dédié à la description des différentes séquences d'événements qui se produisent dans les protocoles de communication [Pet66]. Depuis ce jour, la modélisation par RdP s'est largement répandue dans le monde entier. En revanche, l'exploration systématique et exhaustive des différents états atteints par le système durant son fonctionnement limite l'utilisation de cet outil pour les systèmes de grande dimension. Afin de pallier ce problème, connu sous le nom d'explosion combinatoire du nombre d'états, il est proposé d'étudier les événements discrets via des approximations continues. Cette technique de relaxation s'appelle "fluidification" et remplace le RdP discret (RdPD) par un RdP continu (RdPC) [SR02]. Néanmoins, un tel modèle ne permet pas de représenter les conditions logiques ni les actions discrètes, comme l'ouverture ou la fermeture d'une vanne, l'occurrence d'une panne ou le partage de ressources [GA07]. Pour cette raison, le réseau de Petri Hybride (RdPH) a été défini dans [DA01]. Il constitue un formalisme puissant qui permet de modéliser les systèmes dynamiques hybrides qui font intervenir à la fois des dynamiques continues et événementielles en interaction [GA07]. Ce type de réseau permet de décrire les changements logiques qui peuvent survenir.

Ce premier chapitre fait appel aux concepts fondamentaux relatifs aux trois extensions du RdP et introduit les notations qui seront utilisées tout au long du mémoire. La section I.1 donne un aperçu global sur le RdPD autonome et temporisé ainsi que ses domaines d'application. Elle explique également la notion de "fluidification". La section I.2 se concentre sur le RdPC. Le RdPH est finalement présenté dans la section I.3. Il est défini comme une interaction entre un RdPC et un RdPD.

I.1 Modélisation par réseaux de Petri discrets

I.1.1 Présentation du réseau de Petri discret autonome

Ce paragraphe concerne essentiellement le RdPD autonome. En d'autres termes, la notion du temps ne sera pas évoquée. Par conséquent, un RdPD autonome modélise un système dynamique dont les dates d'occurrence des événements ne sont pas modélisées. Seul l'ordre des événements est pris en compte.

I.1.1.1 Définition du réseau de Petri discret autonome

Le RdP est un graphe orienté biparti composé de deux sortes d'éléments à savoir, les places et les transitions. Des arcs orientés sont mis en places afin de relier ces éléments entre eux [Sma13]. Les éléments de même nature ne peuvent pas être directement raccordés. Des arcs orientés issus d'une (ou de plusieurs) place(s) aboutissant à une (ou plusieurs) transition(s) et réciproquement. Ces arcs peuvent être pondérés. Dans ce cas, le RdP est dit généralisé.

Du point de vue de la modélisation, les places correspondent aux variables d'état du système modélisé et les transitions définissent les événements ou les actions qui entraînent l'évolution des variables d'état. Dans le cas d'un RdP marqué, la valeur de la variable d'état correspond au nombre de marques (ou de jetons) dans chaque place. Ce nombre représente le marquage de la place concernée. L'ensemble des marquages de toutes les places forme le vecteur de marquages qui représente l'état global du système étudié. Étant donné que le nombre de marques porte une information spécifiant le nombre de ressources disponibles, le poids d'un arc dirigé vers une transition précise le nombre de ressources indispensables à la réalisation de l'événement associé à cette transition. De même, un arc dont l'origine est une transition, caractérise le nombre de ressources libérées suite à l'occurrence de l'événement correspondant [Sil13].

Du point de vue de la représentation graphique, les ensembles des places et des transitions d'un RdPD sont respectivement représentés par des cercles et des barres simples. Par ailleurs, les arcs orientés sont représentés par des flèches indiquant le sens de circulation des jetons. Le poids de chaque arc est un entier strictement positif. Généralement, il est indiqué au milieu de l'arc. Par défaut, le poids vaut 1. Si les poids de tous les arcs sont égaux à 1, le RdPD est dit ordinaire. Finalement, les marques sont symbolisées par des disques pleins noirs placés à l'intérieur de chaque place. La figure I.1 présente un exemple simple de RdPD généralisé.

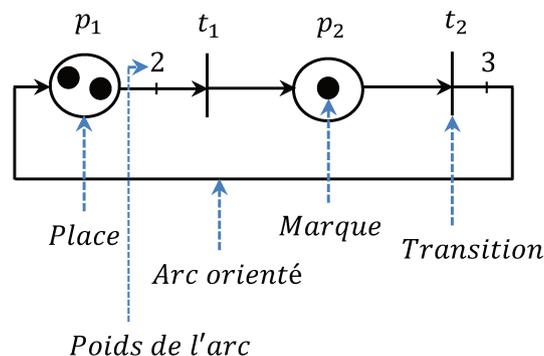


FIGURE I.1 – Exemple d'un RdPD généralisé

Pour terminer, du point de vue mathématique, le RdPD peut être formellement vu comme un couple $\langle \mathcal{N}_d, M_{0d} \rangle$ où \mathcal{N}_d définit la structure du réseau discret et $M_{0d} \in \mathbb{N}_{\geq 0}^n$ spécifie le marquage initial qui correspond à l'état initial du système étudié. A titre d'exemple, le marquage initial du RdPD donné par la figure I.1 est $M_{0d} = \begin{bmatrix} 2 & 1 \end{bmatrix}^t$ où X^t dénote la transposée d'un vecteur ou d'une matrice X . La structure \mathcal{N}_d est représentée par un quadruplet $\mathcal{N}_d = \langle P, T, W_{pr}, W_{po} \rangle$ où $P = \{p_i\}_{i=1, \dots, n}$ et $T = \{t_j\}_{j=1, \dots, q}$ sont deux ensembles finis non-vides composés respectivement de $n = |P|$ places et $q = |T|$ transitions tels que $P \cap T = \emptyset$. La notation $|\cdot|$ désigne le cardinal d'un ensemble. Les deux matrices W_{pr} et $W_{po} \in \mathbb{N}_{\geq 0}^{n \times q}$ désignent respectivement la matrice de pré-incidence et de post-incidence. Ces dernières définissent la structure statique du RdP : $\forall (p_i, t_j) \in P \times T$, $W_{pr}(i, j) \in \mathbb{N}_{\geq 0}$ et $W_{po}(i, j) \in \mathbb{N}_{\geq 0}$ représentent respectivement le poids de l'arc de la place p_i vers la transition t_j et de la transition t_j vers la place p_i . La matrice W , définie par $W = W_{po} - W_{pr}$, désigne la matrice d'incidence du réseau. Chaque élément $v \in P \cup T$, possède des éléments d'entrée (placés en amont) ainsi que des éléments de sortie (situés en aval) qui sont respectivement symbolisés par ${}^\circ v$ et v° .

Le RdPD ne reste pas figé dans son état initial. À partir de M_{0d} , le franchissement d'une transition influence l'état du système et lui permet d'évoluer et d'atteindre un autre état. Le marquage de chaque place p_i est dénoté $m_i \in \mathbb{N}_{\geq 0}$. Ainsi, le vecteur constitué du marquage de toutes les places présentes dans le réseau est désigné par $M_d = \begin{bmatrix} m_1 & \dots & m_i & \dots & m_n \end{bmatrix}^t$. Les conditions d'évolution d'un RdPD autonome seront détaillées dans le paragraphe suivant.

I.1.1.2 Principe d'évolution du réseau de Petri discret autonome

Le franchissement (ou le tir) d'une transition entraîne l'évolution du RdP. Cette évolution consiste à déplacer les jetons au sein du réseau et à modifier le marquage de chaque place, ceci s'interprète comme la consommation et la production de ressources. Le franchissement est soumis à certaines conditions. Avant d'être franchie, la transition doit être validée. La validation d'une transition dépend du nombre de marques dans les places en amont. Soit $\eta_d(t_j, M_d)$ le degré de validation d'une transition t_j , défini par (I.1) :

$$\eta_d(t_j, M_d) = \min_{p_i \in {}^\circ t_j} \left\{ \left\lfloor \frac{m_i}{W_{pr}(i, j)} \right\rfloor \right\} \quad (\text{I.1})$$

où $\lfloor \cdot \rfloor$ symbolise la partie entière par défaut de (\cdot) . Une transition discrète t_j est dite validée par le marquage courant M_d si et seulement si (ssi) $\eta_d(t_j, M_d) > 0$, autrement dit, ssi le nombre de jetons dans toutes les places p_i en amont de t_j ($p_i \in {}^\circ t_j$) est au moins égal

au poids des arcs de p_i à t_j . En particulier, dans le cas d'un RdP ordinaire, il faut que toutes les places en amont soient marquées. D'une manière générale, $\eta_d(t_j, M_d)$ mesure la quantité maximale de franchissements d'une transition t_j validée : t_j ne peut être franchie qu'avec une quantité entière de marque $\alpha_d \in \mathbb{N}$ telle que $0 < \alpha_d \leq \eta_d(t_j, M_d)$. Il résulte du franchissement de t_j le retrait de $\alpha_d \cdot W_{pr}(i, j)$ marques de toutes les places p_i en amont et l'ajout de $\alpha_d \cdot W_{po}(i, j)$ marques à chacune des places p_i en aval. La variation de marquage est définie par (I.2) :

$$\Delta M_d = \alpha_d \cdot W(:, j) \quad (\text{I.2})$$

avec $W(:, j)$ est la $j^{\text{ème}}$ colonne de la matrice W relative à la transition t_j .

Afin de savoir dans quel ordre les transitions ont réellement eu lieu, il est supposé qu'une seule transition ne peut être franchie à la fois. Soit $\sigma_d \in T^*$ (où $*$ représente l'étoile de Kleene), la séquence de franchissements définie comme une série de transitions successivement franchies à partir d'un état initial M_{0d} jusqu'à l'atteinte d'un état M_d . La représentation mathématique résumant cette évolution est donnée par $M_{0d}[\sigma_d > M_d$. Une telle séquence est caractérisée par un vecteur de franchissements noté $\Gamma_d \in \mathbb{N}_{\geq 0}^q$ tel que $\Gamma_d = (\gamma_{dj})_{j=1, \dots, q} \in \mathbb{N}_{\geq 0}^q$ où $\gamma_{dj} \in \mathbb{N}_{\geq 0}$ désigne le nombre d'occurrences de la transition t_j dans la séquence σ_d .

L'espace d'atteignabilité d'un RdPD autonome est désigné par $R_d(\mathcal{N}_d, M_{0d})$. Il représente l'ensemble de tous les marquages M_d qui sont atteignables (accessibles) à partir du marquage initial à travers une séquence finie σ_d . Il est défini par $R_d(\mathcal{N}_d, M_{0d}) = \{M_d \mid M_{0d}[\sigma_d > M_d\}$. Si $M_d \in R_d(\mathcal{N}_d, M_{0d})$, l'équation fondamentale (ou équation d'état) de variation du marquage s'écrit sous la forme (I.3) :

$$M_d = M_{0d} + W \cdot \Gamma_d \quad (\text{I.3})$$

I.1.1.3 Modélisation des comportements des systèmes par réseaux de Petri discrets autonomes

Le RdP possède la capacité de représenter explicitement certains comportements du système dont les structures de choix, la synchronisation, le parallélisme et le partage de ressources.

Les RdP permettent de représenter facilement le concept de choix lorsqu'une place p_i possède plusieurs transitions de sortie : $|p_i^\circ| > 1$ le conflit est dit structurel (voir figure I.2). Quand la place p_i contient un nombre insuffisant de marques pour franchir toutes les transitions de sortie, le conflit est dit effectif.

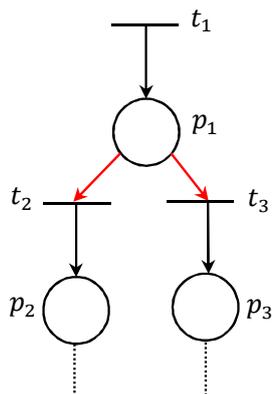


FIGURE I.2 – Exemple d’un RdPD avec un conflit structurel

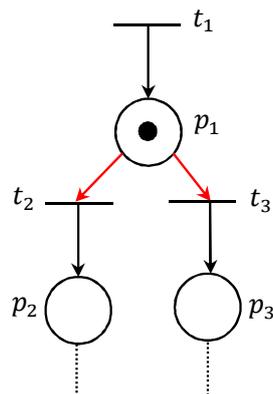


FIGURE I.3 – Exemple d’un RdPD avec un conflit effectif

La synchronisation (ou la structure de rendez-vous) permet de synchroniser les opérations de deux processus. Ce concept est représenté par le RdP lorsqu’une transition t_j possède plusieurs places d’entrée : $|\circ t_j| > 1$ (voir figure I.4). Le franchissement de la transition t_1 ne peut avoir lieu que si les places p_3 et p_2 , qui appartiennent respectivement aux processus 1 et 2, contiennent au moins une marque. Dans le cas contraire, si à titre d’exemple p_2 n’est pas marquée, la transition t_1 reste bloquée en attendant que le processus 2 évolue et que p_2 soit marquée.

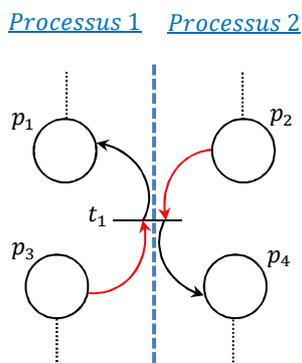


FIGURE I.4 – Exemple d’un RdPD avec synchronisation

Le parallélisme représente l’évolution simultanée de deux ou plusieurs processus au sein d’un même système (voir figure I.5). Le départ simultané de l’évolution des processus est provoqué par la transition t_1 ayant les deux places p_2 et p_3 en sortie. Si p_1 est marquée, le franchissement de t_1 met une marque dans chacune des places de sortie, ce qui permet de déclencher respectivement les processus 1 et 2. La fin des deux processus est indiquée par la présence d’une marque dans chacune des places p_{22} et p_{23} . D’après la figure I.5, l’achèvement des deux processus est synchronisé à l’aide de la transition t_{12} .

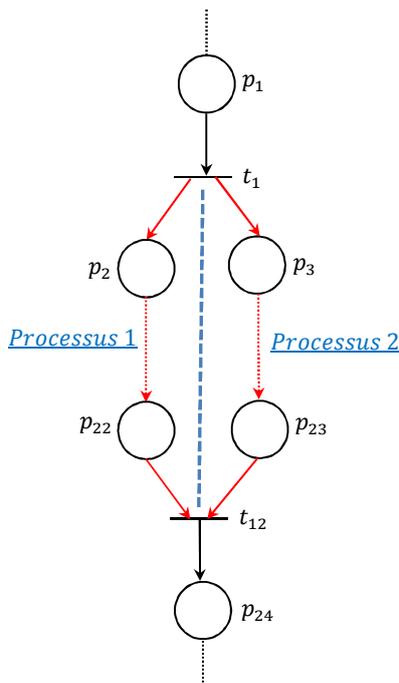


FIGURE I.5 – Structure d’un RdPD avec parallélisme

Finalement, le partage de ressources modélise la mise en commun d’une même ressource entre plusieurs processus au sein d’un même système tel que le partage d’un espace mémoire entre deux programmes informatiques. Un exemple de partage de ressources est présenté par la figure I.6. Le franchissement de la transition t_1 du processus 1 consomme la marque présente dans p_1 . La ressource est donc allouée au processus 1, ce qui provoque le blocage de l’évolution du processus 2 sur la place p_{12} . À la fin de l’évolution du processus 1, lorsque la transition t_{10} est franchie, p_1 redevient marquée et la ressource est à nouveau disponible.

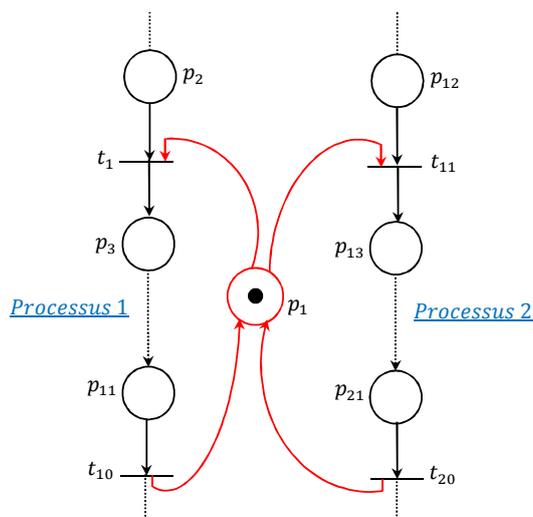


FIGURE I.6 – Structure d’un RdPD avec partage de ressources

I.1.1.4 Propriétés des réseaux de Petri discrets autonomes

Ce paragraphe présente les principales propriétés du RdP qui sont liées à l'aspect borné ainsi qu'aux concepts de binarité, de vivacité, de blocage, de persistance, de conservation et de consistance.

Une place p_i est dite bornée pour un marquage initial M_{0d} si pour toute séquence réalisable à partir de M_{0d} , le nombre de marques dans p_i reste borné. Un RdPD est dit borné si toutes ses places sont bornées. Un RdP est dit structurellement borné s'il est borné pour tout marquage initial [DA10]. Le RdP la figure I.7 est un RdP non-borné. Pour le marquage initial $M_{0d} = [1 \ 2]^t$, la transition t_1 est toujours validée, elle peut être franchie continûment. À chaque franchissement de t_1 , une nouvelle marque est ajoutée dans la place p_2 . En conséquence, m_2 peut tendre vers l'infini.

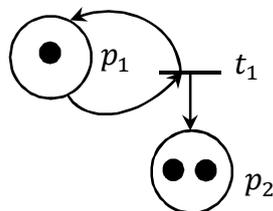


FIGURE I.7 – Exemple d'un RdP non-borné

Un cas particulier du RdP borné est le RdP sauf (ou binaire). Un RdP est dit sauf pour un marquage initial M_{0d} si pour tous les marquages atteignables à partir de M_{0d} , les places contiennent au plus une marque [DA10]. Le RdP représenté sur la figure I.8 est un RdP sauf.

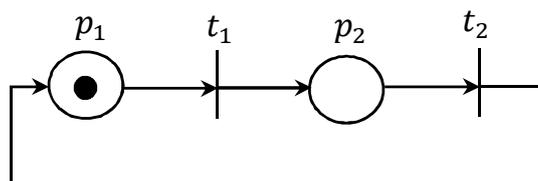


FIGURE I.8 – Exemple d'un RdP sauf

L'évolution du RdP est liée aux franchissements des transitions. Si certaines transitions ne sont pas franchies, cela indique que les événements associés à ces transitions n'ont pas eu lieu. En conséquence, la partie du système modélisée par le sous-RdP ne fonctionne pas correctement. Un problème de conception du système est donc détecté. La suite de ce paragraphe étudie les concepts relatifs à cette problématique.

Une transition t_j est dite vivante pour le marquage initial M_{0d} si à partir de tout marquage atteignable M_d , il existe une séquence de franchissements qui contient t_j [DA10] :

$$\forall M_d \in R_d(\mathcal{N}_d, M_{0d}), \exists \sigma_d \text{ tel que } M_d[\sigma_d > \text{ et } t_j \in \sigma_d \quad (\text{I.4})$$

Un RdP est dit vivant pour un marquage initial M_{0d} si toutes les transitions sont vivantes. Dans un RdP vivant chaque transition peut toujours être franchie quelque soit l'état atteint. Un RdP est dit conforme s'il est sauf et vivant. Le RdP donné par la figure I.8 est conforme, car il est sauf et les transitions t_1 et t_2 sont vivantes.

Le blocage est caractérisé par un marquage pour lequel aucune transition n'est validée. Un RdP est dit sans blocage si aucun marquage atteignable à partir de M_{0d} n'est un blocage [DA10]. Considérons le RdP de la figure I.9, dans cet exemple, nous distinguons deux cycles, un cycle producteur de jetons, qui se déclenche dès le franchissement de la transition t_1 et un cycle consommateur dans le cas du franchissement de la transition t_2 . Le franchissement de la transition t_2 conduit au marquage $M_d = [0 \ 1 \ 0]^t$ qui est un blocage.

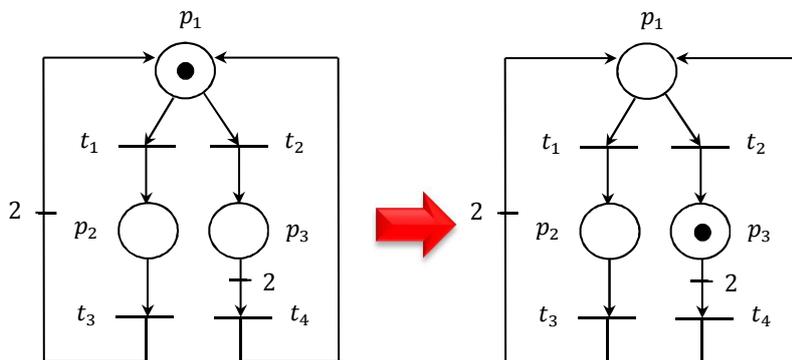


FIGURE I.9 – Exemple d'un RdP avec blocage

Un RdP est dit persistant pour un marquage initial M_{0d} , si pour tout marquage M_d atteignable à partir de M_{0d} et si t_j et $t_{j'}$ sont validées par M_d , alors la séquence $\sigma_d = t_j t_{j'}$ est franchissable à partir de M_d [DA10].

Un RdP est dit structurellement persistant s'il est persistant pour tout marquage initial. Un RdP sans conflit structurel est structurellement persistant [DA10]. Un exemple de RdP structurellement persistant est donné par la figure I.1.

Finalement, certaines propriétés peuvent également être obtenues grâce aux annulateurs à droite et à gauche de la matrice d'incidence. D'une part, si $\exists Z \in \mathbb{N}_{\geq 0}^n$ tel que $Z^t \cdot W = 0_{1 \times q}$ (où $0_{1 \times q}$ est le vecteur nul de q colonnes), Z est appelé P-semi-flot. Dans le cas où $\exists Z \in \mathbb{Z}_{\geq 0}^n$, Z est appelé P-flot.

Si $Z > 0_{n \times 1}$, le RdP est dit conservatif et la loi de conservation est donnée par $Z^t \cdot M_{0d} = Z^t \cdot M_d$. La constante $Z^t \cdot M_{0d}$ s'appelle l'invariant de marquage. Il en découle que si le RdP est conservatif alors il est structurellement borné.

Le RdP de la figure I.10 représente la succession des saisons. Ce réseau possède un seul P-semi-flot qui est $Z = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^t$ et l'invariant de marquage est $m_1 + m_2 + m_3 + m_4 = 1$. Cette propriété indique qu'il y a une et une seule saison à la fois [DA10]. Ce réseau est également sauf et sans blocage.

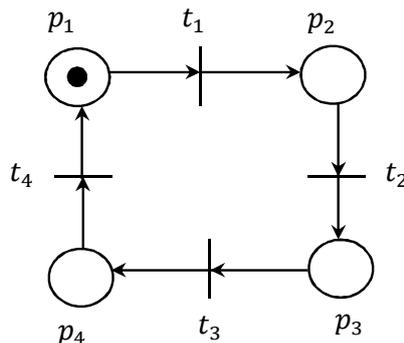


FIGURE I.10 – Exemple d'un RdP conservatif

D'autre part, si $\exists Y \in \mathbb{N}_{\geq 0}^q$ avec $W \cdot Y = 0_{n \times 1}$, Y est appelé T-semi-flot. Dans le cas où $\exists Y \in \mathbb{Z}_{\geq 0}^q$, Y est appelé T-flot.

Si $Y > 0_{q \times 1}$, le RdP est dit consistant. Par conséquent le marquage reste inchangé $M_d = M_{0d} + W \cdot Y = M_{0d}$ [Sil93, JJRS05]. Le RdP donné par la figure I.10 est également consistant, le seul T-semi-flot est $Y = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^t$.

Notons qu'un RdP peut avoir plusieurs P-semi-flots et T-semi-flots. Une combinaison linéaire de T-semi-flots (respectivement de P-semi-flots) est aussi un T-semi-flot (respectivement un P-semi-flot).

Des contraintes temporelles peuvent être ajoutées pour conditionner l'évolution du système et donner des informations sur les dates d'occurrence des événements et les durées de réalisation des actions. Le paragraphe suivant aborde la notion de temporisation pour les RdPD.

I.1.2 Présentation du réseau de Petri discret T-temporisé

Dans la littérature, il existe plusieurs sortes de RdPD temporisés dont les RdPD P-temporisés et T-temporisés pour lesquels la temporisation est associée respectivement aux places et aux transitions. Les deux modèles sont équivalents [DA92, DA01]. Seul le RdPD

T-temporisé est utilisé dans ces travaux de recherche et sera détaillé dans ce paragraphe. L'autre extension sera présentée de façon concise.

Concernant le RdPD P-temporisé, une temporisation est associée à chaque place $p_i \in P$ et par suite à chaque marque déposée dans chacune des places. Les marques restent indisponibles pendant cette durée. Plus précisément, le calcul du degré de validation η_d ne tient compte que des marques disponibles (dont la temporisation est écoulée). Dans ce type de réseau, le franchissement d'une transition validée est instantané.

En ce qui concerne le RdPD T-temporisé, une durée minimale de franchissement, désignée par $D_{min} = (d_{min\ j})_{j=1, \dots, q} \in \mathbb{R}_{\geq 0}^q$, est associée aux transitions telle que $d_{min\ j}$ correspond à la durée minimale de franchissement de la transition t_j . En conséquence, le RdPD T-temporisé est représenté par le triplet $\langle \mathcal{N}_d, D_{min}, M_{0d} \rangle$. Par ailleurs, le marquage, le degré de validation, la séquence de franchissements et le vecteur de franchissements dépendent désormais du temps τ , ils sont respectivement notés $M_d(\tau)$, $\eta_d(M_d(\tau), t_j)$, $\sigma_d(\tau)$ et $\Gamma_d(\tau)$. L'équation fondamentale dépend également du temps.

Chaque transition t_j , $j = 1, \dots, q$, peut être franchie à partir d'un marquage M_d à la date τ ssi elle est validée par le marquage courant et que tous les jetons nécessaires pour le franchissement de cette transition ont été réservés au moins à partir de la date $\tau - d_{min\ j}$, ce qui revient à dire que les jetons ont été bloqués dans les places pendant une durée qui n'est pas inférieure à $d_{min\ j}$ [DA92] (voir figure I.11). Si $d_{min\ j} = 0$, la transition est dite immédiate [DA92].

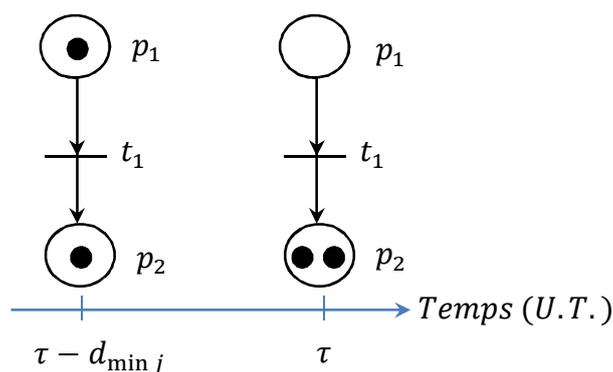


FIGURE I.11 – Franchissement d'une transition discrète temporisée

I.1.3 Domaines d'application du réseau de Petri discret

À son apparition en 1962, le RdP a été considéré comme un outil dédié aux sciences informatiques. Mais progressivement, les domaines d'application se sont élargis et le RdP s'est répandu dans le cadre de la modélisation et de l'analyse de SED dans des secteurs très différents. Cette évolution est mise en évidence dans la littérature non seulement par le nombre de conférences, de cours, de livres, d'outils et de normes qui lui sont consacrés, mais aussi par le nombre d'applications pratiques. Dans ce qui suit nous présenterons quelques domaines d'application du RdPD.

De nombreux travaux utilisent les RdPD pour déterminer l'ordonnancement des activités dans les systèmes manufacturiers flexibles pour atteindre le meilleur rendement et pour profiter pleinement de leur flexibilité [LD94, MYK02, YRCL03a, YRCL03b, KSN07]. À titre d'exemple, [LD94] ont proposé une modélisation par RdPD combinée avec une recherche heuristique. En ce qui concerne la modélisation, le système a été décomposé en plusieurs tâches. Chacune d'entre elles a été modélisée par un sous-RdPD. Finalement, en modélisant les différentes interactions entre les sous-modèles, le modèle global est obtenu. Le RdPD possède l'avantage de représenter toutes les caractéristiques du système en une seule formulation ainsi que ses différents comportements possibles. Le problème d'ordonnancement a été alors reformulé en fonction du RdPD.

Grâce à leur capacité d'analyse des systèmes commandés, les RdPD ont aussi été utilisés dans les réseaux de transport, plus précisément les réseaux urbains. [LC04] ont proposé une modélisation modulaire de la commande des carrefours à feux à l'aide des RdPD. Ensuite, ils ont analysé les caractéristiques du modèle obtenu (P-invariant et blocage) pour vérifier que son fonctionnement est correct, qu'il respecte les règles de sécurité et qu'il ne présente aucun blocage. Dans le même cadre, [DBM⁺14] ont pu optimiser le flux du trafic urbain en utilisant le RdPD couplé à des algorithmes génétiques.

I.1.4 Problème d'explosion combinatoire du nombre d'états

Le graphe d'atteignabilité (ou graphe des marquages) regroupe tous les marquages atteignables à partir d'un marquage initial ainsi que les transitions dont le franchissement entraîne le passage d'un marquage à un autre. Chaque sommet du graphe correspond à un marquage atteignable et chaque arc correspond au franchissement d'une transition validée [DA10]. Soit le RdP représenté par la figure I.10 caractérisé par un marquage initial $M_{0d} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^t$. Ce réseau possède trois marquages atteignables à partir de M_{0d} . Le graphe d'atteignabilité de ce réseau est donné par la figure I.12.

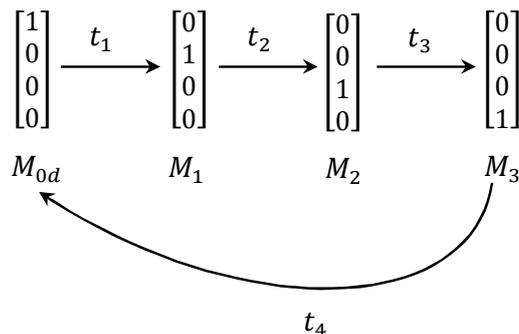


FIGURE I.12 – Graphe d’atteignabilité du RdP de la figure I.10

Le RdPD est dédié essentiellement à la modélisation, à l’analyse et à la vérification du comportement des SED. La construction du graphe d’atteignabilité du système étudié peut s’avérer complexe car le nombre d’états atteignables du RdPD est souvent très grand et augmente exponentiellement en fonction du nombre de jetons compris dans le marquage initial. Le graphe d’atteignabilité devient alors extrêmement volumineux même pour des réseaux de petite taille (à savoir un nombre restreint de places et de transitions). L’effort de calcul requis pour explorer le graphe d’atteignabilité et la difficulté à vérifier certaines propriétés empêchent l’utilisation des techniques d’analyse par énumération [Sil13]. Ce problème est connu sous le nom d’explosion combinatoire du nombre d’états, inhérent à la plupart des DES modélisés. Il constitue une des principales limitations à l’utilisation du RdPD dans l’analyse des SED et soulève une problématique fondamentale : quelle technique pourra être envisagée afin de surmonter cet inconvénient majeur ?

I.2 Modélisation par réseaux de Petri continus

De nombreuses études au cours des trente dernières années ont été menées afin de pallier le problème d’explosion combinatoire du nombre d’états inhérent aux SED. L’une des méthodes les plus efficaces consiste à fluidifier le RdPD.

I.2.1 Principe de la fluidification

La fluidification est une technique classique de relaxation des contraintes qui imposent la circulation d’un nombre réel de marques dans le réseau et qui conduit à des solutions approchées pour l’analyse qualitative et quantitative du RdPD [SJM11]. Elle a été proposée en 1987 dans la conférence “European Workshop on Petri Nets Applications and Theory” par David et Alla dans le but de fluidifier les réseaux de files d’attente en recherche opérationnelle. Dans cet article, la fluidification est réalisée sur le réseau. Dans la même conférence, Silva and Colom ont également proposé la fluidification de l’équation

fondamentale d'état. Le principe du second papier réside dans la relaxation du problème d'optimisation linéaire en nombres entiers ("Integer Programming Problem"(IPP)) afin d'obtenir un problème d'optimisation linéaire ("Linear Programming Problem"(LPP)). Cette transformation permet d'analyser de manière efficace la structure du RdPD [DA10].

Deux types de fluidifications peuvent avoir lieu : la fluidification totale et la fluidification partielle. D'une part, la fluidification totale d'un RdPD autonome signifie que les contraintes de toutes les transitions sont relaxées. Le marquage et le franchissement des transitions continues peuvent être respectivement perçus comme une quantité de fluide stocké et l'écoulement de ce fluide d'une place vers d'autres places (voir figure I.13). Par conséquent, le vecteur de franchissements ainsi que le marquage ne correspondent plus à des valeurs entières, mais à des valeurs réelles positives. D'autre part, la fluidification partielle donne naissance au RdPH. Cette extension sera abordée ultérieurement dans la section I.3.

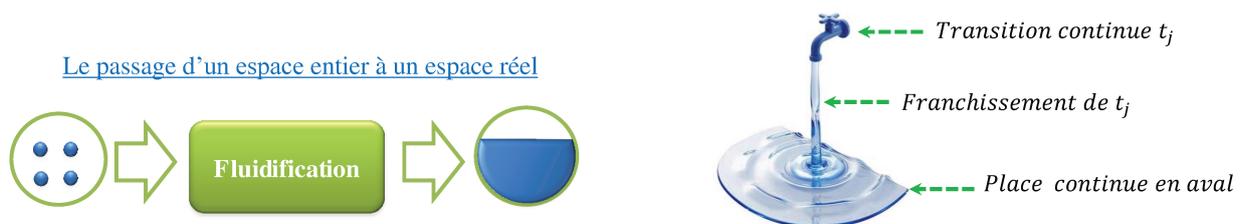


FIGURE I.13 – Schéma explicatif de la fluidification

Le RdPC permet soit d'approcher le comportement d'un RdPD en évitant l'explosion combinatoire du nombre d'états, soit de modéliser un système dynamique continu qui ne peut pas être modélisé par un RdPD, par exemple l'écoulement d'un liquide [DA92]. Les différents domaines d'application du RdPC seront abordés ultérieurement dans ce chapitre. Commençons par présenter cette extension.

I.2.2 Présentation du réseau de Petri continu autonome

I.2.2.1 Définition du réseau de Petri continu autonome

Formellement, le RdPC est défini par $\langle \mathcal{N}_c, M_{0c} \rangle$. De manière identique au RdPD, \mathcal{N}_c dénote la structure statique du réseau $\langle P, T, W_{pr}, W_{po} \rangle$ avec un ensemble de places P , un ensemble de transitions T et les matrices d'incidence avant et arrière. M_{0c} représente le

marquage initial du réseau. En revanche, le poids des arcs ainsi que le nombre de marques ne sont plus des entiers, ils sont désormais des réels non-négatifs : $W \in \mathbb{R}_{\geq 0}^{n \times q}$ et $M_{0c} \in \mathbb{R}_{\geq 0}^n$. Le vecteur de marquages d'un RdPC est représenté par $M_c = \begin{bmatrix} m_1 & \dots & m_i & \dots & m_n \end{bmatrix}^t \in \mathbb{R}_{\geq 0}^n$ tel que m_i correspond au marquage de la place p_i .

Graphiquement, les places et les transitions d'un RdPC sont respectivement symbolisées par des cercles doubles et des rectangles. Puisque le nombre de marques est un réel, le marquage est indiqué à l'aide de nombres placés à l'intérieur de chaque place. La figure I.14 présente un exemple simple d'un RdPC généralisé.

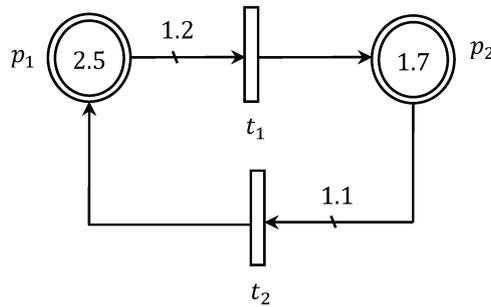


FIGURE I.14 – Exemple d'un RdPC généralisé

Ce réseaux de Petri possède $n = 2$ places continues et $q = 2$ transitions continues. Les matrices d'incidence sont définies par :

$$W_{pr} = \begin{bmatrix} 1.2 & 0 \\ 0 & 1.1 \end{bmatrix}; W_{po} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}; W = \begin{bmatrix} -1.2 & 1 \\ 1 & -1.1 \end{bmatrix}$$

Le marquage initial est donné par : $M_{0c} = \begin{bmatrix} 2.5 & 1.7 \end{bmatrix}^t$.

I.2.2.2 Principe d'évolution du réseau de Petri continu autonome

De manière identique au RdPD, l'évolution du RdPC résulte du franchissement des transitions continues validées. Contrairement au RdPD, une transition continue t_j est dite validée par un marquage M_c ssi $\forall p_i \in {}^\circ t_j, m_i > 0$. En d'autres termes, ssi le marquage des places en amont est strictement positif. Le degré de validation d'une transition continue t_j est désigné par $\eta_c(t_j, M_c)$ et est défini par l'équation (I.5) :

$$\eta_c(t_j, M_c) = \min_{p_i \in {}^\circ t_j} \left\{ \frac{m_i}{W_{pr}(i, j)} \right\} \quad (\text{I.5})$$

Il ne se limite plus à un nombre entier. Ainsi, le franchissement de t_j avec une quantité réelle de marques α_c telle que $0 < \alpha_c \leq \eta_c(t_j, M_c)$ engendre le déplacement d'un nombre réel de marques dans le réseau. Plus précisément, $\alpha_c \cdot W_{pr}(i, j)$ marques sont retirées des places p_i en amont de t_j et simultanément $\alpha_c \cdot W_{po}(i, j)$ marques sont ajoutées à chacune des places p_i en aval. La variation du marquage continu est donnée par (I.6) :

$$\Delta M_c = \alpha_c \cdot W(:, j) \quad (\text{I.6})$$

L'espace d'atteignabilité d'un RdPC est désigné par $R_c(\mathcal{N}_c, M_{0c})$, Soit $\sigma_c \in T^*$ une séquence finie de franchissements permettant d'atteindre un marquage M_c à partir de M_{0c} où $M_c \in R_c(\mathcal{N}_c, M_{0c})$ et soit $\Gamma_c = (\gamma_{cj})_{j=1, \dots, q} \in \mathbb{R}_{\geq 0}^q$ le vecteur de franchissements qui caractérise cette séquence, où $\gamma_{cj} \in \mathbb{R}$ désigne la quantité cumulée de franchissements de la transition t_j dans la séquence σ_c . L'équation fondamentale de variation du marquage est donnée par (I.7) :

$$M_c = M_{0c} + W \cdot \Gamma_c \quad (\text{I.7})$$

Puisque la notion de temps n'intervient pas, le RdPC ainsi défini ne donne aucune information sur l'évolution temporelle du marquage. Le RdPC temporisé est défini afin de prendre en compte explicitement le temps.

I.2.3 Présentation du réseau de Petri continu temporisé

Le RdPC temporisé est un RdPC auquel des spécifications temporelles sont ajoutées. Ces spécifications sont définies par le vecteur λ qui contient la vitesse maximale de franchissement de chaque transition : $\lambda = (\lambda_j)_{j=1, \dots, q} \in \mathbb{R}_{> 0}^q$. Par conséquent, le RdPC temporisé est formellement représenté par le triplet $\langle \mathcal{N}_c, M_{0c}, \lambda \rangle$. De plus, le marquage continu ainsi que l'équation fondamentale de variation du marquage (I.7) dépendent désormais de τ et le vecteur de franchissements $\Gamma_c(\tau)$ caractérise les franchissements dans l'intervalle de temps $[0, \tau]$.

Le franchissement des transitions s'effectue selon un flux continu et les jetons s'écoulent de manière continue, comme un fluide, d'une place d'entrée vers une place de sortie. Le flux de franchissement est noté $F_c(\tau) = \dot{\Gamma}_c(\tau) = (f_j(\tau))_{j=1, \dots, q} \in \mathbb{R}_{\geq 0}^q$. Il mesure la quantité maximale de jetons qui peuvent passer à travers chacune des transitions par unité de temps (U.T.) en régime libre.

L'espace d'atteignabilité d'un RdPC temporisé est désigné par $R_c(\mathcal{N}_c, M_{0c}, \lambda)$. Si $M_c(\tau) \in R_c(\mathcal{N}_c, M_{0c}, \lambda)$, la variation du marquage en fonction du temps est définie par la

dérivée de l'équation (I.7) par rapport au temps. L'équation d'état est définie par (I.8) :

$$\dot{M}_c(\tau) = W \cdot \dot{\Gamma}_c(\tau) = W \cdot F_c(\tau) \quad (\text{I.8})$$

Selon la manière dont le flux de franchissement est défini, plusieurs modèles de RdPC apparaissent. Les plus souvent utilisées dans la littérature sont la sémantique serveurs finis (ou vitesses constantes) et la sémantique serveurs infinis (ou vitesses variables) [DA92]. Dans le cas de la sémantique serveurs finis, la notion de transition validée change. En effet, une transition t_j est dite “fortement validée” si le marquage de toutes les places en amont est strictement positif. Ainsi, le flux de franchissement de cette transition est constant et correspond à sa vitesse maximale de franchissement λ_j , ce qui revient à dire que tous les serveurs travaillent à vitesse maximale [DA92,DA01]. Dans le cas inverse, la transition est dite “faiblement validée” à condition que les places en amont soient alimentées (c-à-d s'il existe au moins une transition en amont de ces places telle que sa vitesse est strictement positive [DA01]). Dans ces conditions, le flux de franchissement de t_j est formellement défini par l'équation (I.9) comme le minimum entre la vitesse maximale de franchissement et le flux total à l'entrée des places en amont de t_j [DA01].

$$f_j(\tau) = \begin{cases} \lambda_j, & \text{si } \forall p_i \in {}^\circ t_j, m_i > 0 \\ \min \left\{ \min_{p_i \in {}^\circ t_j | m_i = 0} \left\{ \sum_{t_{j'} \in {}^\circ p_i} \frac{f_{j'} \cdot W_{po}(i, j')}{W_{pr}(i, j')} \right\}, \lambda_j \right\}, & \text{sinon} \end{cases} \quad (\text{I.9})$$

Le flux de chaque transition est donc constant par morceaux. En fait, il est constant et correspond à la vitesse maximale de franchissement tant qu'aucun événement ne se produit. Le seul événement qui peut modifier le comportement du réseau est l'assèchement complet de la place en amont de la transition [DA10]. Entre deux événements, le RdPC est dit dans un état comportemental invariant appelé IB-state (“Invariant Behavior state”) [DA10]. Dans un IB-state, le marquage varie de manière linéaire. En conséquence, l'évolution temporelle du marquage est linéaire par morceaux.

Concernant le sémantique serveurs infinis, le flux de franchissement de chaque transition t_j est une fonction continue du temps, défini comme le produit du degré de validation par la vitesse maximale de franchissement (I.31) [DA92,SR04] :

$$f_j(\tau) = \lambda_j \cdot \eta_c(t_j, M_c(\tau)) = \lambda_j \cdot \min_{p_i \in {}^\circ t_j} \left\{ \frac{m_i(\tau)}{W_{pr}(i, j)} \right\} \quad (\text{I.10})$$

Contrairement à la sémantique serveurs finis, le flux de chaque transition t_j n'est pas

constant, il varie continûment en fonction du marquage des places en amont et de λ_j .

Dans [MRS06], les auteurs ont expliqué les différentes situations pour lesquelles les approximations données par les deux sémantiques précédemment définies convergent ou divergent. Il existe des RdP pour lesquels la sémantique serveurs finis est la plus adéquate pour approcher le système réel. Mais dans [MRS06], il a été démontré que pour une catégorie importante de RdP, la sémantique serveurs infinis fournit une meilleure approximation du RdPD modélisant le système que la sémantique serveurs finis [MRTRS08]. À cet égard, dans la suite de cette thèse, nous utilisons les RdPC à vitesses variables.

Le RdPC à vitesses variables est caractérisé par des dynamiques non-linéaires dues à l'opérateur "min" qui apparaît dans l'expression du flux. Des commutations se produisent pour chaque synchronisation. Afin de surmonter cette difficulté, deux solutions ont été proposées : l'utilisation de l'algèbre (max, plus) et l'introduction de la notion de régions. Dans ces travaux, nous détaillerons la deuxième solution que nous avons utilisée dans nos travaux de recherche.

L'espace d'atteignabilité $R_c(\mathcal{N}_c, M_{0c}, \lambda)$ peut être partitionné en une ou plusieurs régions : $R_c(\mathcal{N}_c, M_{0c}, \lambda) = R_1 \cup \dots \cup R_k \cup \dots \cup R_K$. Ces régions sont disjointes sauf sur les frontières qu'elles partagent [AOJMS11]. Le nombre de régions dépend non seulement du nombre de synchronisations dans le réseau, mais également du nombre de places qui constituent chaque synchronisation. Ainsi, le nombre total satisfait (I.11) :

$$K \leq \prod_{j=1}^q |{}^\circ t_j| \quad (\text{I.11})$$

Chaque région, R_ζ , $\zeta = 1, \dots, K$, correspond à une configuration. Afin de définir explicitement ces configurations, les places critiques sont définies par le sous-ensemble de places qui limitent le flux des transitions. À l'intérieur de chaque région, une seule place critique limite le flux de chaque transition. Si $p_i \in {}^\circ t_j$ est la place critique de t_j , elle doit satisfaire l'égalité (I.12) [LL11]. Finalement, une configuration regroupe les places critiques des différentes transitions :

$$\frac{m_i(\tau)}{W_{pr}(i, j)} = \eta_c(t_j, M_c(\tau)) \quad (\text{I.12})$$

Sur les frontières des régions, une même transition peut avoir plusieurs places critiques équivalentes [AOJMS11]. Pour terminer, chaque région, $\forall \zeta = 1, \dots, K$ est associée à une matrice de contraintes, désignée par A_ζ , telle que $A_\zeta \in \mathbb{R}_{\geq 0}^{q \times n}$ [MRTRS08]. Elle est

formellement définie par (I.13) :

$$A_\zeta(j, i) = \begin{cases} \frac{1}{W_{pr}(i, j)}, & \text{si } p_i \text{ est la place critique de } t_j \\ 0, & \text{sinon} \end{cases} \quad (\text{I.13})$$

Par conséquent, chaque ligne, $j = 1, \dots, q$, de la matrice de contraintes n'a qu'un seul élément non-nul à la $i^{\text{ème}}$ position. Par soucis de clarté, notons respectivement par $R(\tau)$ et $A(\tau)$ la région du marquage M_c à la date τ et la matrice de contraintes correspondante. Le degré de validation de toutes les transitions est explicitement exprimé par le produit $A(\tau) \cdot M_c(\tau)$. Soit Λ une matrice diagonale dont les éléments sont ceux de λ ($\Lambda = \text{diag}(\lambda)$), le flux de franchissement peut ainsi s'écrire sous la forme matricielle donnée par (I.14) :

$$F_c(\tau) = \Lambda \cdot A(\tau) \cdot M_c(\tau) \quad (\text{I.14})$$

L'équation fondamentale (I.8) est désormais exprimée par (I.15) :

$$\dot{M}_c(\tau) = W \cdot \Lambda \cdot A(\tau) \cdot M_c(\tau) \quad (\text{I.15})$$

Ainsi, dans chaque région (ou configuration), un seul système linéaire régit l'évolution du marquage. En conséquence, le RdPC à vitesses variables est linéaire par morceaux [MRS06].

Un exemple de RdPC continu ainsi que ses régions sont respectivement représentés sur les figures I.15 et I.16 :

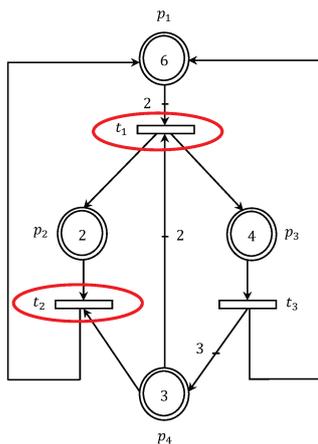


FIGURE I.15 – Exemple d'un réseau continu avec plusieurs régions

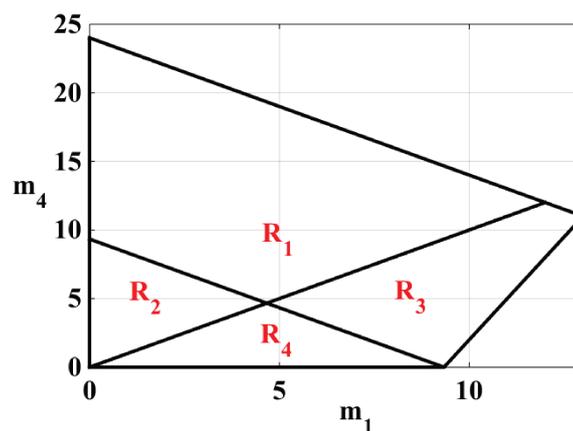


FIGURE I.16 – Projection des régions dans le plan (m_1, m_4)

Comme illustré sur la figure I.16, cet exemple possède $K = 4$ régions définies par (I.16) :

$$\begin{aligned}
 R_1 &= \{M_c | m_1 \leq m_4, m_2 \leq m_4\} \\
 R_2 &= \{M_c | m_1 \leq m_4, m_4 \leq m_2\} \\
 R_3 &= \{M_c | m_4 \leq m_1, m_2 \leq m_4\} \\
 R_4 &= \{M_c | m_4 \leq m_1, m_4 \leq m_2\}
 \end{aligned} \tag{I.16}$$

La matrice de contraintes de chacune des régions est :

$$A_1 = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}; A_2 = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}; A_3 = \begin{bmatrix} 0 & 0 & 0 & 0.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}; A_4 = \begin{bmatrix} 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

La résolution numérique des équations différentielles nécessite d'échantillonner le temps. De plus, dans [Gho12], les auteurs insistent sur la discrétisation du temps afin de garder le comportement événementiel dans le cas de la sémantique serveurs infinis. Le temps continu τ est dorénavant échantillonné grâce à une horloge externe de période d'échantillonnage θ telle que $\tau_k = \theta \cdot k$ où k définit l'indice d'échantillonnage et τ_k correspond au temps échantillonné (ou discret). Dans ce qui suit, le RdPC à temps échantillonné sera désigné par θ -RdPC = $\langle \mathcal{N}_c, M_0, \lambda, \theta \rangle$.

Toutefois, suite à cet échantillonnage, des questions cruciales se posent : est-ce que le modèle en temps échantillonné préserve toutes les informations du modèle en temps continu ? Est-ce que les deux espaces d'atteignabilité sont équivalents ? Afin de répondre à ces questions, une étude complète a été menée dans [MRSS06]. En vue de préserver la positivité du marquage et dans le but d'éviter les solutions non-admissibles, une condition sur la période d'échantillonnage a été fournie : La période d'échantillonnage doit satisfaire l'inégalité (I.17) [MRSS06] :

$$\forall p_i \in P, \sum_{t_j \in p_i^\circ} \lambda_j \cdot \theta < 1 \tag{I.17}$$

Dans ce qui suit, notons respectivement par R_k et A_k la région de M_c et sa matrice de contraintes à la date τ_k . À chaque date τ_k , le système d'équations (I.18) régit le comportement du θ -RdPC dans chaque région R_k :

$$\begin{cases} M_c(\tau_{k+1}) = M_c(\tau_k) + \theta \cdot W \cdot F_c(\tau_k) \\ F_c(\tau_k) = \Lambda \cdot A_k \cdot M_c(\tau_k) \end{cases} \tag{I.18}$$

En conséquence, $M_c(\tau_{k+1})$ peut s'exprimer par (I.19) :

$$M_c(\tau_{k+1}) = B_k \cdot M_c(\tau_k) \quad (\text{I.19})$$

où la matrice B_k est donnée par (I.20) :

$$B_k = I_n + \theta \cdot W \cdot \Lambda \cdot A_k \quad (\text{I.20})$$

avec I_n la matrice identité de dimensions $n \times n$.

Une propriété importante résulte de (I.17). Soit $R_c(\mathcal{N}_c, M_{0c}, \lambda, \theta)$ l'espace d'atteignabilité du θ -RdPC, étant donné que $M_{0c} \geq 0_{n \times 1}$, alors tous les marquages atteignables à partir de M_{0c} sont aussi non-négatifs [MRSS06]. En d'autres termes, $\forall M_c \in R_c(\mathcal{N}_c, M_{0c}, \lambda, \theta) \Rightarrow M_c(\tau_k) \geq 0_{n \times 1}$.

Une place déjà marquée ne peut en aucun cas être vidée après un nombre fini de pas [MRSS06]. Plus généralement, les remarques suivantes peuvent être formulées concernant les espaces d'atteignabilité des RdPC et θ -RdPC :

- Un marquage est dit atteignable dans un θ -RdPC ssi il est atteignable dans le RdPC autonome correspondant, à travers une séquence qui ne vide aucune place déjà marquée [MRSS06, MGR⁺08].
- Tant que θ vérifie (I.17), les deux espaces d'atteignabilité $R_c(\mathcal{N}_c, M_{0c}, \lambda, \theta)$ et $R_c(\mathcal{N}_c, M_{0c}, \lambda)$ sont confondus [MRSS06, MGR⁺08].
- Cependant, $R_c(\mathcal{N}_c, M_{0c}, \lambda, \theta)$ ne coïncide pas parfaitement avec $R_c(\mathcal{N}_c, M_{0c})$. En effet, si $M_c \in R_c(\mathcal{N}_c, M_{0c})$ tel que $M_c \geq 0_{n \times 1}$, alors ce marquage est lim-atteignable dans le θ -RdPC correspondant [MGR⁺08]. Le terme lim-atteignable fait référence à la notion d'atteignabilité en temps infini (ou en nombre infini de pas).

Les autres théorèmes présentés dans [MRSS06, MGR⁺08] concernent le θ -RdPC forcé. Ils seront détaillés dans le chapitre suivant.

Avant de passer à la présentation d'autres extensions du RdP, nous présentons quelques domaines d'application du RdPC.

I.2.4 Domaines d'application du réseau de Petri continu

Les RdPC ont été développés pour permettre la modélisation de classes importantes de SED qui souffrent du problème de l'explosion combinatoire du nombre d'états inhérent à la modélisation par les RdPD. Les RdPC permettent également de proposer des techniques efficaces d'analyse, d'évaluation de performance et de commande. En conséquence, les RdPC ont été largement utilisés dans des applications réelles dans différents domaines, spécialement le domaine manufacturier et de transport. En particulier, [AD88] ont proposé deux types de modélisations par les RdPC à vitesses constantes des systèmes de production typiques composés par des stations et des convoyeurs continus.

[JB10] ont étudié le comportement des réseaux de transport et ils ont proposé une modélisation macroscopique de ce type de réseaux par les RdPC. De plus, ils ont utilisé ce modèle, d'une part, pour résoudre le problème de variation rapide des conditions de circulation, et d'autre part pour limiter le retard total des voitures dans le réseau. Dans ce même contexte, [TLTM05] se sont focalisés sur les RdPC à vitesses variables et les RdPD temporisés afin de proposer deux approches macroscopique et microscopique du flux de circulation respectivement à l'aide de RdPC à vitesses variables et de RdPD temporisés. L'approche macroscopique considère les segments routiers larges dont les caractéristiques globales (la densité, la vitesse et le débit) ont été exprimées en fonction des paramètres du RdPC. L'approche microscopique utilise une discrétisation (diviser chaque segment en sous-segments) dans le but de décrire individuellement les caractéristiques de chaque véhicule sur la route. Ces deux modèles ont été combinés afin d'analyser, d'évaluer la performance et de contrôler les réseaux de trafic urbains et interurbains.

D'autres domaines d'application ont été également intéressés par ce type de réseaux, notamment le domaine de la logistique. Les auteurs de [SBB⁺11] ont été motivés par la modélisation, l'analyse, l'évaluation de performance et le pilotage des flux de la chaîne logistique. Ils ont proposé une démarche modulaire et générique afin d'établir un modèle de simulation macroscopique continu et dynamique de la chaîne logistique par les RdPC à vitesses variables.

I.2.5 Limitation du réseau de Petri continu

Le RdPC est présenté comme une solution pour éviter le problème de l'explosion combinatoire du nombre d'états. En revanche, d'après [SR02, Sil13, San15], tous les SED ne peuvent pas être fluidifiés de manière satisfaisante. Certains travaux de recherche tels que [SR05] ont été effectués afin d'étudier la conservation des propriétés importantes du

RdP à travers la fluidification. Nous pouvons espérer que certaines propriétés du RdPD peuvent être conservées, par exemple, si le RdPC est borné, alors le RdPD correspondant sera également borné [SR05]. Cependant, cela ne vaut pas pour d'autres propriétés quantitatives qui ne peuvent être ni observées ni analysées une fois la relaxation effectuée telles que la vivacité et le blocage, car le RdPC et le RdPD n'ont pas toujours un comportement similaire. Ils peuvent avoir des comportements totalement différents. En effet, les transitions dans le RdPC sont franchies avec une quantité réelle de marques. Par conséquent, certains marquages qui ne sont pas atteignable dans le RdPD deviennent atteignables dans le RdPC. En d'autres terme, $R_d(\mathcal{N}_d, M_{0d}) \neq R_c(\mathcal{N}_c, M_{0c})$. Par conséquent, les propriétés liées à la vivacité et au blocage ne peuvent pas être conservées via la fluidification [SR02, SR05]. De plus, le RdPC ne permet pas de représenter ni des conditions logiques ni des actions discrètes (par exemple l'ouverture ou la fermeture d'une vanne, le partage de ressources) [GA07]. Les variables qui correspondent à ces actions doivent pourtant garder leur caractère discret. Pour cette raison, le réseau de Petri Hybride (RdPH) a été défini dans [DA01].

I.3 Modélisation par réseaux de Petri hybrides

Le RdPH représente un formalisme puissant qui permet de modéliser les systèmes dynamiques hybrides et de décrire les changements logiques qui peuvent survenir. Le RdPH est obtenu grâce à une relaxation partielle du RdP afin de tenir compte des événements discrets et des évolutions continues. De ce fait, le RdPH résulte d'une interaction entre une partie continue et une partie discrète dans un même modèle [BGM00, DA01, JMV11].

I.3.1 Présentation du réseau de Petri Hybride autonome

I.3.1.1 Définition du réseau de Petri hybride autonome

Un RdPH est un couple $\langle \mathcal{N}_h, M_{0h} \rangle$ où \mathcal{N}_h est la structure du réseau et M_{0h} est le marquage initial. \mathcal{N}_h est représentée par le quadruplet $\langle P, T, W_{pr}, W_{po} \rangle$ où $P = \{p_i\}_{i=1, \dots, n}$ et $T = \{t_j\}_{j=1, \dots, q}$ sont les deux ensembles finis non-vides composés respectivement de n places et de q transitions. Suivant [DA01], P et T se décomposent explicitement en deux sous-ensembles continu et discret disjoints : $P = P_d \cup P_c$ avec $P_d \cap P_c = \emptyset$ et $T = T_d \cup T_c$ avec $T_d \cap T_c = \emptyset$ tels que $n_d = |P_d|$, $n_c = |P_c|$, $q_d = |T_d|$ et $q_c = |T_c|$ sachant que $n = n_d + n_c$ et $q = q_d + q_c$. P_d et P_c contiennent respectivement les places discrètes et continues, et T_d et T_c regroupent respectivement les transitions discrètes et continues.

W_{pr} et W_{po} désignent la structure statique du réseau : $\forall (p_i, t_j) \in P \times T$, $W_{pr}(i, j)$ et $W_{po}(i, j)$ représentent respectivement le poids de l'arc de p_i vers t_j et de t_j vers p_i . Si

$p_i \in P_d$, $W_{pr}(i, j) \in \mathbb{N}$ et $W_{po}(i, j) \in \mathbb{N}$, si $p_i \in P_c$, $W_{pr}(i, j) \in \mathbb{R}_{\geq 0}$ et $W_{po}(i, j) \in \mathbb{R}_{\geq 0}$. Tout au long du mémoire et sous réserve de changer l'ordre de numérotation des places et transitions, la représentation (I.21) est adoptée afin d'explicitier les interactions qui existent entre la partie continue et la partie discrète du RdPH :

$$W_{pr} = \begin{bmatrix} W_{pr}^{dd} & W_{pr}^{dc} \\ W_{pr}^{cd} & W_{pr}^{cc} \end{bmatrix}; W_{po} = \begin{bmatrix} W_{po}^{dd} & W_{po}^{dc} \\ W_{po}^{cd} & W_{po}^{cc} \end{bmatrix} \quad (\text{I.21})$$

Par conséquent, la matrice d'incidence W est désormais représentée par (I.22) :

$$W = \begin{bmatrix} W^{dd} & W^{dc} \\ W^{cd} & W^{cc} \end{bmatrix} \quad (\text{I.22})$$

de telle manière que les sous-matrices de W_{pr} , W_{po} et W soient de dimensions adéquates. Les sous-matrices W^{dd} et W^{cc} représentent respectivement la structure des parties discrètes et continues, alors que W^{dc} et W^{cd} regroupent respectivement les liens des éléments discrets vers les éléments continus et ceux des éléments continus vers les éléments discrets. Le marquage courant du RdPH est noté $M_h = \begin{bmatrix} M_d^t & M_c^t \end{bmatrix}^t$ où $M_d \in \mathbb{N}_{\geq 0}^{n_d}$ et $M_c \in \mathbb{R}_{\geq 0}^{n_c}$ désignent respectivement le marquage courant des places discrètes et continues. Le marquage initial M_{0h} est désormais défini par $M_{0h} = \begin{bmatrix} M_{0d}^t & M_{0c}^t \end{bmatrix}^t$.

Dans un RdPH, les éléments des deux parties continue et discrète sont inter-connectés. Comme dans les RdPC et les RdPD, chaque élément $v \in P \cup T$ possède des éléments d'entrée et de sortie. D'une manière générale, chaque transition (discrète ou continue) peut avoir en amont ou en aval, sans restriction, des places continues ou discrètes sous réserve de respecter la condition suivante : $\forall (p_i, t_j) \in P_d \times T_c$, $W_{pr}^{dc}(i, j) = W_{po}^{dc}(i, j)$ ce qui implique $W^{dc}(i, j) = 0$. Cette propriété fondamentale du RdPH assure que le franchissement d'une transition continue ne modifie pas le marquage d'une place discrète et maintient un marquage entier pour les places discrètes [DA01]. Par ailleurs, en fonction des liens entre les places continues et les transitions discrètes, il existe trois types de RdPH à savoir les RdPH Élémentaires (RdPHE), les RdPH D-Élémentaires (RdPHDE) et les RdPH non-Élémentaires (RdPHnE) [GA07].

Par définition, un RdPH est dit Élémentaire si toute transition discrète qui possède en amont une place continue, possède la même place continue en aval avec le même poids d'arc. Autrement dit, $W_{pr}^{cd} = W_{po}^{cd}$ [DA01]. Par conséquent, $W^{cd} = 0_{n_c \times q_d}$, ce qui revient à affirmer que le franchissement des transitions discrètes peut influencer le comportement de la partie continue sans modifier le marquage, il n'y a pas de conversion du marquage

continu en marquage discret [DA01]. La matrice d'incidence d'un RdPHE est notée selon (I.23) [DA01, GA07] :

$$W = \begin{bmatrix} W^{dd} & 0_{n_d \times q_c} \\ 0_{n_c \times q_d} & W^{cc} \end{bmatrix} \quad (I.23)$$

Le RdPHDE est une sous-classe du RdPHE qui a été définie dans [LHS05] comme une combinaison d'un RdPD temporisé et d'un RdPC à vitesses constantes afin de modéliser les systèmes dynamiques hybrides à flux continu commandés par des transitions discrètes temporisées. Dans ce type de réseau, il n'y a pas d'interaction directe entre les places continues et les transitions discrètes. Plus précisément, $W_{pr}^{cd} = W_{po}^{cd} = 0_{n_c \times q_d}$ ce qui veut dire que seule la partie discrète influence le comportement de la partie continue [LHS05]. La figure I.17 montre un RdPHE et un RdHDE :

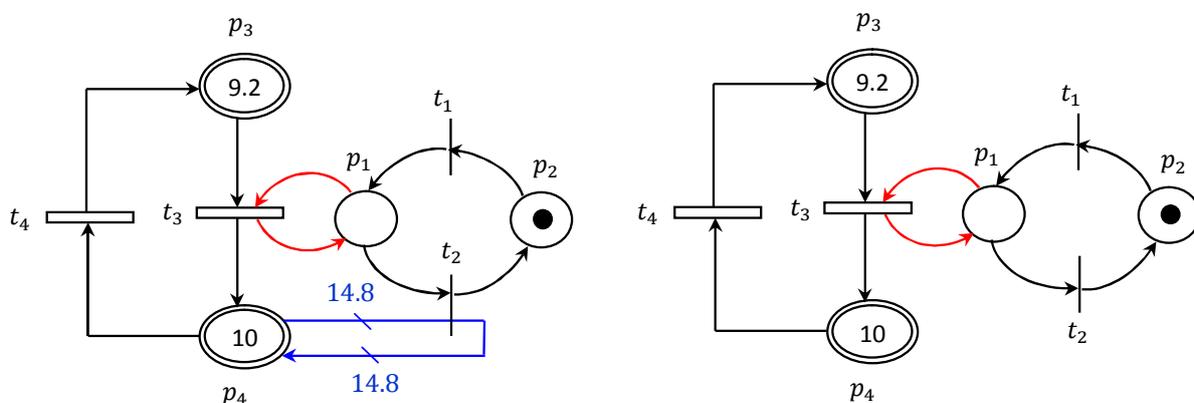


FIGURE I.17 – Exemple d'un RdPHE (à gauche) et d'un RdPHDE (à droite) [DA01]

Les matrices W_{pr} et W_{po} du RdPHE sont données par :

$$W_{pr} = \begin{matrix} & T_d & T_c \\ \begin{matrix} P_d \\ P_c \end{matrix} & \left[\begin{array}{cc|cc} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 14.8 & 0 & 1 \end{array} \right] & ; W_{po} = \begin{matrix} & T_d & T_c \\ \begin{matrix} P_d \\ P_c \end{matrix} & \left[\begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \\ 0 & 14.8 & 1 & 0 \end{array} \right] \end{matrix}$$

Tandis que les matrices W_{pr} et W_{po} du RdPHDE sont données par :

$$W_{pr} = \begin{matrix} & T_d & T_c \\ P_d & \left[\begin{array}{cc|cc} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{array} \right] \\ P_c & \left[\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right] \end{matrix} ; W_{po} = \begin{matrix} & T_d & T_c \\ P_d & \left[\begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{array} \right] \\ P_c & \left[\begin{array}{cc|cc} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{array} \right] \end{matrix}$$

Même si les sous-matrices W_{pr}^{cd} et W_{po}^{cd} des RdPHE et RdPHDE ne sont pas identiques, la matrice d'incidence de ces deux réseaux est la même, elle est donnée par :

$$W = \begin{matrix} & T_d & T_c \\ P_d & \left[\begin{array}{cc|cc} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \end{array} \right] \\ P_c & \left[\begin{array}{cc|cc} 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{array} \right] \end{matrix}$$

Dans le cas de RdPHnE, il existe une interaction directe entre les transitions discrètes et les places continues. Cette interaction se traduit par des valeurs non-nulles de la sous-matrice W^{cd} . Le franchissement d'une transition discrète modifie le marquage des places continues associées. La figure I.18 présente un exemple de RdPHnE :

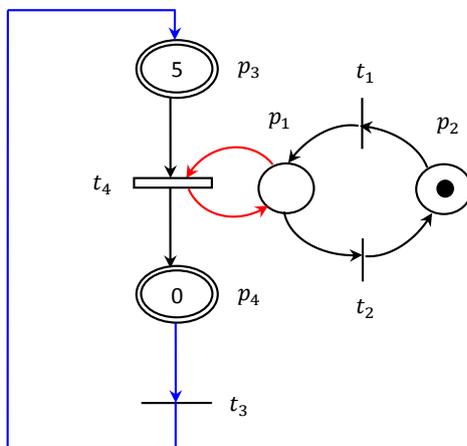


FIGURE I.18 – Exemple d'un RdPHnE

La matrice d'incidence de ce réseau est donnée par :

$$W = \begin{array}{c} P_d \\ P_c \end{array} \begin{array}{cc|c} T_d & T_c & \\ \hline \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \end{bmatrix} & \end{array}$$

Ce mémoire concerne essentiellement les RdPHE les RdPHDE.

I.3.1.2 Principe d'évolution du Réseau de Petri hybride élémentaire autonome

L'évolution du RdPHE résulte du franchissement des transitions continues et discrètes validées. Soit $\eta_d(t_j, M_h)$ le degré de validation d'une transition discrète $t_j \in T_d$. Ce dernier est exprimé par l'équation (I.24) :

$$\eta_d(t_j, M_h) = \min_{p_i \in {}^\circ t_j} \left\{ \left\lfloor \frac{m_i}{W_{pr}(i, j)} \right\rfloor \right\} \quad (\text{I.24})$$

Une transition $t_j \in T_d$ est dite validée par le marquage courant M_h ssi $\eta_d(t_j, M_h) > 0$. Étant donné que $\eta_d(t_j, M_h)$ mesure la quantité maximale de franchissement d'une transition discrète validée t_j , cette dernière ne peut être franchie qu'avec une quantité entière de marques $\alpha_d \in \mathbb{N}$ telle que $0 < \alpha_d \leq \eta_d(t_j, M_h)$.

Le franchissement de cette transition discrète modifie uniquement la partie discrète M_d du marquage hybride M_h dont la variation ΔM_h est exprimée par (I.25) :

$$\Delta M_h = \alpha_d \cdot \begin{bmatrix} W^{dd}(:, j) \\ 0_{n_c \times 1} \end{bmatrix} \quad (\text{I.25})$$

Contrairement aux transitions discrètes, une transition continue $t_j \in T_c$ est validée par le marquage M_h ssi $\forall p_i \in {}^\circ t_j, m_i > 0$. Son degré de validation, noté $\eta_c(t_j, M_h)$, n'est pas limité à un nombre entier, il est défini selon l'équation (I.26) :

$$\eta_c(t_j, M_h) = \min_{p_i \in {}^\circ t_j} \left\{ \frac{m_i}{W_{pr}(i, j)} \right\} \quad (\text{I.26})$$

Cette transition validée peut être franchie avec une quantité réelle de marque $\alpha_c \in \mathbb{R}$ telle que $0 < \alpha_c \leq \eta_c(t_j, M_h)$ et conduit à un nouveau marquage M_h pour lequel seule la partie continue M_c est modifiée. La variation du marquage ΔM_h est donnée par l'équation (I.27) :

$$\Delta M_h = \alpha_c \cdot \begin{bmatrix} 0_{n_d \times 1} \\ W^{cc}(:, j) \end{bmatrix} \quad (\text{I.27})$$

Le vecteur de franchissements des transitions d'un RdPHE est défini par $\Gamma_h = \begin{bmatrix} \Gamma_d^t & \Gamma_c^t \end{bmatrix}^t$ où $\Gamma_d \in \mathbb{N}_{\geq 0}^{q_d}$ désigne le vecteur de franchissements des transitions discrètes et $\Gamma_c \in \mathbb{R}_{\geq 0}^{q_c}$ représente le vecteur de franchissements des transitions continues.

L'espace d'atteignabilité d'un RdPHE autonome, noté $R_h(\mathcal{N}_h, M_{0h})$, est constitué de l'ensemble de tous les marquages atteignables à partir du marquage initial. Si $M_h \in R_h(\mathcal{N}_h, M_{0h})$, l'équation d'état qui traduit l'évolution du RdPH est donnée par (I.28) :

$$M_h = M_{0h} + W \cdot \Gamma_h \quad (\text{I.28})$$

La temporisation du RdPHE est introduite dans la section suivante.

I.3.2 Présentation du réseau de Petri Hybride élémentaire temporisé

Un RdPHE temporisé est un RdPHE auquel des contraintes de temps sont ajoutées. Le marquage courant du RdPHE temporisé ainsi que le vecteur de franchissements dépendent du temps τ . Ils sont désormais respectivement notés par $M_h(\tau) = \begin{bmatrix} M_d(\tau)^t & M_c(\tau)^t \end{bmatrix}^t$ et $\Gamma_h(\tau) = \begin{bmatrix} \Gamma_d(\tau)^t & \Gamma_c(\tau)^t \end{bmatrix}^t$. Dans ce mémoire et contrairement à [LHS05], aucun intervalle temporel de franchissement n'est spécifié, seules des contraintes de durée minimale de franchissement sont considérées, ce qui implique qu'à chaque transition discrète est associée une seule valeur. Un RdPHE temporisé est alors représenté par le quadruplet $\langle \mathcal{N}_h, M_{0h}, D_{min}, \lambda \rangle$. D'une part, $D_{min} = (d_{min j})_{j=1, \dots, q_d} \in \mathbb{R}_{\geq 0}^{q_d}$ représente la durée minimale de franchissement des transitions discrètes. Autrement dit, une transition discrète t_j validée par le marquage discret courant à la date τ ne peut être franchie qu'à partir de la date $\tau + d_{min j}$. Si $d_{min j} = 0$, la transition t_j est immédiatement franchissable [DA92]. D'autre part, $\lambda = (\lambda_j)_{j=q_d+1, \dots, q} \in \mathbb{R}_{> 0}^{q_c}$ dénote la vitesse maximale de franchissement des transitions continues.

En ce qui concerne la partie discrète, le marquage discret qui résulte du franchissement des transitions discrètes, donné par le vecteur Γ_d à la date τ , est exprimé par (I.29) :

$$M_d(\tau^+) = M_d(\tau) + W^{dd} \cdot \Gamma_d(\tau) \quad (\text{I.29})$$

La notation τ^+ correspond à l'instant qui succède immédiatement le franchissement des transitions discrètes à la date τ . La dérivée du marquage continu par rapport au temps est définie par l'équation (I.30) :

$$\dot{M}_c(\tau^+) = W^{cc} \cdot \dot{\Gamma}_c(\tau^+) \quad (\text{I.30})$$

Contrairement aux transitions discrètes, le franchissement des transitions continues s'effectue selon un flux continu en considérant le flux comme la quantité de marques franchie pendant une unité de temps. À la date τ^+ , le flux des transitions continues est défini par $F_c(\tau^+) = \dot{\Gamma}_c(\tau^+)$ avec $F_c(\tau^+) = (f_j(\tau^+))_{j=q_d+1, \dots, q}$.

Il existe plusieurs sémantiques pour approcher le flux au travers d'une transition temporisée dont les plus utilisées sont les sémantiques serveurs finis et infinis (vitesses constantes et vitesses variables). Contrairement à [DA01, LHS05, GA07, Gho12, JCBM13] où les auteurs ont choisi de présenter le modèle continu avec des vitesses maximales constantes, ces travaux de recherche sont basés essentiellement sur la sémantique serveurs infinis, à l'instar des travaux présentés dans [JMV11]. Le flux de chaque transition continue t_j , donné par l'équation (I.31), correspond au produit de la vitesse maximale de franchissement par le degré de validation. Il mesure la quantité maximale de jetons qui peut franchir $t_j \in T_c$ par unité de temps :

$$\forall t_j \in T_c, f_j(\tau^+) = \lambda_j \cdot \eta_c(t_j, M_h(\tau^+)) \quad (\text{I.31})$$

L'espace d'atteignabilité d'un RdPHE est noté $R_h(\mathcal{N}_h, M_{0h}, D_{min}, \lambda)$. Il peut être partitionné en K régions. Ces régions sont définies uniquement par rapport aux transitions continues. Le nombre total de régions K est donné par (I.32) :

$$K \leq \prod_{j=q_d+1}^q |{}^\circ t_j|, t_j \in T_c \quad (\text{I.32})$$

Chaque région, $\forall \zeta = 1, \dots, K$, correspond à une configuration unique caractérisée par la matrice de contraintes A_ζ^c telle que $A_\zeta^c \in \mathbb{R}_{\geq 0}^{q_c \times n}$. Chaque ligne, $j = 1, \dots, q_c$, de A_ζ^c ne

possède qu'un seul élément non-nul à la $i^{\text{ème}}$ position :

$$A_{\zeta}^c(j, i) = \begin{cases} \frac{1}{W_{pr}(i, j)}, & \text{si } p_i \text{ est la place critique de } t_j \in T_c \\ 0, & \text{sinon} \end{cases} \quad (\text{I.33})$$

Dans ce qui suit, la représentation suivante de la matrice de contraintes A_{ζ}^c est considérée : $A_{\zeta}^c = \begin{bmatrix} A_{\zeta}^{cd} & A_{\zeta}^{cc} \end{bmatrix}$ telle que $A_{\zeta}^{cd} \in \mathbb{R}_{\geq 0}^{q_c \times n_d}$ et $A_{\zeta}^{cc} \in \mathbb{R}_{\geq 0}^{q_c \times n_c}$. Soient $R(\tau)$ et $A^c(\tau)$ la région de $M_h(\tau^+)$ et sa matrice de contraintes, la représentation matricielle du flux des transitions continues est désormais donnée par (I.34) :

$$F_c(\tau^+) = \Lambda \cdot A^c(\tau) \cdot M_h(\tau^+) \quad (\text{I.34})$$

Le temps τ est échantillonné grâce à une horloge externe de période d'échantillonnage θ comme dans le cas du RdPC, θ doit satisfaire la condition donnée par (I.17), mais dans un RdPH seules les transitions continues sont prises en compte :

$$\forall p_i \in P_c, \quad \sum_{t_j \in p_i^{\circ} / t_j \in T_c} \lambda_j \cdot \theta < 1 \quad (\text{I.35})$$

Le RdPHE (ou le RdPHDE) à temps discret est noté θ -RdPHE (ou θ -RdPHDE) = $\langle \mathcal{N}_h, M_{0h}, D_{min}, \lambda, \theta \rangle$.

Les transitions discrètes sont synchronisées sur l'horloge externe et sont franchies dès que leur durée minimale de franchissement est écoulée. Par conséquent, $M_d(\tau_{k+1}) = M_d(\tau_{k+})$ et $M_c(\tau_{k+1}) = M_c(\tau_k)$. À chaque date $\tau_k > 0$, l'équation fondamentale d'un θ -RdPHE ou d'un θ -RdPHDE s'écrit selon le système d'équations (I.36) :

$$\begin{cases} M_d(\tau_{k+1}) = M_d(\tau_k) + W^{dd} \cdot \Gamma_d(\tau_k) \\ M_c(\tau_{k+1}) = M_c(\tau_k) + \theta \cdot W^{cc} \cdot F_c(\tau_{k+}) \end{cases} \quad (\text{I.36})$$

Désignant respectivement par R_k et A_k^c la région et la matrice de contraintes de M_h à la date τ_k , le flux de franchissement des transitions continues en régime libre est exprimé par l'équation (I.37) :

$$\begin{aligned} F_c(\tau_{k+}) &= \Lambda \cdot A_k^c \cdot M_h(\tau_{k+}) \\ &= \Lambda \cdot (A_k^{cc} \cdot M_c(\tau_k) + A_k^{cd} \cdot M_d(\tau_{k+})) \\ &= \Lambda \cdot (A_k^{cc} \cdot M_c(\tau_k) + A_k^{cd} \cdot (M_d(\tau_k) + W^{dd} \cdot \Gamma_d(\tau_k))) \\ &= \Lambda \cdot A_k^c \cdot M_h(\tau_k) + \Lambda \cdot A_k^{cd} \cdot W^{dd} \cdot \Gamma_d(\tau_k) \end{aligned} \quad (\text{I.37})$$

En remplaçant dans (I.36) $F_c(\tau_{k+})$ par son expression donnée par (I.37), le marquage continu $M_c(\tau_{k+1})$ peut s'exprimer selon (I.38) :

$$\begin{aligned} M_c(\tau_{k+1}) &= M_c(\tau_k) + \theta \cdot W^{cc} \cdot \Lambda \cdot A_k^c \cdot M_h(\tau_k) + \theta \cdot W^{cc} \cdot \Lambda \cdot A_k^{cd} \cdot W^{dd} \cdot \Gamma_d(\tau_k) \\ &= G_k^c \cdot M_c(\tau_k) + G_k^d \cdot (M_d(\tau_k) + W^{dd} \cdot \Gamma_d(\tau_k)) \end{aligned} \quad (\text{I.38})$$

où G_k^c et G_k^d sont définis selon l'équation (I.39) :

$$\begin{cases} G_k^c = I_{n_c} + \theta \cdot W^{cc} \cdot \Lambda \cdot A_k^{cc} \\ G_k^d = \theta \cdot W^{cc} \cdot \Lambda \cdot A_k^{cd} \end{cases} \quad (\text{I.39})$$

I.3.3 Domaines d'application du réseau de Petri hybride

Le RdPH permet de modéliser un grand nombre de systèmes. La littérature a mis en évidence l'efficacité ainsi que l'utilité de la modélisation par ce réseau dans l'étude du comportement, l'évaluation de performance, la simulation, l'optimisation ainsi que la commande et le contrôle de différents types de systèmes dans différents domaines, y compris les systèmes manufacturiers, les systèmes de trafic et de chaîne logistique. En particulier, dans [BGM00], les auteurs ont commandé les systèmes manufacturiers modélisés par les RdPH temporisés tels que les transitions continues sont définies sous l'approximation serveurs finis et les transitions discrètes sont immédiates, déterministes ou stochastiques. Les RdPH sont adaptés pour la modélisation des activités distribuées au sein des systèmes complexes. Dans ce cadre, [DFGS08] ont étudié les systèmes manufacturiers distribués. La structure globale de ces systèmes fait intervenir plusieurs entités, à savoir les fournisseurs, les fabricants, les assembleurs, les transporteurs et les consommateurs. Les auteurs ont utilisé les RdPH afin de proposer un modèle modulaire hybride de ses systèmes. En effet, chaque entité a été indépendamment décrite par un RdPH. Le modèle complet du système manufacturier distribué résulte de l'interaction entre ces différents modèles élémentaires. Ce modèle a servi par la suite pour simuler, analyser et optimiser les performances de ces systèmes pour différentes conditions de fonctionnement.

Dans [JMV11], la contribution consiste à présenter un outil intégré dans l'environnement MATLAB, appelé " SimHPN (Hybride Petri Net Simulator)". Ce logiciel permet de simuler et d'analyser efficacement les systèmes manufacturiers complexes qui sont modélisés par les RdPH pour lesquels les transitions continues sont définies sous la sémantique serveurs infinis et les transitions discrètes sont déterministes ou stochastiques (loi exponentielle). D'une part, cet outil est caractérisé par une interface graphique conviviale. D'autre part, il utilise plusieurs solveurs standards d'équations, disponibles dans MAT-

LAB, afin de résoudre chaque problème d'optimisation (linéaire ou quadratique).

En ce qui concernent les systèmes de trafic, [FIMU14] ont proposé un modèle modulaire basé sur les RdPH dans le but de modéliser la circulation autoroutière. Cette méthode suit la méthode de décomposition en sous-modèles présentée dans [DFGS08]. Le modèle proposé combine la dynamique temporelle et la dynamique événementielle. La première correspond au flux classique de circulation tandis que la deuxième correspond au comportement du réseau en présence des accidents et des stratégies de recouvrement. Ils ont également conçu une technique de commande en ligne inspirée de la commande prédictive. Cette méthode permet de prédire le comportement du système et de proposer une limite de vitesse si le trafic est congestionné. Le formalisme développé a été directement appliqué à un tronçon d'une autoroute dans le nord-est de l'Italie.

Par souci de brièveté, nous énumérons dans le tableau I.1, d'autres domaines d'application des RdPH.

TABLE I.1 – Domaines d'application des RdPH

Contributions et domaines d'application	Auteurs
Proposer un modèle modulaire hybride basé sur le RdPH afin de résoudre le problème d'ordonnancement des opérations d'exploitation du pétrole brut dans les raffineries.	[WCCZ10] [WCCZ11]
Proposer un modèle modulaire hybride basé sur le RdPH pour la gestion et le contrôle de la chaîne logistique au niveau opérationnel.	[DFIM09]
Générer un modèle de RdPH afin d'évaluer les propriétés quantitatives des gares ferroviaires.	[KHM06]
Utiliser les RdPH temporisés afin d'analyser les performances des réseaux de capteurs sans fil.	[GMLM13]

Conclusion

Ce premier chapitre a rappelé les concepts fondamentaux du RdP et a également introduit les notations utilisées dans la suite du mémoire. Le principe d'évolution du RdPD autonome et T-temporisé ainsi que les différents domaines d'application ont été abordés. Il ressort de la littérature que le RdPD souffre du problème de l'explosion combinatoire du nombre d'états. La fluidification, utilisée comme technique de relaxation, a été présentée pour surmonter cet inconvénient majeur. Une brève description des concepts de base

des RdPC ainsi que les domaines d'application ont été présentés. Cette classe de réseaux n'autorise ni les conditions logiques ni les actions discrètes, le RdPH a ainsi été introduit. Il résulte d'une fluidification partielle du RdPD. Une étude bibliographique sur les différents types du RdPH a été menée, pour se concentrer sur les RdPHE et les RdPHDE.

Tout au long de ce chapitre, seuls les RdPD, les RdPC et les RdPHE (ou RdPHDE) en régime libre ont été considérés. Pourtant, les systèmes modélisés peuvent être soumis à une action extérieure de commande en vue d'atteindre un état désiré. Le chapitre suivant prend en compte les actions de commande qui peuvent être appliquées aux transitions (continues et/ou discrètes) commandables. Il récapitule, pour chaque type de RdP (continu, discret et hybride), les différentes approches de commande qui existent dans la littérature.

Sommaire

Introduction	42
II.1 Commande des réseaux de Petri discrets	42
II.1.1 Commande supervisée	43
II.1.2 Commande d'ordonnancement	44
II.2 Commande des réseaux de Petri continus	47
II.2.1 Commande des réseaux de Petri continus sous la sémantique serveurs infinis	48
II.2.1.1 Commande multiplicative des réseaux de Petri continus	48
II.2.1.2 Commande additive des réseaux de Petri continus . .	49
II.2.2 Éléments d'état de l'art sur la commande des réseaux de Petri continus sous la sémantique serveurs finis	53
II.2.3 Éléments d'état de l'art sur la commande des réseaux de Petri continus sous la sémantique serveurs infinis	54
II.2.3.1 Commande optimale en régime permanent	55
II.2.3.2 Commande "ON/OFF"	55
II.2.3.3 Commande pour les trajectoires linéaires et linéaires par morceaux	57
II.2.3.4 Commandes "ON/OFF+" et "B-ON/OFF"	58
II.2.3.5 Commande proportionnelle	59
II.2.3.6 Commande prédictive	60
II.3 Commande des réseaux de Petri hybrides	66
II.3.1 Commande des réseaux de Petri hybrides élémentaires	66
II.3.2 Éléments d'état de l'art sur la commande des réseaux de Petri hybrides	68
II.4 Synthèse de l'analyse bibliographique et choix de la commande	70
Conclusion	72

Introduction

DANS ce chapitre, nous nous sommes intéressés à la commande des RdP. Selon le concept de commandabilité, un système est dit commandable s'il est possible de le faire évoluer entre deux états quelconques en temps fini [SJMV11]. Le but de la commande des RdP est alors d'atteindre n'importe quel état désiré à partir d'un état actuel. Cet état désiré est défini soit par le marquage désiré, soit par une configuration composée d'un marquage et d'un flux désirés en régime permanent [MRTRS08, SJMV11, JVMS13]. Afin d'atteindre cet objectif, des actions de commande sont appliquées uniquement sur les transitions commandables.

La commande des RdP a fait l'objet de nombreux travaux de recherche menés ces trente dernières années. Différents concepts de commande peuvent être considérés. En particulier, les techniques de commande destinées aux RdPD et celles consacrées aux RdP fluidifiés (RdPC et RdPH). La section II.1 présente différentes méthodes de commande relatives aux RdPD, à savoir les commandes supervisées et d'ordonnancement. Dans la section II.2, nous nous focalisons sur les RdPC munis de la sémantique serveurs infinis pour lesquels toutes les transitions sont considérées commandables. Nous supposons que seules les actions de commande qui diminuent le flux de transitions sont admissibles. Dans un premier temps, le principe des commandes additive et multiplicative est expliqué et les différentes contraintes sur le flux de franchissement sont précisées. Les méthodes de commande du RdPC, développées durant les dernières décennies, sont ensuite présentées. La section II.3 explique le principe général de commande des RdPH et introduit quelques techniques de commande. Une synthèse de l'analyse bibliographique est finalement menée. Cette étude sert à choisir la commande la mieux adaptée pour commander tout type de RdP. Cette commande est la commande prédictive malgré quelques inconvénients qui seront signalés en vue de les contourner dans l'approche proposée.

II.1 Commande des réseaux de Petri discrets

Dans le chapitre précédent, nous avons expliqué l'évolution du RdPD en régime libre. Or, des actions de commande peuvent être appliquées aux transitions afin d'atteindre un objectif donné. Le RdPD est alors dit forcé ou commandé. Une transition est dite commandable si une fois validée (et la durée minimale de franchissement écoulée dans le cas des transitions discrètes temporisées), son franchissement peut être empêché ou retardé par un agent de commande [GDS92].

Soit T_d^{comm} l'ensemble des transitions commandables, le RdPD T-temporisé commandé est désormais représenté par le quadruplet $\langle \mathcal{N}_d, M_{0d}, D_{min}, T_d^{comm} \rangle$.

Plusieurs travaux ont été menés dans le cadre de la conception de la commande et plusieurs méthodes ont été développées pour ce type de réseaux selon le but désiré. Nous distinguons deux types de commandes, la commande supervisée (“supervisory control”) et la commande d’ordonnancement (“scheduling control”).

Cette section présente brièvement quelques références bibliographiques de ces deux types de commandes appliquées aux RdPD.

II.1.1 Commande supervisée

La théorie de la supervision des SED a été introduite par Ramadge et Wonham [RW87,RW89]. D’une manière générale, le but de la supervision est d’assurer le bon fonctionnement du système. Le principe de la supervision est donné par la figure II.1. Dans ce schéma, le procédé est un SED qui génère des événements. Il est relié à un superviseur qui a pour rôle d’interdire ou d’autoriser l’occurrence des événements dans le procédé, ce qui permet de modifier son fonctionnement dans le but de respecter des spécifications.

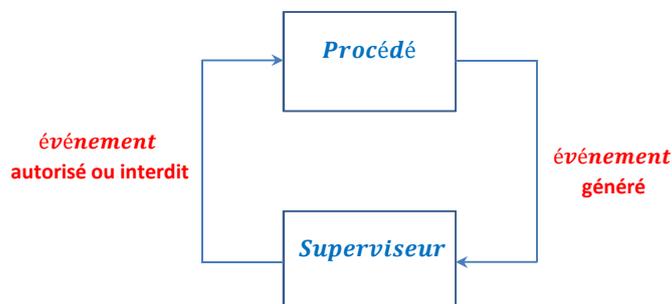


FIGURE II.1 – Principe de la supervision

Dans le cas du RdP, les spécifications imposées par la commande supervisée sont l’évitement des blocages, la vivacité, la commandabilité et le nombre maximal de tâches simultanées [BCP13]. L’objectif est de construire une commande permissive qui restreint convenablement l’espace d’atteignabilité du RdP à l’ensemble des marquages légaux (ou non-interdits) définis par les spécifications imposées [BCP13].

Plusieurs techniques efficaces ont été développées pour la synthèse de cette commande. En particulier, les auteurs [GDS92, YaLA96, BCG06, DA08, BCP13] ont exprimé les marquages légaux en introduisant un ensemble d’inégalités linéaires sur le marquage.

Ces inégalités s'appellent contraintes d'exclusion mutuelle ("Generalized Mutual Exclusion Constraint"). Si toutes les transitions sont commandables, ces contraintes sont facilement mises en œuvre par le biais de places de supervision pour créer des invariants de marquage adéquats (un invariant de marquage pour chaque contrainte) [GDS92]. Il a été également démontré que cette méthode offre une permissivité maximale [BCG06]. Cependant, le nombre de contraintes augmente considérablement en fonction des états interdits. [GDS92] a prouvé qu'il est possible de simplifier les contraintes. Pour que le superviseur soit minimal (nombre minimal de contraintes) et que la détermination des marquages légaux soit faite de manière permissive, [BCP13] ont utilisé une méthode "*branch&bound*". Cette méthode permet de résoudre un problème d'optimisation combinatoire, en partitionnant l'espace des solutions en sous-ensembles ("*branch*") et en majorant ensuite chaque sous-ensemble ("*bound*"), ce qui permet de ne considérer que les sous-ensembles qui contiennent une solution meilleure que la solution courante. Une autre méthode basée sur la notion de "over-state" (l'ensemble des marquages qui ont les mêmes propriétés) a été proposée dans [DA08] pour réduire systématiquement le nombre d'états et des contraintes. Cette méthode n'est applicable que dans le cas des RdP saufs.

[GDS92] ont étudié l'utilisation de contraintes d'exclusion mutuelle en considérant des transitions discrètes non-commandables, ce qui engendre davantage d'états interdits et réduit la permissivité de la commande. Ils ont signalé que les contraintes peuvent être imposées par les places de supervision uniquement dans le cas des RdPD saufs et conservatifs. Dans le but de résoudre ce problème, dans [YaLA96] les auteurs ont proposé quelques restrictions sur l'espace d'atteignabilité. Ils ont utilisé des conditions suffisantes afin de garantir la commandabilité structurelle de l'ensemble des marquages légaux. Dans ce même contexte, [BCG06] ont proposé un algorithme qui permet de calculer les places de supervision structurellement commandables.

II.1.2 Commande d'ordonnement

Les RdPD modélisent efficacement les SED et spécialement les systèmes manufacturiers. À cet égard, de nombreuses contributions sur la commande d'ordonnement des systèmes manufacturiers flexibles ont été proposées. L'ordonnement consiste à choisir la date de début ainsi que les ressources à allouer aux différentes tâches à traiter. Ce dernier objectif équivaut à déterminer la séquence de franchissements qui permet d'atteindre un marquage désiré à partir d'un marquage initial. Elle peut également intégrer une procédure de supervision afin de donner lieu à une commande d'ordonnement sans blocage ("deadlock-free scheduling control") [BPG04].

La méthode présentée dans [LD94] formalise le problème d'ordonnancement en utilisant les paramètres du RdPD. Une fois le système modélisé, son évolution peut être complètement décrite par la trajectoire du marquage. Ceci donne la possibilité de suivre les différents comportements possibles du système à l'aide du graphe d'atteignabilité. [LD94] ont proposé quelques fonctions heuristiques basées sur l'algorithme de recherche A^* afin d'estimer la durée de la trajectoire allant du marquage courant jusqu'au marquage désiré M_{refd} et de générer uniquement la portion nécessaire de ce graphe contenant le trajet optimal (la séquence de franchissements des transitions). Ce trajet correspond à l'ordonnancement optimal ou quasi-optimal du système modélisé. Cette méthode évite également les situations de blocage en introduisant la notion de places intermédiaires. Les performances de cette méthode et l'optimalité de la solution obtenue sont directement liées à la fonction heuristique $f(M_d)$ utilisée. Cette fonction est définie par $f(M_d) = g(M_d) + h(M_d)$ où $h(M_d)$ représente une estimation de la durée restante pour atteindre le marquage désiré et $g(M_d)$ représente la durée actuelle la plus faible obtenue à partir du marquage initial M_{0d} jusqu'au marquage courant M_d . D'après [MYK02], l'effort de calcul engendré par l'utilisation d'une fonction heuristique admissible peut être très important même pour des problèmes de petite dimension. Afin de surmonter cette difficulté, les auteurs de [MYK02] ont utilisé une méthode hybride appelée DLSS* ("Dynamic Look-Ahead Stage Search") qui vise à réduire le champ d'application de A^* pour se limiter aux recherches locales. Finalement, la génération des trajets inutiles dans le cadre de l'exploration du graphe d'atteignabilité a été systématiquement évitée. Dans ce contexte, les auteurs de [KSN07] ont proposé deux sous-modèles pour les systèmes de production flexibles en utilisant les RdPD temporisés afin de décrire les comportements physique et logique des différentes tâches. Ils ont été motivés par deux problèmes réels, le premier consiste à minimiser le temps maximal d'achèvement des tâches et le deuxième vise à minimiser le retard total. Afin de résoudre le premier problème, un algorithme de recherche graphique rapide et réactif basé sur RTA* ("Real Time A*"), a été développé. Cet algorithme a été combiné avec trois règles de supervision pour réduire l'espace de recherche. Puis, il a été appliqué au RdPD inverse (en inversant le sens des arcs) pour traiter le deuxième problème.

[LL15] ont proposé une commande inspirée de la commande prédictive (détaillée dans la section suivante) qui explore partiellement l'espace d'atteignabilité et qui détermine les séquences de commande de longueur minimale ou quasi-minimale. En effet, à chaque instant et sur un horizon de N_d franchissements discrets, cette méthode explore, à partir du marquage discret courant, l'espace d'atteignabilité. Elle détermine ensuite, de manière incrémentale, toutes les séquences de commande contenant au plus N_d transitions discrètes commandables. Les marquages résultants de ces séquences sont également calculés. Une

fonction coût J est mise en place pour évaluer l'écart quadratique entre le marquage discret prédit et le marquage discret désiré et permet de déterminer les marquages qui la minimisent :

$$J(M_d) = (M_d - M_{refd})^t \cdot (M_d - M_{refd}) \quad (\text{II.1})$$

Les séquences de commande correspondantes sont ensuite déterminées, une transition choisie parmi toutes les premières transitions de ces séquences est franchie et le problème d'optimisation est réitéré, en partant du marquage calculé, jusqu'à atteindre le marquage désiré. Pour mieux expliquer cette méthode, la figure II.2 montre un exemple illustratif de RdPD non-temporisé pour lequel toutes les transitions sont commandables, $N_d = 2$, le marquage initial est $[1 \ 0 \ 0]^t$ et le marquage de référence est $[5 \ 5 \ 5]^t$. La figure II.3 présente l'exploration partielle du graphe d'atteignabilité [LL15] :

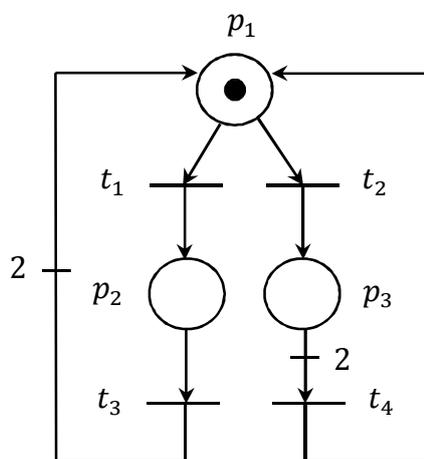


FIGURE II.2 – Exemple d'un RdPD

Au premier pas de calcul, la partie explorée du graphe d'atteignabilité est encadrée par un rectangle. Elle correspond à l'ensemble des marquages atteignables avec une profondeur de 2. Les séquences de commande possibles sont t_1 , t_2 et $t_1 t_3$. Le critère de performance J est calculé pour chaque séquence (indiqué sur le schéma à côté du marquage). $t_1 t_3$ est la séquence pour laquelle la valeur de J est minimale ($J = 59$), la transition t_1 est alors franchie (le franchissement de la transition sélectionnée est indiqué par une flèche sur le schéma), le marquage qui résulte de ce franchissement est $M_d = [0 \ 1 \ 0]^t$. Le processus est alors réitéré au pas suivant.

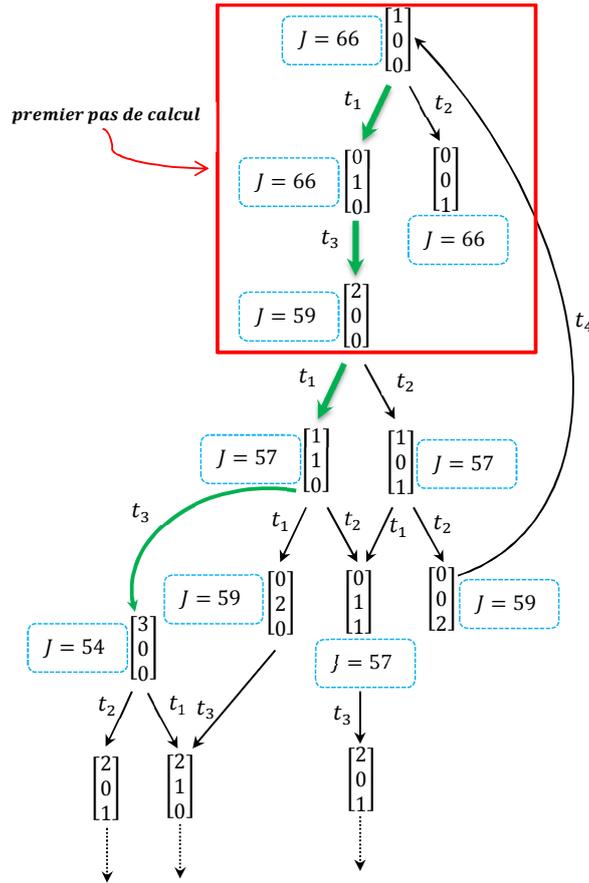


FIGURE II.3 – Processus d’exploration partielle du graphe d’atteignabilité pour le RdPD de la figure II.2

Les méthodes de commande d’ordonnancement présentées dans cette partie ne sont malheureusement pas applicables aux systèmes avec un grand nombre d’états car elles peuvent mener au problème de l’explosion combinatoire du nombre d’états. Pour s’affranchir de ce problème et pour éviter l’exploration du graphe d’atteignabilité, la commande développée dans le chapitre III génère la séquence de commandes discrètes du RdPD en utilisant la commande continue du RdPC sous-jacent. Dans la littérature, il existe plusieurs méthodes de commande destinées aux RdPC qui sont détaillées dans la section suivante.

II.2 Commande des réseaux de Petri continus

La plupart des travaux de recherche qui se trouvent dans la littérature traitent la commande des RdPC sous la sémantique serveurs infinis [SJM11], car pour une large classe de RdPC, cette sémantique fournit une meilleure approximation des RdPD [MRTS08, MRS06]. Pour cette raison, cette section présente le principe de commandabilité

et de commande des RdPC définis sous cette sémantique. Quelques notions de commande du RdPC sous la sémantique serveurs finis seront aussi brièvement présentées. Un état de l'art des différentes approches de commande, sous la sémantique serveurs infinis, est ensuite adressé.

II.2.1 Commande des réseaux de Petri continus sous la sémantique serveurs infinis

Un RdPC temporisé commandé est soumis à une action de commande extérieure appliquée aux transitions. Cette action est désignée par $U_c = (u_j)_{j=1, \dots, n} \in \mathbb{R}_{\geq 0}^n$ telle que u_j est l'action de commande appliquée à la transition t_j . Le flux de franchissement forcé est désigné par $X_c = (x_j)_{j=1, \dots, q}$. L'ensemble des transitions est partitionné en deux sous-ensembles disjoints, à savoir l'ensemble des transitions commandables et l'ensemble des transitions non-commandables. Soient T_c^{comm} l'ensemble des transitions continues commandables. Le RdPC et le θ -RdPC forcés sont respectivement représentés par le quadruplet $\langle \mathcal{N}_c, M_{0c}, \lambda, T_c^{comm} \rangle$ et le quintuplet $\langle \mathcal{N}_c, M_{0c}, \lambda, \theta, T_c^{comm} \rangle$.

D'après [SR04], les transitions modélisent les actionneurs dans une machine réelle ou les serveurs dans un système informatique qui ne devraient pas dépasser leur vitesse nominale. Par conséquent, l'action de commande ne peut que ralentir ou arrêter leur activité. Le vecteur U_c intervient donc afin de réduire le flux maximal de franchissement donné par le flux en régime libre F_c des transitions commandables. À l'inverse, les transitions non-commandables sont franchies avec leur flux maximal de franchissement. Si toutes les transitions sont commandables, le comportement global d'un RdPC commandé, dans la région $R(\tau)$ du marquage $M_c(\tau)$, est régi par le système d'équations (II.2) :

$$\begin{cases} \dot{M}_c(\tau) = W \cdot X_c(\tau) \\ 0_{q \times 1} \leq X_c(\tau) \leq \Lambda \cdot A(\tau) \cdot M(\tau) \end{cases} \quad (\text{II.2})$$

Dans la littérature, il existe deux approches de commande, la commande additive [SR04] et la commande multiplicative [LLD13].

II.2.1.1 Commande multiplicative des réseaux de Petri continus

Concernant la commande multiplicative, le vecteur de commandes U_c pondère le flux de franchissement en régime libre de la transition commandable. Le flux forcé et le marquage

résultant, à la date τ_k , sont respectivement donnés par (II.3) et (II.4) :

$$\begin{aligned} X_c(\tau_k) &= \text{diag}(U_c(\tau_k)) \cdot F_c(\tau_k) \\ &= \text{diag}(U_c(\tau_k)) \cdot \Lambda \cdot A_k \cdot M_c(\tau_k) \end{aligned} \quad (\text{II.3})$$

$$\begin{aligned} M_c(\tau_{k+1}) &= M_c(\tau_k) + \theta \cdot W \cdot X_c(\tau_k) \\ &= M_c(\tau_k) + \theta \cdot W \cdot \text{diag}(U_c(\tau_k)) \cdot \Lambda \cdot A_k \cdot M_c(\tau_k) \end{aligned} \quad (\text{II.4})$$

tel que

$$\begin{cases} 0 \leq u_j \leq 1, & \text{si } t_j \in T_c^{\text{comm}} \\ u_j = 1, & \text{sinon} \end{cases} \quad (\text{II.5})$$

La première inégalité de (II.5) assure que le flux forcé est toujours non-négatif et la deuxième contraint le flux à ne pas dépasser le flux maximal de franchissement.

L'approche multiplicative a été traitée dans [LL11], la commande suppose que toutes les transitions soient commandables et consiste à déterminer la vitesse maximale de franchissement μ_j de chaque transition t_j telle que $0 < \mu_j < \lambda_j$ avec λ_j la valeur maximale admissible par μ_j . Cette commande s'intéresse aux RdPC avec vitesse maximale constante par morceaux pour lesquels la vitesse de franchissement reste constante dans une même région et commute lors du passage d'une région à une autre jusqu'à atteindre un marquage désiré.

II.2.1.2 Commande additive des réseaux de Petri continus

En ce qui concerne l'approche additive, le vecteur de commandes est soustrait du flux maximal de franchissement. De même que pour l'approche multiplicative, l'action de commande est dynamiquement bornée. La borne supérieure correspond au flux maximal de franchissement afin d'imposer un flux non-négatif et la borne minimale garantit que la variation correspond bien à une diminution :

$$\begin{cases} 0 \leq u_j \leq \lambda_j \cdot A_k(:, j) \cdot M_c(\tau_k), & \text{si } t_j \in T_c^{\text{comm}} \\ u_j = 0, & \text{sinon} \end{cases} \quad (\text{II.6})$$

Le comportement d'un θ -RdPC commandé en temps continu, dans la région R_k , est

donné par le système d'équations (II.7) :

$$\begin{cases} M_c(\tau_{k+1}) = M_c(\tau_k) + \theta \cdot W \cdot X_c(\tau_k) \\ X_c(\tau_k) = F_c(\tau_k) - U_c(\tau_k) = \Lambda \cdot A_k \cdot M_c(\tau_k) - U_c(\tau_k) \\ 0_{q \times 1} \leq U_c(\tau_k) \leq \Lambda \cdot A_k \cdot M_c(\tau_k) \end{cases} \quad (\text{II.7})$$

Finalement, le marquage en régime forcé est défini par (II.8) :

$$M_c(\tau_{k+1}) = B_k \cdot M_c(\tau_k) - \theta \cdot W \cdot U_c(\tau_k) \quad (\text{II.8})$$

L'approche additive et l'approche multiplicative sont équivalentes [SR04]. Dans ce qui suit, nous utilisons l'approche additive.

Le but de la commande consiste à faire évoluer le système jusqu'à atteindre une configuration désirée [SR04]. Cette configuration est définie par le couple (M_{refc}, X_{refc}) où M_{refc} correspond au marquage désiré qui doit être atteignable à partir de M_{0c} , et X_{refc} désigne le flux désiré. La configuration désirée doit être un point d'équilibre [MRTRS08], une fois atteinte, elle doit être maintenue. Considérons le RdPC commandé en temps continu, d'après l'équation d'état $\dot{M}_c(\tau) = W \cdot X_c(\tau)$, si $M_c(\tau) = M_{refc}$, $X_c(\tau) = X_{refc}$ doit satisfaire les conditions données par (II.9) :

$$\begin{cases} X_{refc} \geq 0_{q \times 1} \\ X_{refc} \leq \Lambda \cdot A_k \cdot M_{refc} \\ W \cdot X_{refc} = 0_{n \times 1} \end{cases} \quad (\text{II.9})$$

Plusieurs méthodes ont été mises en place afin de déterminer X_{refc} . En fait, X_{refc} dépend en premier lieu de la structure du réseau. Dans [JVMS13], les auteurs ont précisé que le marquage désiré peut être maintenu avec un flux non-nul à condition que le réseau possède au moins un T-semi-flot. À titre d'exemple, pour le RdP mono T-semi-flot, X_{refc} peut être simplement exprimé par $X_{refc} = v \cdot Y$, où $v \in \mathbb{R}_{\geq 0}$ et où Y est le T-semi-flot [MRTRS08]. Dans le cas contraire, l'unique solution pour maintenir le marquage désiré est de poser un flux désiré nul $X_{refc} = 0_{q \times 1}$ [JVMS13].

L'objectif de commande, ainsi défini, est lié à la notion de commandabilité. Dans la littérature, quelques travaux de recherche ont été consacrés à ce concept dans le but de donner des conditions de commandabilité des RdPC sous la sémantique serveurs infinis. D'une manière générale, un système est dit commandable s'il est possible de le faire évoluer d'un état initial vers un état final en temps fini par le biais d'une action de commande

non contrainte. Les auteurs de [JJRS05, MRTRS08] affirment que cette définition classique de système commandable n'est pas applicable au RdPC, car l'action de commande doit impérativement vérifier un ensemble de contraintes. D'autre part, la loi de conservation limite l'espace d'atteignabilité. À cet égard, dans [JJRS05], les auteurs ont supposé qu'un marquage M_{refc} est dit commandable à partir d'un marquage M_{0c} s'il existe une action de commande qui permet de l'atteindre (en temps fini ou infini) et de le maintenir. La commandabilité des RdPC conservatifs, consistants et sans synchronisation a été alors étudiée en se basant sur cette définition. En fait, l'étude proposée dans [JJRS05] revient à caractériser l'espace de commandabilité (c-à-d l'ensemble des marquages qui peuvent être atteints et maintenus) en fonction des transitions commandables. Si toutes les transitions sont commandables, cet espace coïncide avec l'espace d'atteignabilité du RdPC autonome. Dans le cas où une seule transition est commandable, cet espace est un segment de droite [JJRS05]. D'autres résultats sont également présentés lorsqu'un sous-ensemble de transitions est commandable.

Contrairement à ce qui a été développé dans [JJRS05], les auteurs de [MRTRS08, VRRS08] ont adapté la définition de la commandabilité du RdPC conformément à l'ensemble des marquages invariants défini par $Class(M_{0c}) = \{M_{refc} \in \mathbb{R}_{\geq 0}^n | B_Z^t \cdot M_{refc} = B_Z^t \cdot M_{0c}\}$ tel que B_Z est une base de P-flots. En effet, un RdPC est commandable si $\forall M_{0c} \geq 0_{n \times 1}$ et $M_{refc} \in Class(M_{0c})$, M_{refc} est atteignable et maintenu en temps fini ou infini. De plus, il a été démontré dans [MRTRS08] que toutes les transitions doivent obligatoirement être commandables pour qu'un RdPC conservatif soit commandable. La caractérisation de la commandabilité des RdPC avec des actions de commande bornées a été proposée dans [VRRS08]. Si toutes les transitions sont commandables, la commandabilité, par rapport à $Class(M_{0c})$, est garantie ssi le RdPC est consistant. Toutefois, cette condition n'est pas suffisante pour garantir la commandabilité si certaines transitions sont non-commandables. D'autres pistes ont été explorées dans [VRRS08], à savoir l'étude de la commandabilité conformément à l'ensemble de points d'équilibre. Ce choix est motivé par le fait que, la plupart du temps, l'objectif de commande est d'atteindre un point d'équilibre.

Dans le cas des RdPC munis des transitions non-commandables, le problème de commande devient beaucoup plus complexe, puisque le système peut ne pas être commandable, il existe dans la littérature des méthodes de commande qui permettent de résoudre ce problème. Par exemple, dans [VS09], les auteurs ont utilisé les conditions de commandabilité introduites dans [VRRS08] en présence de transitions non-commandables. Premièrement, il a été supposé que les marquages initial et final sont des points d'équi-

libre et qu'ils appartiennent à la même région. La commande en boucle fermée est définie par (II.10) avec U_{refc} la commande en régime permanent associée à M_{refc} et G la matrice de gain [VS09] :

$$U_c(\tau) = U_{refc} + G \cdot (M_c(\tau) - M_{refc}) \quad (\text{II.10})$$

La procédure de commande calcule le vecteur de gain G , puis détermine itérativement les marquages d'équilibre intermédiaires M'_{refc} qui appartiennent au segment de droite reliant M_{0c} et M_{refc} suite à la résolution d'un problème d'optimisation linéaire. La commande continue convenablement bornée, qui permet d'atteindre un marquage M'_{refc} , est alors calculée. En conséquence, M_{refc} est atteint en reliant les marquages intermédiaires M'_{refc} . Cette étude a été généralisée afin de prendre en considération plusieurs régions. Le problème de commande revient alors à faire évoluer le système entre les points d'équilibre appartenant aux frontières des régions consécutives, jusqu'à atteindre le marquage désiré. La commande est alors obtenue suite à la résolution du problème d'optimisation linéaire défini précédemment, dans chaque région R_ζ .

Une commande constante par morceaux, permettant d'atteindre un voisinage du marquage désiré en un nombre fini de pas, a été proposée dans [LL13]. Cette commande utilise la caractérisation des régions attractives en temps fini afin de prouver l'existence d'une commande bornée et constante par région qui assure l'atteignabilité de la région voisine selon la séquence des régions à traverser entre les marquages initial et final. Le domaine des commandes admissibles est ensuite déterminé à l'aide d'inégalités matricielles linéaires. En se basant sur ces résultats de calcul, un algorithme itératif a été mis en place afin de déterminer la commande constante par morceaux qui minimise le nombre de pas requis pour commuter entre les régions jusqu'à atteindre le marquage désiré. Il est à noter que cette optimisation locale ne garantit pas la minimalité du nombre de pas de l'ensemble de la trajectoire [LL13].

Dans le cadre de l'étude présentée dans [LLD13], la configuration n'inclut qu'un sous-ensemble de marquages (qui représente la sortie du système). Pour cette raison, la sensibilité des sorties par rapport aux entrées a été étudiée afin de sélectionner, en fonction de l'objectif désiré, les transitions qui doivent être commandées. Puis, une fonction de sensibilité $S(\tau_k)$, définie par le gradient des sorties en fonction des entrées, a été mise en place. L'erreur quadratique instantanée $E_{rr}(\tau_k)$ entre la sortie désirée et la sortie mesurée, a également été déterminée afin de calculer l'action de commande des transitions

commandables. Le vecteur de commandes $U_c(\tau_k)$ est alors donné par (II.11) :

$$\begin{cases} U_c(\tau_k) = U_c(\tau_{k-1}) + (S(\tau_k)^t \cdot S(\tau_k) + \theta \cdot I_q)^{-1} \cdot S(\tau_k)^t \cdot E_{rr}(\tau_{k-1}) \\ 0 \leq u_j(\tau_k) \leq 1, \text{ et } u_j(\tau_k) = 1, \forall t_j \notin T_c^{comm} \end{cases} \quad (\text{II.11})$$

En plus de la complexité du problème de commande lorsque les transitions ne sont pas toutes commandables, les auteurs de [MRTRS08] ont montré que, premièrement, si un marquage $M_{refc} > 0_{n \times 1}$ est atteignable dans le RdPC autonome et, deuxièmement, si toutes les transitions sont commandables, alors ce marquage est atteignable en nombre fini de pas dans le θ -RdPC commandé correspondant. À cet égard, dans le cadre de la conception d'une commande continue, ces deux conditions sont respectées ($M_{refc} > 0_{n \times 1}$ et toutes les transitions sont commandables). Par ailleurs, nous nous intéresserons uniquement, dans la sous-section II.2.3, aux techniques de commande définies sous la sémantique serveurs infinis. Avant d'entamer cette présentation, le paragraphe suivant rappelle quelques notions de base de la commande des RdPC définis sous la sémantique serveurs finis.

II.2.2 Éléments d'état de l'art sur la commande des réseaux de Petri continus sous la sémantique serveurs finis

La commande des RdPC sous la sémantique serveurs finis a les mêmes principes et le même objectif de commande que celle des RdPC sous la sémantique serveurs infinis. En effet, le flux de franchissement d'une transition commandable est ralenti par le vecteur de commandes U_c . De plus, chaque composante u_j , relative à chaque transition t_j , doit vérifier l'inégalité suivante : $0 \leq u_j \leq \lambda_j$. Trois cas peuvent être distingués afin de déterminer le flux de franchissement forcé [JBRS04] :

- Si t_j est fortement validée, le flux x_j est donné par $x_j = \lambda_j - u_j$
- Si t_j est faiblement validée, le flux x_j est calculé en considérant inégalité $0 \leq x_j \leq \lambda_j - u_j$
- Si t_j n'est pas validée, $x_j = 0$

Dans la littérature, il existe diverses méthodes de commande qui visent à calculer la commande optimale pour les RdPC sous la sémantique serveurs finis. Généralement, ces commandes sont divisées en deux groupes en fonction du modèle temporel considéré (temps continu ou temps discret) [SJMV11]. Une autre approche intermédiaire a été proposée dans [JBRS04]. Le principe de cette commande repose sur la transformation du RdPC sous la sémantique serveurs finis en RdPH équivalent dont l'évolution est décrite

par l'occurrence d'un événement discret. De manière plus précise, chaque pas de calcul ne correspond plus à l'écoulement de la période d'échantillonnage. Désormais, il correspond à l'occurrence d'un événement dans le RdPC (c-à-d une place continue vidée). En conséquence, la période d'échantillonnage n'est pas constante, elle est adaptée en fonction des événements survenus. Le RdPC est donc transformé en un système dynamique logique mixte événementiel ("event-driven mixed logical dynamical system") décrit par un ensemble d'égalités et d'inégalités linéaires qui font intervenir des variables réelles et booléennes [JBRS04].

Différents problèmes de commande peuvent être adressés via cette commande, y compris la minimisation du temps de convergence, la maximisation du flux en régime permanent ou bien l'optimisation d'une fonction multi-variable. Pour ce faire, chaque problème est décrit par une fonction de performance linéaire qui sera intégrée dans le système. Il en résulte un problème d'optimisation linéaire mixte en nombres entiers ("mixed integer linear programming problem") dont la solution correspond à la commande optimale par rapport au critère d'optimisation choisi [JBRS04].

Bien que nos travaux de recherche se focalisent sur la commande des RdPC sous la sémantique serveurs infinis, l'originalité de cette méthode a attiré notre attention. En effet, d'une part, la commande est maintenue constante tant qu'aucun événement permettant de modifier le comportement du système, n'est déclenché [JBRS04] (ce principe est une source d'inspiration de la commande des RdPH hybrides temporisés présentés dans le chapitre IV), d'autre part, la fonction de performance n'est pas restreinte à l'atteinte du marquage désiré, elle peut combiner explicitement et simultanément plusieurs objectifs de commande ce qui permet de répondre à des exigences réelles.

Passons maintenant à la présentation de quelques techniques de commande proposées dans la littérature et développées dans le cadre de la conception des commandes additives, destinées aux RdPC temporisés définis sous la sémantique serveurs infinis, pour lesquels toutes les transitions sont commandables.

II.2.3 Éléments d'état de l'art sur la commande des réseaux de Petri continus sous la sémantique serveurs infinis

La commande des RdPC sous la sémantique serveurs infinis a fait l'objet de plusieurs travaux de recherche et différentes techniques de commande ont été proposées dans la littérature. Afin de sélectionner la commande la plus adéquate, il faut donc, dans un premier temps, avoir une vision claire de l'ensemble des caractéristiques de chaque technique

de commande. Ces caractéristiques peuvent inclure le type de RdPC pris en compte, le critère d'optimisation, la robustesse face aux perturbations, la complexité de calcul ainsi que la stabilité. L'analyse de ces caractéristiques et l'identification des différents avantages et inconvénients de chaque commande par rapport aux autres, sont deux étapes primordiales. À travers cette section, nous présentons quelques méthodes de commande performantes proposées dans la littérature.

II.2.3.1 Commande optimale en régime permanent

D'une manière générale, le régime permanent représente un point de fonctionnement stable pour le système. L'existence et le choix de la commande optimale en régime permanent ont été traités dans [MRTRS08]. L'objectif de cette contribution est d'étudier comment calculer le flux ainsi que la commande optimale en régime permanent qui maximisent une fonction coût utilisée dans le contrôle de production, tout en respectant la loi de franchissement des transitions continues. Cette fonction, définie par (II.12), dépend du vecteur prix (P), le flux en régime permanent (X_c), le vecteur de coûts des travaux en cours (C) ainsi que la dépréciation ou l'amortissement des investissements initiaux (D) :

$$P^t \cdot X_c - C^t \cdot M_c - D^t \cdot M_{0c} \quad (\text{II.12})$$

D'après [MRTRS08], tant que toutes les transitions sont commandables et que la fonction coût est linéaire, la commande est calculée par la résolution d'un problème d'optimisation linéaire. Cette optimisation utilise des variables d'écart v telles que $\forall p_j \in {}^\circ t_i, v(j, i) > 0$, afin d'éviter l'opérateur "min" qui apparaît dans l'expression du flux et de transformer l'inégalité en égalité dans certaines contraintes. Ces variables vont servir pour calculer le flux et le marquage en régime permanent ainsi que la commande optimale correspondante [MRTRS08]. Cette commande est exprimée par : $u_{ci} = \min_{p_j \in {}^\circ t_i} v(j, i)$.

En somme, la commande optimale en régime permanent est la solution d'un problème d'optimisation linéaire simple. En revanche, le régime permanent du système n'est pas précisé au préalable. Dans ce qui suit, nous présentons des commandes pour lesquelles le régime permanent est défini a priori : le marquage désiré est atteignable à partir du marquage initial et le flux respecte les contraintes données par (II.9).

II.2.3.2 Commande "ON/OFF"

La commande "ON/OFF" est une méthode de commande proposée dans [WMJS10] pour commander les RdPC sans choix. Son objectif est d'atteindre le marquage désiré en temps minimal. Les auteurs de [WMJS10] ont présenté cette méthode pour le RdPC en

temps continu (τ) ainsi que pour le temps échantillonné (τ_k).

Tout d'abord, le vecteur de franchissements minimal $\Gamma_c = (\gamma_{cj})_{j=1,\dots,q} \in \mathbb{R}_{\geq 0}^q$, qui permet d'atteindre M_{refc} à partir de M_{0c} , est déterminé suite à la résolution du problème d'optimisation linéaire donné par l'équation (II.13) :

$$\begin{aligned} \max \{ & \mathbf{1}_{q \times 1}^t \cdot \Gamma_c \text{ tel que} \\ & M_{refc} = M_{0c} + W \cdot \Gamma_c \\ & \Gamma_c \geq \mathbf{0}_{q \times 1} \} \end{aligned} \quad (\text{II.13})$$

Un simple problème d'optimisation linéaire est ensuite résolu à chaque étape de calcul jusqu'à ce que Γ_c soit obtenu : $\int_0^{\tau^-} X_c(\tau) d\tau = \Gamma_c$ en temps continu et $\sum_{e=1}^{k-1} X_c(\tau_e) \cdot \theta = \Gamma_c$ en temps échantillonné. La complexité de calcul de cette commande est faible.

En ce qui concerne le RdPC en temps continu, la commande continue est définie pour chaque transition de la manière suivante [WMJS10] : si la valeur de γ_{cj} n'est pas encore atteinte, la transition t_j est dite complètement "ON", elle est franchie avec sa vitesse de franchissement maximale. En d'autres termes, $u_j(\tau) = 0$ et $x_j(\tau) = f_j(\tau)$. Une fois γ_{cj} atteinte, t_j est dite complètement "OFF", elle est arrêtée brusquement ce qui implique $u_j(\tau) = f_j(\tau_k)$ et $x_j(\tau) = 0$.

La même procédure est utilisée dans le cas du θ -RdPC en vérifiant, à chaque période d'échantillonnage, que le franchissement de la transition avec son flux maximal de franchissement n'engendre pas un dépassement du vecteur minimal de franchissements Γ_c . Si c'est le cas, le flux maximal de franchissement au dernier pas est limité à la valeur requise pour atteindre γ_{cj} [WMJS10].

Cette commande est très performante en termes de complexité de calcul. De plus, la convergence en temps minimal pour les RdPC sans choix a été démontrée. Néanmoins, l'efficacité de cette contribution est restreinte à une classe particulière du RdPC qui est la classe des RdPC sans choix. Elle est inappropriée pour résoudre le problème de conflit, car le franchissement de la transition la plus rapide peut réduire le franchissement de la transitions moins rapide, ce qui entraîne une augmentation considérable du temps de convergence [WMJS12]. La commande "ON/OFF" peut également amener le système dans une situation de blocage. Par conséquent, la convergence du système n'est pas garantie [WMJS12]. En outre, la configuration désirée est réduite au marquage désiré uniquement (le flux désiré n'est pas évoqué dans la détermination de la commande).

D'autres techniques de commande ont également été motivées par la commande en temps minimal, elles sont détaillées dans les paragraphes suivants.

II.2.3.3 Commande pour les trajectoires linéaires et linéaires par morceaux

Dans cette partie, nous présentons, dans un premier temps, la commande de poursuite proposée dans [XRS06, XRS08a] afin de commander les RdPC conservatifs, sans synchronisation (pour éviter la commutation entre les régions) et fortement connectés. La configuration désirée a été définie par le marquage désiré M_{refc} et la commande désirée U_{refc} qui permet de maintenir M_{cref} en régime permanent. Le défi principal de cette commande est de développer une loi de commande qui respecte les contraintes liées aux flux de franchissement des transitions de sorte que la convergence globale soit garantie. Pour ce faire, le marquage désiré a été remplacé par une trajectoire désirée mixte $M_r(\tau)$ définie par une étape suivie d'une rampe exprimée par (II.14) :

$$M_r(\tau) = \begin{cases} M_{0r} + \frac{M_{refc} - M_{0r}}{h} \cdot \tau, & \tau \in [0, h] \\ M_{refc}, & \tau \in [h, +\infty[\end{cases} \quad (\text{II.14})$$

avec M_{0r} la valeur initiale du marquage définie par $M_{0r} = M_{0c} + \beta \cdot (M_{refc} - M_{0c})$ et $h > 0$ le temps à partir duquel $M_r(\tau)$ correspond à M_{refc} . Ensuite, un algorithme de commande basé sur deux gains, un faible et l'autre élevé ("low and high gain algorithm"), a été proposé. En utilisant une fonction de Lyapunov, la convergence asymptotique globale du marquage a été démontrée.

Cette méthode a été étendue dans [XRS08b] afin de commander des RdPC avec synchronisation. En outre, afin de diminuer le temps de convergence, des marquages intermédiaires ont été ajoutés dans la trajectoire de poursuite. Cette notion de marquages intermédiaires a également été utilisée dans [AOJMS11]. L'objectif de cette contribution est de concevoir une loi de commande qui permette de minimiser τ_f , la durée de la trajectoire d'un RdPC entre un marquage initial et un marquage final, en utilisant des problèmes d'optimisation linéaire et bilinéaire.

Cette commande comporte deux étapes primordiales. Tout d'abord, la trajectoire entre les deux marquages est supposée être une ligne droite. Une optimisation linéaire, qui vise à minimiser le temps de convergence τ_f , est donc proposée par (II.15) où $X'_c = X_c \cdot \tau_f$

[AOJMS11] :

min $\{\tau_f$ tel que

$$M_{refc} = M_{0c} + W \cdot X'_c \quad (\text{II.15})$$

$$0 < x'_j < \lambda_j \cdot A_k(i, j) \cdot \min\{m_{0i}, m_{refi}\} \cdot \tau_f, \forall j = 1, \dots, q \text{ où } A_k(i, j) \neq 0\}$$

Si les deux marquages ne sont pas dans la même région, la ligne droite, reliant M_{0c} et M_{refc} , coupe $F \leq K$ frontières de régions. Soient $M_c^1, M_c^2, \dots, M_c^F$, les marquages qui résultent de ces intersections et soient $M_c^0 = M_{0c}$ et $M_c^{F+1} = M_{refc}$. La trajectoire est donc découpée en plusieurs segments $[M_c^\zeta, M_c^{\zeta+1}]$, $\zeta = 0, \dots, F$ pour lesquels le flux est considéré constant. L'optimisation linéaire, donnée par (II.15), est appliquée pour chaque segment de droite et la durée de la trajectoire est la somme des durées de chaque segment $\tau_f = \sum_{\zeta=1}^{F+1} \tau_\zeta$ [AOJMS11].

Une amélioration de cette commande est effectuée dans la deuxième phase afin de trouver une trajectoire plus rapide. Pour ce faire, la trajectoire est désormais linéaire par morceaux en introduisant des marquages intermédiaires [XRS08b]. Ces marquages sont situés sur les frontières des régions traversées mais ne sont pas forcément sur la ligne droite reliant le marquage initial au marquage final. Ils sont calculés, par le biais d'une optimisation bilinéaire, de telle sorte que la durée totale la trajectoire soit minimale. Finalement, l'algorithme proposé a été adapté en vue de déterminer, de manière récursive, les marquages intermédiaires dans une même région jusqu'à ce que la durée de la trajectoire soit la plus faible possible [AOJMS11].

En revanche, la commande proposée est en boucle ouverte, elle ne tient pas compte des perturbations qui peuvent affecter l'évolution du système. Afin de résoudre ce problème, une commande en boucle fermée est mise en place dans le but de suivre la trajectoire de la commande calculée en boucle ouverte et d'assurer la convergence vers le marquage désiré [AOJMS11]. La faisabilité (c-à-d l'existence d'une solution) et la convergence vers le marquage désiré ont été démontrées. L'apport de cette commande est la complexité calculatoire qui est significativement faible par rapport aux commandes basées sur un problème d'optimisation non-linéaire. En revanche, cette commande est très contraignante.

II.2.3.4 Commandes “ON/OFF+” et “B-ON/OFF”

Les commandes “ON/OFF+ (ON/OFF plus)” et “B-ON/OFF (Balanced ON/OFF)” sont deux extensions de la commande “ON/OFF”. Elles sont proposées dans [WMJS12] avec pour but de commander tout type de RdPC pour atteindre un marquage désiré.

Dans le cas de la première commande, à chaque période d'échantillonnage, le flux de franchissement des transitions en conflit est forcé pour être proportionnel au vecteur de franchissements minimal. En effet, soient t_j et $t_{j'}$ deux transitions en conflit structurel, $x_j(\tau_k)$ et $x_{j'}(\tau_k)$ sont calculés de telle sorte que $\frac{x_j(\tau_k)}{x_{j'}(\tau_k)} = \frac{\gamma_{cj}}{\gamma_{cj'}}$. La commande "ON/OFF" standard est conservée pour les transitions qui ne font pas partie du conflit. En conséquence, la procédure de détermination de la commande "ON/OFF+" et par la suite le problème d'optimisation linéaire sont identiques à ceux de la commande "ON/OFF" avec une contrainte de proportionnalité ajoutée [WMJS12]. Pour cette raison, la convergence en nombre fini de pas a également été garantie pour cette commande.

Cependant, en dépit de sa capacité à résoudre le problème de conflit, la commande "ON/OFF+" est en général une méthode sous optimale pour le temps minimal contrairement à la commande "ON/OFF". En fait, le flux de franchissement des transitions en conflit ainsi que le nombre de pas nécessaires pour converger sont imposés par les transitions les plus lentes. Par conséquent, le système peut être ralenti [WMJS12].

À cet égard, la commande "B-ON/OFF" a été introduite. Comme son nom l'indique, cette commande équilibre les vitesses des transitions lentes et rapides, qui sont en conflit avant d'appliquer la commande "ON/OFF+" [WMJS12]. À chaque période d'échantillonnage, les transitions lentes et rapides sont identifiées en se basant sur le calcul d'une estimation du nombre de pas. Ensuite, les plus rapides sont franchies (en utilisant "ON/OFF+") et les autres sont bloquées pour un certain temps en attendant que leur flux de franchissement augmentent et qu'elles deviennent rapides [WMJS12].

La commande "B-ON/OFF" présente une complexité de calcul plus élevée par rapport à "ON/OFF" et à "ON/OFF+". En revanche, elle permet de réduire le temps nécessaire pour converger vers le marquage désiré. La convergence en nombre fini de pas a également été démontrée pour cette commande [WMJS12].

II.2.3.5 Commande proportionnelle

Cette méthode de commande a été proposée dans [KALD09]. Elle consiste à concevoir une commande en boucle fermée telle que le flux forcé, dans la région R_k , est défini par (II.16) :

$$X_c(\tau_k) = F_c(\tau_k) - U_c(\tau_k) = g_k \cdot (M_c(\tau_k) - M_{refc}) + X_{refc} \quad (\text{II.16})$$

avec g_k le gain de rétroaction dans la région R_k . Elle prend en considération les contraintes sur la commande et la stabilité du système et elle vise à déterminer le gain de rétroaction

qui répond à ces exigences afin d'atteindre une configuration désirée définie par le marquage et le flux en régime permanent.

Cette commande peut être mise en œuvre quel que soit le type de RdPC. Elle prend en considération la notion de région définie par des polyèdres. De plus, il est supposé que le marquage du RdPC est soumis à des contraintes linéaires exprimées géométriquement par un polytope (un polyèdre fermé et borné) $P(Q, \mu) = \{S(\tau_k), Q \cdot (S\tau_k) \leq \mu\}$ tel que $S(\tau_k) = M_c(\tau_k) - M_{refc}$. L'intersection de ce polytope et des polyèdres, qui définissent les régions, donne l'ensemble des polytopes qui définissent la totalité des contraintes qui doivent être vérifiées par le marquage dans chaque région. D'autre part, les contraintes sur la commande, relatives à chaque région, sont également représentées par un ensemble de polyèdres. Des conditions géométriques nécessaires et suffisantes sont aussi obtenues pour déterminer un gain de rétroaction g_k vérifiant la stabilité et les contraintes requises sur le marquage et sur la commande. Ces conditions sont reformulées algébriquement sous la forme d'inégalités matricielles linéaires, dont la résolution à l'aide d'outils numériques classiques, permet de calculer le gain g_k [KALD09]. La convergence vers l'état désiré est ainsi garantie. Une fonction de Lyapunov a été mise en place afin de démontrer la stabilité du système.

La limitation principale de cette méthode réside dans l'effort de calcul. En effet, les inégalités matricielles linéaires doivent être résolues pour chaque région. Par conséquent, le nombre d'inégalités à résoudre augmente exponentiellement en fonction du nombre de régions (lié au nombre de synchronisations) [KALD09].

II.2.3.6 Commande prédictive

La commande prédictive ("Model predictive Control (MPC)") est une méthode de commande qui anticipe l'évolution d'un système linéaire en temps discret à partir de son modèle et pour un nombre fini de pas (connu sous le nom d'horizon de prédiction). Son but est de déterminer la commande qui optimise une certaine fonction de performance. Cette fonction est généralement définie par l'écart quadratique entre l'état actuel du système et l'état de référence. À chaque pas d'échantillonnage, un problème d'optimisation doit être résolu sous un certain nombre de contraintes. Une séquence optimale de commande est alors déterminée. Seule la première commande de cette séquence est appliquée au système. L'état du système est alors actualisé et le processus d'optimisation est réitéré au pas suivant en prenant comme point de départ le nouvel état du système et en considérant un nouvel horizon de prédiction (notion d'horizon de prédiction glissant) [RRTP78, BMDP02].

Cette commande présente beaucoup d'avantages à savoir [BMDP02, QB03, Ben09] :

- Elle est compatible avec les contraintes temps réel.
- Elle permet de faire face à des contraintes dures sur les entrées et les sorties du système. Toute violation des contraintes est anticipée et bloquée.
- Elle peut combiner plusieurs fonctions coût quadratiques sous diverses contraintes.
- Elle fournit une loi de commande en boucle fermée, seul le premier élément de la séquence de commandes, calculé à partir des mesures réelles, est appliqué, les autres sont basés sur une prédiction.
- Elle est capable de traiter le problème de commande en présence d'entrées multiples.
- Elle représente une commande robuste aux perturbations qui sont prises en compte lors du processus d'optimisation en ligne.

Grâce à ces caractéristiques, la commande prédictive a connu un succès important dans le monde industriel. Elle est maintenant largement utilisée pour résoudre des problèmes complexes et multivariables contraints [BMDP02]. Elle intervient dans divers domaines d'application, tels que l'automobile, la pétrochimie et les applications aérospatiales. Dans ce contexte, les auteurs de [QB03] présentent les principaux algorithmes de commandes prédictives utilisés dans l'industrie. Ils fournissent également une étude de tous les domaines dans lesquels la commande prédictive peut être considérée dans le futur. Les travaux de recherche présentés dans [MRRS00] quant à eux, sont plutôt consacrés à l'étude des aspects théoriques de la commande prédictive tels que la faisabilité et la stabilité de la boucle fermée.

Les auteurs de [GMR⁺06, MGR⁺08] ont proposé une adaptation de cette méthodologie aux systèmes dynamiques modélisés par RdPC. L'idée de base de cette commande prédictive implicite est de minimiser, à chaque période d'échantillonnage et sous un certain nombre de contraintes, un critère de performance noté J , sur un horizon de prédiction continu désigné par N_c . Ce critère de performance permet au RdPC d'évoluer d'un état initial constitué par le marquage initial associé au flux en régime libre (M_{0c}, X_{0c}) , jusqu'à une configuration désirée (ou un état d'équilibre) (M_{refc}, X_{refc}) définie par le marquage et le flux désirés.

La forme standard de la fonction coût utilisée par la commande prédictive implicite pour commander le RdPC est donnée par (II.17) [GMR⁺06, MGR⁺08] :

$$\begin{aligned}
J(\tau_{k+N_c}) &= (M_c(\tau_{k+N_c}) - M_{refc})^t \cdot Q_{M_c} \cdot (M_c(\tau_{k+N_c}) - M_{refc}) \\
&+ \sum_{e=0}^{N_c-1} \left((M_c(\tau_{k+e}) - M_{refc})^t \cdot Q_{M_{ce}} \cdot (M_c(\tau_{k+e}) - M_{refc}) \right. \\
&\quad \left. + (X_c(\tau_{k+e}) - X_{refc})^t \cdot Q_{X_{ce}} \cdot (X_c(\tau_{k+e}) - X_{refc}) \right)
\end{aligned} \tag{II.17}$$

où $(M_c(\tau_{k+N_c}) - M_{refc})^t \cdot Q_{M_c} \cdot (M_c(\tau_{k+N_c}) - M_{refc})$ représente l'erreur quadratique entre le marquage obtenu au terme de la prédiction (après N_c pas de calcul) et le marquage désiré. $\sum_{e=0}^{N_c-1} ((M_c(\tau_{k+e}) - M_{refc})^t \cdot Q_{M_{ce}} \cdot (M_c(\tau_{k+e}) - M_{refc}))$ et $\sum_{e=0}^{N_c-1} ((X_c(\tau_{k+e}) - X_{refc})^t \cdot Q_{X_{ce}} \cdot (X_c(\tau_{k+e}) - X_{refc}))$ dénotent respectivement la somme des erreurs quadratiques entre les marquages intermédiaires $M_c(\tau_{k+e})$ (c-à-d les marquages obtenus à chaque pas de prédiction $e = 0, \dots, N_c - 1$) et M_{refc} , et entre les flux intermédiaires $X_c(\tau_{k+e})$ et le flux désiré X_{refc} . Les matrices $Q_{M_{ce}} \in \mathbb{R}_{\geq 0}^{n \times n}$, $Q_{M_c} \in \mathbb{R}_{\geq 0}^{n \times n}$ et $Q_{X_{ce}} \in \mathbb{R}_{\geq 0}^{q \times q}$ sont des matrices définies positives précisant le poids de chaque terme.

L'objectif de la commande prédictive est de minimiser $J(\tau_{k+N_c})$ sous des contraintes liées au flux de franchissement des transitions qui doivent être vérifiées à chaque pas de prédiction. Le problème d'optimisation à résoudre, à chaque date τ_k , est énoncé par (II.18) :

$$\begin{aligned}
&\min_{X_c} \{J(\tau_{k+N_c}) \text{ tel que} \\
&\quad M_c(\tau_{k+e}) = M_c(\tau_{k+e-1}) + \theta \cdot X_c(\tau_{k+e-1}), \forall e = 1, \dots, N_c \\
&\quad X_c(\tau_{k+e-1}) \geq 0_{n \times 1}, \forall e = 1, \dots, N_c \\
&\quad X_c(\tau_{k+e-1}) \leq F_c(\tau_{k+e-1}), \forall e = 1, \dots, N_c\}
\end{aligned} \tag{II.18}$$

Or, $X_c(\tau_{k+j})$ dépend de $U_c(\tau_{k+j})$, le problème d'optimisation peut ainsi être exprimé en fonction de U_c . Un schéma explicatif simplifié de la commande prédictive est présenté figure II.4 :

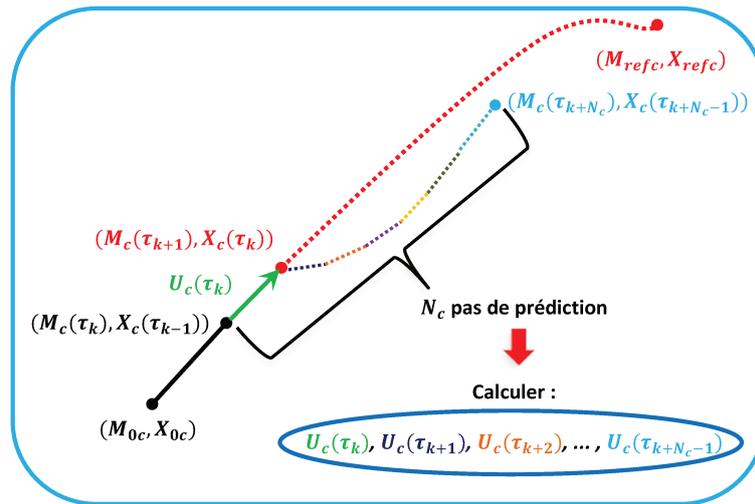


FIGURE II.4 – Principe de la commande prédictive appliquée au RdPC

Un chaque date τ_k , à partir de l'état actuel du système donné par $(M_c(\tau_k), X_c(\tau_{k-1}))$, la commande prédictive effectue une prédiction sur N_c pas de calcul afin de déterminer les actions de commande $U_c(\tau_k), U_c(\tau_{k+1}), \dots, U_c(\tau_{k+N_c})$ qui répondent au problème d'optimisation (II.18). Seule la commande $U_c(\tau_k)$ est appliquée au système au point courant. Cette commande est optimale par rapport aux contraintes imposées. Cependant, l'existence d'une solution à ce problème pour chaque pas de calcul n'est pas garantie [BMDP02]. Cette question primordiale a attiré l'attention des auteurs de [MGR⁺08]. Ils ont démontré que, grâce à la structure de ces contraintes, la faisabilité de (II.18) est toujours assurée pour tout marquage $M_c(\tau_k) \geq 0_{n \times 1}$. En revanche, la faisabilité n'assure pas la stabilité. L'étude de la stabilité n'est pas triviale, diverses approches standards ont été examinées dans [MRRS00, BMDP02, MGR⁺08, WMS13] pour garantir cette propriété. En particulier, parmi les méthodes classiques nous citons l'utilisation d'un horizon de prédiction fini ou l'introduction d'une contrainte terminale appropriée : forcer la trajectoire à rester à l'intérieur d'un convexe de l'espace d'atteignabilité ou à suivre une trajectoire linéaire. Cette dernière solution a été mise en place dans [MGR⁺08]. Le marquage $M_c(\tau_{k+N_c})$ est contraint à rester sur la droite qui joint $M_c(\tau_k)$ à M_{refc} . Analytiquement, la contrainte terminale est exprimée par (III.33) :

$$M_c(\tau_{k+N_c}) = M(\tau_k) + \alpha \cdot (M_{refc} - M_c(\tau_k)) \text{ tel que } 0 < \alpha \leq 1 \quad (\text{II.19})$$

Dans [MGR⁺08], il a été démontré que la stabilité asymptotique, à l'aide de cette contrainte, ne peut être assurée que si $M_{0c} > 0_{n \times 1}$ et que $N_c = 1$. Cette contrainte est imposée tout au long de la trajectoire ce qui implique que la trajectoire obtenue est une ligne droite.

Afin de mettre en évidence l'ajout de la contrainte terminale, nous avons utilisé l'exemple II.5 présenté dans [MGR⁺08]. Il est caractérisé par $M_{0d} = [3 \ 3 \ 1 \ 3]^t$, $\lambda = [1 \ 1 \ 1]^t$. La période d'échantillonnage choisie est $\theta = 0.3$ U.T. (elle vérifie la condition (I.17)), les matrices de pondération sont $Q_{M_{ce}} = Q_{M_c} = I_n$ et $Q_{X_{ce}} = 0.001 \cdot I_n$. La configuration désirée est définie par ($M_{refc} = [0.75 \ 6 \ 0.25 \ 8.25]$, $X_{refc} = [0.125 \ 0.125 \ 0.125]^t$). La projection de la trajectoire du marquage dans le plan (m_2, m_3) est illustrée par la figure II.6 :

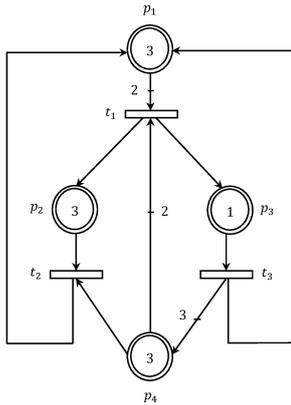


FIGURE II.5 – Exemple d'un RdPC [MGR⁺08]

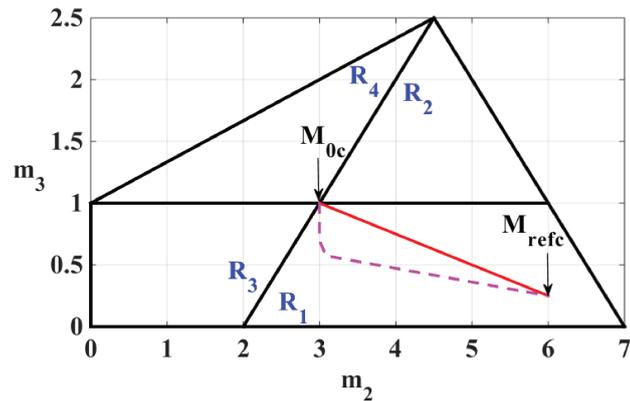


FIGURE II.6 – Trajectoire du marquage dans le plan (m_2, m_3)

Dans la figure II.6, deux trajectoires sont présentées, celle en traits discontinus est obtenue avec la commande prédictive lorsque $N_c = 1$, tandis que celle en trait continu est obtenue suite à l'introduction de la contrainte terminale.

Le gros inconvénient de la commande prédictive implicite est la complexité et l'effort de calcul requis pour résoudre le problème d'optimisation. En réalité, pour des horizons de prédiction relativement larges, ce temps de calcul peut dépasser la période d'échantillonnage du système ce qui est problématique. En conséquence, la commande prédictive implicite est peu adaptée à la résolution de ce genre de problème [BMDP02, GMR⁺06, MGR⁺08]. Afin de pallier cette difficulté, une commande prédictive explicite a été proposée dans [BMDP02]. Cette commande résout des problèmes d'optimisation multi-paramétriques ("multi-parametric quadratic programming") afin d'établir une dépendance explicite entre l'état du système et la commande appliquée. Elle est composée de deux parties, l'une hors ligne et l'autre en ligne. La partie hors ligne consiste à partitionner l'ensemble des états commandables en des polytopes et à définir explicitement, dans chaque polytope, la loi de commande associée comme une fonction continue affine par morceaux qui dépend de l'état du système [BMDP02, SJMV11]. La partie en ligne correspond à l'application des q premiers éléments de la commande adéquate implémentée en

accord avec le polytope contenant l'état actuel du système [BMDP02,GMR⁺06,MGR⁺08].

Une comparaison entre les deux commandes implicite et explicite a été menée dans [GMR⁺06,MGR⁺08]. Contrairement à la commande prédictive implicite, la stabilité de la boucle fermée est garantie avec la commande prédictive explicite sans imposer de contrainte terminale. En outre, les calculs les plus longs sont effectués hors ligne. Néanmoins, la commande prédictive explicite n'est pas applicable pour les RdPC complexes (ou de taille réelle) ou quand l'horizon de prédiction choisi est large, car la complexité de calcul hors ligne augmente considérablement et la détermination des polytopes devient impossible [BMDP02,SJMV11]. D'autre part, la partie en ligne peut rencontrer quelques difficultés liées au nombre de régions. En conséquence, dans le cas des systèmes complexes, la commande prédictive implicite est la mieux adaptée [GMR⁺06,MGR⁺08]. De ce fait, dans le cadre de cette étude, la commande prédictive implicite est retenue.

Dans la littérature récente, plusieurs travaux de recherche ont utilisé la commande prédictive pour commander efficacement les RdPC. En particulier, dans [JB10], les auteurs ont considéré le cas particulier des réseaux de transport. Afin d'éviter le problème de l'explosion combinatoire du nombre d'états, ils ont proposé un modèle macroscopique basé sur le RdPC pour concevoir les lois de commande. Une commande prédictive en boucle fermée a ensuite été mise en place afin d'améliorer le comportement des systèmes de circulation en déterminant la séquence de commutations des feux de circulation.

Dans [WMJS14], la commande prédictive a été combinée avec la commande "ON/OFF". Certes la commande "ON/OFF" ainsi que ses extensions sont efficaces et simples à implémenter, mais la résolution des conflits et la détermination de la commande se basent essentiellement sur l'état courant sans avoir une vision claire de ce qui se passe dans le futur [WMJS14]. Les auteurs de [WMJS14] ont proposé de combiner la commande "ON/OFF" et la commande prédictive afin de mettre en place une commande nommée "MPC-ON/OFF". Étant donné que la commande prédictive ne permet pas d'obtenir un temps minimal, elle a été appliquée uniquement aux transitions en conflit. Les transitions persistantes sont franchies avec leur flux maximal de franchissement jusqu'à atteindre le vecteur de franchissements minimal, ce qui revient à appliquer la commande "ON/OFF". La fonction coût à minimiser, à chaque pas de calcul, est donnée par (II.20). Toutes les contraintes sur le flux à chaque pas de prédiction sont également prises en

compte [WMJS14] :

$$\begin{aligned}
J(\tau_{k+N_c}) &= (M_c(\tau_{k+N_c}) - M_{refc})^t \cdot Q_{M_c} \cdot (M_c(\tau_{k+N_c}) - M_{refc}) \\
&+ \sum_{e=0}^{N_c-1} \left((M_c(\tau_{k+e}) - M_{refc})^t \cdot Q_{M_{ce}} \cdot (M_c(\tau_{k+e}) - M_{refc}) \right) \\
&- (X_c(\tau_{k+e}))^t \cdot Q_{X_{ce}} \cdot (X_c(\tau_{k+e}))
\end{aligned} \tag{II.20}$$

La stabilité asymptotique de la commande ‘‘MPC-ON/OFF’’ a été démontrée, le temps de convergence a été réduit pour un horizon de prédiction large, mais la complexité de calcul de cette méthode est élevée par rapport aux autres extensions [WMJS14].

La commande prédictive a été employée non seulement en tant que commande centralisée mais aussi comme une stratégie de commande distribuée. Elle permet de faire face soit aux systèmes complexes et de grande taille, soit aux systèmes qui sont fonctionnellement distribués. Cette idée a été évoquée dans [WMS13]. Les auteurs ont proposé tout d’abord une commande prédictive centralisée qui assure la stabilité asymptotique du système en boucle fermée. Puis, ils se sont concentrés sur les sous-systèmes afin de les faire évoluer vers leur marquage désiré. Ils ont ensuite appliqué la commande prédictive développée comme une commande locale pour chaque sous-système. L’algorithme proposé garantit la convergence de chaque sous-système en un temps fini.

La section suivante présente un état de l’art sur les méthodes de commande destinées aux RdPH et elle insiste sur l’utilisation de la commande prédictive à cet égard.

II.3 Commande des réseaux de Petri hybrides

Cette partie concerne la présentation des RdPHE commandés qui rentrent dans le cadre de nos travaux de recherche. La synthèse des lois de commande pour tout type de RdPH sera brièvement évoquée, notamment la commande prédictive.

II.3.1 Commande des réseaux de Petri hybrides élémentaires

D’une manière générale, un RdPH est dit en régime forcé s’il est soumis à une action de commande externe appliquée aux transitions commandables (continues et discrètes) afin de modifier l’évolution du système. Dans le cadre de nos travaux de recherche, le but est de commander les RdPHE de façon optimale par rapport à certains critères afin d’atteindre un marquage désiré. Nous supposons dans cette partie que certaines transitions peuvent être non-commandables. Soit $T^{comm} \subseteq T$ tel que $T^{comm} = T_c^{comm} \cup T_d^{comm}$, l’ensemble des

transitions continues et discrètes commandables. Par conséquent, le θ -RdPHE commandé est désormais représenté par l'ensemble $\langle \mathcal{N}_h, M_0, D_{min}, \lambda, \theta, T^{comm} \rangle$.

Dans un premier temps, seules les transitions discrètes immédiates sont prises en compte. De ce fait, dès qu'une transition discrète t_j est validée par le marquage discret courant, la commande appliquée valide ou non son franchissement. Si $t_j \in T_d^{comm}$ est temporisée, le franchissement ne peut avoir lieu avant l'écoulement d'une durée $d_{min j}$. Le franchissement des transitions discrètes validées non-commandables ($T_d \setminus T_d^{comm}$) n'est pas prévisible et donc ne peut être ni forcé ni évité. Il se produit d'une façon intempestive et dépend de conditions internes ou externes inconnues. Les franchissements des transitions discrètes non-commandables seront donc considérés comme des perturbations. Par ailleurs, il est supposé que les actions de commande discrètes commandables auront lieu aux dates τ_k et seront représentées par un vecteur de franchissements noté $X_d(\tau_k)$.

En ce qui concerne les transitions continues commandables, le flux ne peut qu'être ralenti ou stoppé par le vecteur de commandes $U_c \in \mathbb{R}_{\geq 0}^{q_c}$. En revanche, les transitions continues non-commandables ($T_c \setminus T_c^{comm}$) seront franchies avec leur flux maximal de franchissement qui correspond au flux en régime libre $F_c(\tau_k)$. En conséquence, le flux en régime forcé et le marquage continu d'un θ -RdPHE sont désormais exprimés selon (II.21) et (II.22) :

$$\begin{aligned} X_c(\tau_{k+}) &= F_c(\tau_{k+}) - U_c(\tau_{k+}) \\ &= \Lambda \cdot (A_k^c \cdot M(\tau_k) + A_k^{cd} \cdot W^{dd} \cdot X_d(\tau_k)) - U_c(\tau_{k+}) \end{aligned} \quad (\text{II.21})$$

$$\begin{aligned} M_c(\tau_{k+1}) &= M_c(\tau_k) + \theta \cdot W^{cc} \cdot X_c(\tau_{k+}) \\ &= G_k^c \cdot M_c(\tau_k) + G_k^d \cdot (M_d(\tau_k) + W^{dd} \cdot X_d(\tau_k)) - \theta \cdot W^{cc} \cdot U_c(\tau_{k+}) \end{aligned} \quad (\text{II.22})$$

Soit $V_c^{ncomm} \in \{0, 1\}^{q_c}$ un vecteur, défini par l'équation (II.23), qui détermine les transitions continues non-commandables :

$$\forall j = q_d + 1, \dots, q, V_c^{ncomm}(j) = \begin{cases} 1, & \text{si } t_j \in T_c \setminus T_c^{comm} \\ 0, & \text{sinon} \end{cases} \quad (\text{II.23})$$

La commande continue $U_c(\tau_{k+})$ doit respecter les trois conditions données par l'équa-

tion (II.24) :

$$\begin{aligned}
\mathbf{a}/ U_c(\tau_{k+}) &\geq 0_{q_c \times 1} \\
\mathbf{b}/ \text{diag}(V_c^{ncomm}) \cdot U_c(\tau_{k+}) &= 0_{q_c \times 1} \\
\mathbf{c}/ U_c(\tau_{k+}) &\leq \Lambda \cdot A_k^c \cdot M(\tau_k) + \Lambda \cdot A_k^{cd} \cdot W^{dd} \cdot X_d(\tau_k)
\end{aligned}
\tag{II.24}$$

À chaque date τ_k , les deux premières conditions (a) et (b) imposent respectivement que la commande continue soit positive pour les transitions continues commandables (car l'action de commande étudiée ne peut que réduire le flux de franchissement de ces transitions) et qu'elle soit nulle pour les transitions continues non-commandables. La condition (c) impose un flux forcé positif (c-à-d la commande continue doit être inférieure ou égale au flux continu en régime libre).

II.3.2 Éléments d'état de l'art sur la commande des réseaux de Petri hybrides

Dans cette sous-section, nous présentons certaines méthodes de commande des RdPH proposées dans la littérature. Généralement, ces méthodes utilisent la sémantique serveurs finis afin d'approcher le flux au travers d'une transition continue temporisée.

Dans [BGM00], l'objectif est de commander de façon optimale les systèmes manufacturiers modélisés par des RdPH temporisés pour lesquels les transitions continues sont définies sous la sémantique serveurs finis et les transitions discrètes peuvent être immédiates, déterministes ou stochastiques. La commande proposée détermine le mode de fonctionnement optimal conformément à un objectif donné (par exemple maximiser l'utilisation des machines, maximiser la production du système et répartir la charge). Sachant que chaque mode de fonctionnement possible peut être représenté par une vitesse de franchissement, l'ensemble des vitesses de franchissement admissibles a été caractérisé, dans chaque IB-state, par des inégalités linéaires. Si le système est sans choix, le calcul des vitesses maximales revient à maximiser indépendamment chaque composante par la résolution d'un problème d'optimisation linéaire. Dans le cas contraire, le problème de conflit entre les transitions doit être résolu. Les auteurs ont proposé des problèmes d'optimisation linéaires avec des fonctions objectifs locales et globales.

Certes, plusieurs méthodes de commande des RdPH existent dans la littérature, mais nous nous focalisons sur la commande prédictive.

Dans [JCBM13], les auteurs ont proposé une commande prédictive événementielle destinée aux RdPH temporisés. Cette commande est une amélioration de la commande événementielle ("event-driven") proposée dans [JBRS04] pour commander les RdPC défi-

nis sous la sémantique serveurs finis. Pour le RdPH temporisé, ils ont conservé cette même sémantique pour le franchissement des transitions continues tandis que pour les transitions discrètes, ils ont supposé que le franchissement était déterministe. Comme nous l'avons expliqué dans la sous-section II.2.2, la méthode consiste à éviter la discrétisation du temps et propose une évolution qui dépend de l'occurrence d'un événement (une place vidée ou le franchissement d'une transition discrète), ce qui implique que le temps séparant deux événements successifs est variable.

L'action de commande $U = \begin{bmatrix} U_c^t & U_d^t \end{bmatrix}^t$, dans chaque IB-state, est composée de deux parties, une partie $U_c \in \mathbb{R}_{\geq 0}^{q_c}$ qui correspond à l'action de commande appliquée aux transitions continues et une autre partie booléenne $U_d \in \{0, 1\}^{q_d}$ qui active ou non le franchissement des transitions discrètes.

Étant donné que le flux de franchissement est constant entre deux événements et que la temporisation des transitions discrètes validées se décrémente après chaque occurrence d'événement, la dynamique du RdPH est reformulée en tenant compte de l'évolution du marquage dans chaque IB-state. Ils ont proposé un ensemble de problèmes de commande optimale en boucle ouverte sur un horizon fini (qui correspond à un nombre fini d'événements). Chaque problème est exprimé, à un instant donné, à l'aide d'un problème d'optimisation linéaire mixte en nombres entiers. La fonction coût à optimiser ainsi que les contraintes sont définies par des équations dynamiques par rapport à l'objectif de commande [JCBM13].

Finalement, afin de faire face aux perturbations qui peuvent survenir, une commande événementielle en boucle fermée, qui utilise le principe de la commande prédictive, a également été proposée. En effet, cette commande utilise une prédiction sur un horizon fini d'événements. Elle résout un problème d'optimisation linéaire mixte en nombres entiers, puis elle applique la première commande calculée. L'état résultant est déterminé et actualisé. La procédure est répétée jusqu'à atteindre l'objectif.

Si l'objectif de commande est de minimiser le temps de convergence vers un marquage désiré, cet objectif est atteint en temps fini [JCBM13].

L'efficacité de cette commande a été mise en évidence sur des RdPDE modélisant une machine multi-classes et sur un réseau de production. Le seul inconvénient de cette commande est la complexité de calcul qui augmente exponentiellement en fonction de la taille du RdPH et du nombre d'événements.

Dans le cadre de l'utilisation de la commande prédictive, une méthode de commande des systèmes autoroutiers a été proposée dans [FIMU14]. Les auteurs ont conçu une technique de commande en ligne inspirée de la commande prédictive afin de réguler la circulation sur les autoroutes en maximisant le flux de circulation dans les voies congestionnées.

Une modélisation de la circulation autoroutière par les RdPH a été mise en œuvre. Le but recherché concerne l'optimisation de la vitesse des véhicules des voies critiques lorsque le nombre de véhicules dépasse la densité critique, ceci revient à optimiser la vitesse instantanée de franchissement des transitions en amont des places modélisant les voies critiques. Sachant que cette méthode utilise la notion de macro-étape qui correspond à l'occurrence d'un événement imprévu (un accident par exemple), la vitesse maximale de franchissement est considérée constante entre deux macro-événements successifs et la prédiction du comportement du système est effectuée sur N macro-étapes afin de déterminer la vitesse de franchissement qui minimise la fonction de performance souhaitée.

II.4 Synthèse de l'analyse bibliographique et choix de la commande

Une synthèse de l'analyse bibliographique permet de choisir la commande la mieux adéquate à la problématique étudiée : déterminer une commande qui peut commander efficacement tout type de RdP et qui présente une complexité de calcul réduite. Le tableau II.1 présente une comparaison entre les différentes techniques de commande continues.

TABLE II.1 – Comparaison entre les différentes techniques de commande [SJM11]

Technique de commande	type de RdPC	critère d'optimisation	complexité	stabilité
"ON/OFF"	sans choix	temps minimal	faible	oui
Commande pour les trajectoires linéaires	tout	heuristique pour temps minimal	élevée	oui
"ON/OFF+"	tout	heuristique pour temps minimal	moyenne	oui
"B-ON/OFF"	tout	heuristique pour temps minimal	moyenne	oui
Commande proportionnelle	tout	pas de critère	élevée	oui
Commande prédictive	tout	fonction quadratique	élevée	sous condition
"MPC-ON/OFF"	tout	heuristique pour temps minimal	moyenne	oui

En ce qui concerne le RdPC, la commande proportionnelle ne permet que d'atteindre un régime permanent. De plus, sa complexité de calcul est élevée due au nombre d'inégalités matricielles linéaires à résoudre qui augmente exponentiellement en fonction du nombre de configurations. Même si la commande "ON/OFF" présente une complexité de calcul faible, elle assure la convergence et garantit la stabilité, par contre elle est uniquement applicable aux RdPC sans choix, ce qui limite son champ d'application.

Les commandes "B-ON/OFF", "ON/OFF+", MPC-ON/OFF ainsi que la commande pour

les trajectoires linéaires (et linéaires par morceaux) peuvent être utilisées pour tout type de RdPC et leur complexité de calcul est faible par rapport aux autres commandes telles que la commande prédictive. Néanmoins, elles ne permettent que de résoudre la minimisation du temps de convergence et elles imposent souvent des trajectoires linéaires (ou linéaires par morceaux). Or, d'autres problèmes doivent être pris en compte par la commande. En particulier, assurer la convergence des RdPC qui possèdent un comportement particulier comme le "hill-climbing" et limiter la forte variation du flux, ce qui se traduit dans la pratique par une sollicitation excessive des actionneurs du système commandé. Ce problème représente l'une des causes du vieillissement précoce des composants.

La commande prédictive calcule une commande optimale, sous certaines conditions, par rapport à une fonction coût. Elle peut être appliquée à tout type de RdPC et elle assure la convergence vers un régime permanent. La stabilité peut également être garantie en imposant une contrainte terminale. Cette commande prédictive peut engendrer une forte variation du flux, mais contrairement aux autres méthodes, un critère quadratique qui tient compte de la variation du flux peut facilement être inclus dans la fonction coût à minimiser. De plus, l'un des points positifs de cette commande est la prédiction à long terme de l'évolution du système et l'adaptation de la commande au comportement du système. Par conséquent, la commande prédictive est la commande la plus appropriée pour répondre aux différentes exigences. En conclusion, nous nous intéresserons à cette commande dans nos travaux de recherche. D'un autre côté, la complexité de cette commande devient importante, d'une part, dans le cas de systèmes complexes et d'autre part si l'horizon de prédiction est important. Afin de profiter des avantages de cette commande et de réduire la complexité de calcul, il s'avère important de proposer des simplifications sans dégrader les performances de la commande. Ces simplifications seront abordées dans le chapitre III.

Parmi les commandes présentées précédemment, seule la commande prédictive a été étendue afin de commander les RdPD dans [LL15] et les RdPH dans [JCBM13,FIMU14]. Dans le cadre de notre étude, la commande prédictive continue proposée sera étendue pour commander les RdPD et les RdPH. En ce qui concerne le RdPD, dans le but de limiter l'effort de calcul nécessaire pour explorer l'espace d'atteignabilité et de s'affranchir du problème de l'explosion combinatoire du nombre d'états, la commande prédictive développée pour commander le RdPC sera utilisée pour proposer une commande prédictive discrète qui permet de générer la séquence de commandes discrètes du RdPD correspondant.

Finalement, dans le cas des RdPH, les transitions continues sont à vitesses variables et les transitions discrètes sont immédiates et/ou temporisées. Notre objectif est semblable à celui de [JCBM13], à savoir atteindre un marquage désiré. De plus, nous utilisons la même commande, nous supposons aussi qu'en l'absence d'un événement (franchissement d'une transition discrète), la commande est constante (pendant la prédiction) et que les transitions discrètes ne sont franchies qu'au début de chaque pas de calcul. Or, contrairement à [JCBM13], la période d'échantillonnage est constante et les transitions discrètes ne sont pas commandées par la commande prédictive continue. La limitation de la sollicitation des actionneurs sera également considérée dans le problème d'optimisation. La commande prédictive hybride proposée sera détaillée dans le chapitre IV.

Conclusion

L'objectif de ce chapitre est de présenter les commandes appliquées aux trois types de RdP et de choisir la commande la mieux adaptée à la problématique étudiée. Concernant la commande des RdPD. Il existe deux types de commande, à savoir la commande supervisée et la commande d'ordonnancement. Dans notre étude, seul le deuxième type de commande est concerné. Généralement, les différentes méthodes proposées dans la littérature sont basées sur une exploration partielle du graphe d'atteignabilité. En conséquence, leur efficacité est limitée dû au problème de l'explosion combinatoire du nombre d'états.

Dans la deuxième partie, nous avons défini le principe et l'objectif de la commande continue qui consiste à atteindre un point d'équilibre. Nous avons étudié la notion de commandabilité dans le cas des transitions commandables et non-commandables pour lequel la complexité de commande est accrue. Un des objectifs étant de diminuer la complexité, et comme la sémantique serveurs infinis présente une meilleure approximation du RdPD, seules les techniques de commande des RdPC à vitesses variables seront considérées et les transitions seront toutes supposées commandables. Parmi ces commandes, les avantages de la commande prédictive ont été rappelés.

La troisième partie de ce chapitre a concerné les commandes des RdPH et la synthèse de l'analyse bibliographique. Généralement, les commandes de RdPH qui existent dans la littérature sont événementielles, car les transitions continues sont à vitesses constantes. Cependant, dans nos travaux, nous considérons les transitions continues à vitesses variables.

La commande prédictive a été comparée aux différentes méthodes de commande des RdPC présentées. Notamment, grâce à son critère de performance quadratique, cette commande

est la mieux adaptée pour reformuler le problème de limitation de la sollicitation excessive des actionneurs continus. Grâce à la prédiction, elle permet de commander des RdPC pour lesquels le marquage désiré ne peut être atteint directement par une décroissance monotone du critère. La résolution de ce problème nécessite un horizon de prédiction important, ce qui peut augmenter considérablement la complexité de cette commande. Quelques modifications seront ainsi apportées à cette commande dans le chapitre suivant. La commande prédictive peut être étendue pour commander les RdPD et les RdPH. Afin d'éviter l'exploration de l'espace d'atteignabilité, une application de la commande prédictive développée dans le cas des RdPC sera présentée dans le but de déterminer la séquence de franchissements du RdPD correspondant. Une commande prédictive hybride sera également présentée dans le dernier chapitre, elle traite le problème de temporisation des transitions discrètes et limite la sollicitation excessive des actionneurs continus. Elle combine deux types de commandes prédictives : celle développée dans le chapitre suivant ainsi que celle proposée par [LL15].

Sommaire

Introduction	76
III.1 Commande prédictive constante sur l'horizon de prédiction .	76
III.1.1 Principe de la commande prédictive	76
III.1.2 Problème du "hill-climbing"	80
III.1.3 Étude de la complexité de calcul	84
III.1.4 Détermination de l'horizon de prédiction maximal	87
III.1.5 Exemple illustratif 1	87
III.2 Limitation de la variation du flux	93
III.2.1 Principe de la commande prédictive avec limitation de la varia- tion du flux	93
III.2.2 Exemple illustratif 2	95
III.3 Étude de stabilité	98
III.3.1 Modification de la fonction coût	98
III.3.2 Étude de la stabilité asymptotique	99
III.3.3 Exemple illustratif 3	102
III.4 Application de la commande prédictive aux réseaux de Petri discrets T-temporisés	105
III.4.1 Principe de la commande prédictive du RdPD T-temporisé . .	105
III.4.2 Exemple illustratif 4	109
Conclusion	112

Introduction

Durant ces dernières années, grâce à sa robustesse, la commande prédictive est devenue une stratégie attractive, répandue et très utilisée pour commander et assurer la convergence de nombreuses sous-catégories de RdP. Dans le chapitre précédent, nous venons de présenter l'idée générale de cette commande et d'énumérer ses différents avantages. Néanmoins, elle présente trois problèmes. Le premier est la complexité de calcul qui est directement proportionnelle au nombre de pas de prédiction et à la taille du système modélisé [JVMS13]. En vue d'éviter ce problème, nous pouvons nous contenter d'un nombre de pas réduit ($N_c = 1$ ou $N_c = 2$). Cependant, à ce stade, le deuxième problème apparaît. Pour certains types de réseaux, il est indispensable de prédire sur un horizon large afin de bien prévoir le comportement du système. En conclusion, il faut simplifier la commande prédictive et faire un compromis dans le choix de N_c . En outre, la troisième difficulté rencontrée réside dans les fortes variations du flux engendrées par la commande qui provoque une sollicitation excessive des actionneurs.

Ce chapitre tente de remédier à ces inconvénients. Il propose, en premier lieu, une simplification de la commande prédictive en maintenant la commande constante durant la prédiction. Par ailleurs, la taille de l'horizon de prédiction est adaptée en fonction des variations de J et des changements de région. Nous montrons que ces simplifications limitent la complexité de calcul par rapport à la commande prédictive usuelle. De plus, afin de limiter les variations de flux, une modification du critère de performance est effectuée. Une étude de stabilité est ensuite menée en vue de garantir la stabilité du système autour de la configuration souhaitée. Finalement, la section III.4 concerne une application de la commande prédictive pour commander les RdPD T-temporisés.

III.1 Commande prédictive constante sur l'horizon de prédiction

III.1.1 Principe de la commande prédictive

Choisir un horizon de prédiction large est une arme à double tranchant ; elle permet, d'une part, d'améliorer les résultats de la commande, mais d'autre part, elle augmente la complexité calculatoire de l'algorithme de commande. Pour cette raison, une méthode prédictive constante sur l'horizon de prédiction est proposée. Elle représente une simplification de la commande prédictive qui vise à limiter la complexité et le temps de calcul et à éviter le problème de l'explosion combinatoire lié au nombre de configurations à consi-

dérer et aux contraintes à prendre en compte. Elle limite aussi la dimension de l'espace de recherche au cours de la prédiction. Son idée de base est de supposer qu'au cours de la prédiction la commande reste constante et qu'il n'y aura pas de changement de région avant le dernier pas de prédiction. Tous les marquages intermédiaires, $M_c(\tau_{k+e})$, $e = 1, \dots, N_c - 1$, doivent rester dans la région R_k du marquage $M_c(\tau_k)$. Dans le cas contraire, N_c diminue afin de garantir l'appartenance de tous les marquages $M_c(\tau_{k+e})$, $e = 1, \dots, N_c - 1$, à R_k . En outre, le critère de performance utilisé par cette commande est une forme simplifiée du critère (II.17). L'expression de $J(\tau_{k+N_c})$ est donnée par (III.1) :

$$J(\tau_{k+N_c}) = J_M(\tau_{k+N_c}) + J_X(\tau_{k+N_c-1}) \quad (\text{III.1})$$

où $J_M(\tau_{k+N_c})$ et $J_X(\tau_{k+N_c-1})$ sont respectivement définis par (III.2) et (III.3) :

$$J_M(\tau_{k+N_c}) = (M_c(\tau_{k+N_c}) - M_{refc})^t \cdot Q_{M_c} \cdot (M_c(\tau_{k+N_c}) - M_{refc}) \quad (\text{III.2})$$

$$J_X(\tau_{k+N_c-1}) = (X_c(\tau_{k+N_c-1}) - X_{refc})^t \cdot Q_{X_c} \cdot (X_c(\tau_{k+N_c-1}) - X_{refc}) \quad (\text{III.3})$$

Enfin, les deux matrices Q_{M_c} et Q_{X_c} sont choisies diagonales telles que $Q_{M_c} = r_{M_c} \cdot I_n$, $Q_{X_c} = r_{X_c} \cdot I_q$, avec $r_{M_c} = 1$ et $r_{X_c} \in \mathbb{R}_{\geq 0}$.

Remarque 1. *La commande est supposée être constante durant la prédiction. De ce fait, nous notons qu'il est inutile d'inclure les marquages et les flux intermédiaires dans le critère puisqu'ils résultent tous de l'application de la même commande U_c . Par conséquent, la prise en compte de ces points intermédiaires ne change pas de manière significative le calcul de U_c et les résultats obtenus dans les deux cas sont similaires.*

Il est important de souligner que même si J ne tient pas compte des marquages et des flux intermédiaires, ces derniers doivent satisfaire toutes les contraintes (en terme d'invariance de région et de positivité du flux de franchissement). Par conséquent, le problème d'optimisation à résoudre, à chaque date τ_k , est donné par (III.4) :

$$\begin{aligned} & \min_{U_c} \{J(\tau_{k+N_c}) \text{ tel que} \\ & M_c(\tau_{k+e}) = M_c(\tau_{k+e-1}) + \theta \cdot X_c(\tau_{k+e-1}), \forall e = 1, \dots, N_c \\ & M_c(\tau_{k+e}) \in R_k, \forall e = 1, \dots, N_c - 1 \\ & U_c(\tau_{k+e}) = U_c(\tau_{k+e-1}), \forall e = 1, \dots, N_c \\ & X_c(\tau_{k+e-1}) \geq 0_{n \times 1}, \forall e = 1, \dots, N_c \\ & X_c(\tau_{k+e-1}) \leq F_c(\tau_{k+e-1}), \forall e = 1, \dots, N_c \} \end{aligned} \quad (\text{III.4})$$

Le problème décrit par (III.4) consiste à déterminer l'action de commande U_c qui minimise J sous un ensemble de contraintes, comme dans le cas du problème (II.18) défini dans le chapitre précédent. Les nouveautés apportées par (III.4) par rapport à (II.18), sont : la commande qui est supposée être constante sur l'horizon de prédiction, les marquages et les flux intermédiaires qui ne sont pas pris en compte dans le critère de performance et les marquages qui ne changent pas de région avant le dernier pas de prédiction.

Un schéma explicatif de la commande prédictive constante est présenté figure III.1 :

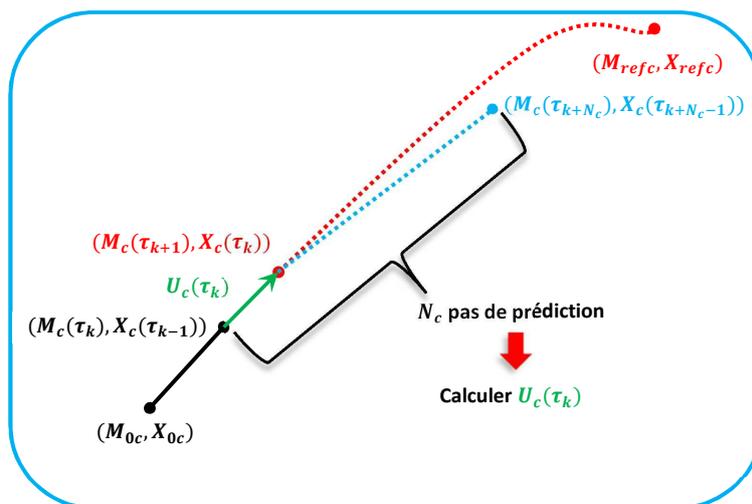


FIGURE III.1 – Principe de la commande prédictive constante appliquée au RdPC

À chaque date τ_k et à partir de l'état actuel du système donné par $(M_c(\tau_k), X_c(\tau_{k-1}))$, la commande prédictive constante effectue une prédiction sur N_c pas de calcul afin de déterminer l'action de commande constante $U_c(\tau_k)$ qui répond au problème d'optimisation (III.4), $U_c(\tau_k)$ est ensuite appliquée à l'état courant du système.

Puisque J est une fonction quadratique, le problème d'optimisation donné par (III.4) peut être réécrit sous la forme d'un problème d'optimisation quadratique. La résolution de type de problème est effectuée à l'aide de la fonction "quadprog" de Matlab.

Proposition III.1.1. *Soit $\langle \mathcal{N}_c, M_0, \lambda, \theta, T_c^{comm} \rangle$ un θ -RdPC tel que θ vérifie la condition (I.35) et que $T_c^{comm} = T$. Soit $N_c > 0$ l'horizon de prédiction continu. Supposons que $M_{refc} > 0_{n \times 1}$ est atteignable à partir de $M_{0c} > 0_{n \times 1}$, que X_{refc} respecte les conditions (II.9) pour que la configuration désirée (M_{refc}, X_{refc}) soit un état d'équilibre et que, durant la prédiction, les marquages $M(\tau_{k+e})$, $e = 1 \dots, N_c - 1$ restent dans la même région R_k . Alors, la commande continue $U_c(\tau_k)$ solution du problème d'optimisation (III.4), est aussi*

la solution du problème d'optimisation quadratique (III.5) sous les contraintes (III.7) :

$$U_c(\tau_k) = \arg \min_{U_c} \{J_{U_c}(\tau_{k+N_c})\} \quad (III.5)$$

tel que le critère $J_{U_c}(\tau_{k+N_c})$ s'écrit sous la forme (III.6) :

$$J_{U_c}(\tau_{k+N_c}) = \frac{1}{2} \cdot U_c(\tau_k)^t \cdot \Omega \cdot U_c(\tau_k) + \rho(\tau_k)^t \cdot U_c(\tau_k) \quad (III.6)$$

et :

$$\forall e = 1, \dots, N_c \quad \begin{array}{l} \mathbf{a}/ S_{e-1} \cdot U_c(\tau_k) \geq 0_{q \times 1} \\ \mathbf{b}/ S_{e-1} \cdot U_c(\tau_k) \leq \Lambda \cdot A_k \cdot B_k^{e-1} \cdot M_c(\tau_k) \end{array} \quad (III.7)$$

Les termes Ω et $\rho(\tau_k)$ sont respectivement définis par (III.8) et (III.9) :

$$\Omega = \theta^2 \cdot W^t \cdot \Sigma_{N_c}^t \cdot Q_{M_c} \cdot \Sigma_{N_c} \cdot W + S_{N_c-1}^t \cdot Q_{X_c} \cdot S_{N_c-1} \quad (III.8)$$

$$\rho(\tau_k) = \theta \cdot W^t \cdot \Sigma_{N_c}^t \cdot Q_{M_c} \cdot (M_{refc} - B_k^{N_c} \cdot M_c(\tau_k)) + S_{N_c-1}^t \cdot Q_{X_c} \cdot (X_{refc} - \Lambda \cdot A_k \cdot B_k^{N_c-1} \cdot M_c(\tau_k)) \quad (III.9)$$

B_k est défini par (I.20), Σ_e et S_e sont respectivement donnés par (III.10) et (III.11) :

$$\Sigma_0 = 0, \Sigma_e = \sum_{j=0}^{e-1} (B_k)^j, e = 1, \dots, N_c \quad (III.10)$$

$$S_e = \theta \cdot \Lambda \cdot A_k \cdot \Sigma_e \cdot W + I_q, e = 0, \dots, N_c \quad (III.11)$$

Démonstration. Sous l'hypothèse que dans l'intervalle de temps $[\tau_k, \tau_{k+N_c}[$ la commande est constante et que les marquages intermédiaires restent dans la région R_k , nous pouvons en déduire que ces marquages ainsi que les flux intermédiaires, exprimés respectivement par (III.12) et (III.13), sont obtenus de manière itérative à partir du système d'équations (II.7) :

$$M_c(\tau_{k+e}) = B_k^e \cdot M_c(\tau_k) - \theta \cdot \Sigma_e \cdot W \cdot U_c(\tau_k) \quad (III.12)$$

$$X_c(\tau_{k+e-1}) = \Lambda \cdot A_k \cdot B_k^{e-1} \cdot M_c(\tau_k) - S_{e-1} \cdot U_c(\tau_k) \quad (III.13)$$

Ces deux expressions sont utilisées pour réécrire les deux critères $J_{M_c}(\tau_{k+N_c})$ et $J_{X_c}(\tau_{k+N_c})$. Le critère J est utilisé dans un algorithme d'optimisation, tous les termes qui ne dépendent pas de $U_c(\tau_k)$ sont éliminés. Par conséquent, une forme réduite est obtenue pour chacun de ces critères. Elles sont respectivement définies par $J_{M_{U_c}}(\tau_{k+N_c})$ et $J_{X_{U_c}}(\tau_{k+N_c})$:

$$\begin{aligned} J_{M_{U_c}}(\tau_{k+N_c}) &= \theta^2 \cdot U_c(\tau_k)^t \cdot W^t \cdot \Sigma_{N_c}^t \cdot Q_{M_c} \cdot \Sigma_{N_c} \cdot W \cdot U_c(\tau_k) \\ &+ 2 \cdot \theta \cdot (M_{refc} - B_k^{N_c} \cdot M_c(\tau_k))^t \cdot Q_{M_c} \cdot \Sigma_{N_c} \cdot W \cdot U_c(\tau_k) \end{aligned} \quad (III.14)$$

$$\begin{aligned}
J_{X_{U_c}}(\tau_{k+N_c-1}) &= U_c(\tau_k)^t \cdot S_{N_c-1}^t \cdot Q_{X_c} \cdot S_{N_c-1} \cdot U_c(\tau_k) \\
&+ 2 \cdot (X_{ref} - \Lambda \cdot A_k \cdot B_k^{N_c-1} \cdot M_c(\tau_k))^t \cdot Q_{X_c} \cdot S_{N_c-1} \cdot U_c(\tau_k)
\end{aligned} \tag{III.15}$$

En conséquence, $J(\tau_{k+N_c})$ est aussi réduit à $J_{U_c}(\tau_{k+N_c})$ qui ne dépend que de $U_c(\tau_k)$.

Ce critère réduit est défini par (III.16) :

$$J_{U_c}(\tau_{k+N_c}) = \frac{1}{2} \cdot (J_{M_{U_c}}(\tau_{k+N_c}) + J_{X_{U_c}}(\tau_{k+N_c-1})) \tag{III.16}$$

En remplaçant $J_{M_{U_c}}(\tau_{k+N_c})$ et $J_{X_{U_c}}(\tau_{k+N_c-1})$ par leur expression, $J_{U_c}(\tau_{k+N_c})$ est réécrit sous la forme (III.6) où Ω et $\rho(\tau_k)$ sont respectivement définis selon (III.8) et (III.9). Sachant que, $\forall e = 1, \dots, N_c$, $F_c(\tau_{k+e-1}) = \Lambda \cdot A_k \cdot B_k^{e-1} \cdot M_c(\tau_k)$, les contraintes (III.7) sur $U_c(\tau_k)$ sont obtenues en remplaçant $X_c(\tau_{k+e-1})$ par (III.13) dans le problème d'optimisation (III.4). \square

Par conséquent, la méthode prédictive constante proposée permet d'utiliser des horizons de prédiction larges tout en évitant le problème de la complexité calculatoire. Une étude de la complexité sera menée ultérieurement dans ce chapitre. Néanmoins, dans [JVMS13], les auteurs ont souligné que la méthode de commande pour laquelle la trajectoire est une droite rectiligne par morceaux est moins complexe. Une trajectoire rectiligne est faisable pour certains cas mais peut être inadéquate pour d'autres qui nécessitent une trajectoire curviligne. Ce problème correspond au cas du "hill-climbing", détaillé dans le paragraphe suivant.

III.1.2 Problème du "hill-climbing"

Le "hill-climbing" est une phase pour laquelle le marquage s'éloigne dans un premier temps du marquage désiré ; ceci se traduit par une augmentation transitoire du critère J_M . Ce problème est lié à la fois à la structure du RdP et aux marquages initial et final. Pour atteindre un état final, l'état actuel doit parfois s'éloigner de la cible avant de l'atteindre car il n'existe aucun trajet direct tel que l'écart quadratique entre le marquage courant et la cible décroisse de façon monotone. À cause de son principe (minimisation du critère J), la commande prédictive peut bloquer le système dans un état à partir duquel J_M commence à croître. Afin d'éviter ce problème, il faut prévoir un horizon de prédiction large afin de prédire suffisamment loin jusqu'à un marquage qui fait décroître l'écart quadratique. La question cruciale est comment savoir au préalable l'horizon de prédiction nécessaire pour dépasser une phase de "hill-climbing" ? Théoriquement, il n'est pas possible d'avoir une telle information a priori. En outre, nous ne pouvons savoir si le θ -RdPC commandé possède ou non une phase de "hill-climbing" dans sa trajectoire. De ce fait, il faut élaborer une méthode qui détecte au fur et à mesure les phases de "hill-climbing" et

qui détermine de manière systématique l'horizon de prédiction requis. La solution explorée dans ces travaux de recherche est d'incrémenter l'horizon de prédiction, à partir d'une valeur initiale et sans dépasser une valeur maximale, tant que le critère obtenu à la fin de la prédiction $J(\tau_{k+N_c})$ ne diminue pas suffisamment par rapport au critère obtenu à la date τ_k . L'horizon de prédiction est donc variable ; il dépend non seulement du changement de région mais aussi des variations de J .

Les résultats précédents sont utilisés pour proposer la commande prédictive codée dans l'algorithme 1. Soient $N_0 \in \mathbb{N}_{>0}$ et $N_{\max} \in \mathbb{N}_{>0}$ respectivement les deux valeurs initiale et maximale de N_c . À l'instant initial, N_0 est affectée à N_c . Nous définissons $N_{limit}(\tau_k) \in \mathbb{N}_{>0}$ comme la borne supérieure de N_c qui garantit que tous les marquages intermédiaires sont dans la même région R_k durant l'intervalle $[\tau_k, \tau_{k+N_{limit}}]$. La commande d'un θ -RdPC est effectuée de la manière suivante : tout d'abord $N_{limit}(\tau_k)$ est initialisée à N_{\max} . La commande U_c , solution de l'optimisation quadratique (III.5) sous les contraintes (III.7), est déterminée et tous les marquages intermédiaires sont calculés. Une vérification de l'appartenance à la même région est ensuite effectuée. Si un marquage $M(\tau_{k+e})$, $e = 1, \dots, N_c - 1$, change de région, alors la valeur de e est attribuée à N_c et à $N_{limit}(\tau_k)$. Puis, le problème d'optimisation est répété en considérant cette fois-ci la nouvelle valeur de N_c . Dans le cas inverse (tous les marquages sont dans la même région R_k), une vérification de la décroissance de J est effectuée (vérifier si $J(\tau_k) - J(\tau_{k+N_c}) \leq 0$). Si ce n'est pas le cas, N_c s'incrémente tant qu'il ne dépasse ni $N_{limit}(\tau_k)$ ni N_{\max} jusqu'à ce que J devienne décroissant. Finalement, la commande et la valeur de N_c obtenues à la dernière itération sont celles retenues. U_c est appliquée à l'état courant du système, le marquage est calculé et ce même processus est répété à la période d'échantillonnage suivante sachant que la valeur N_c est la dernière obtenue et que $N_{limit}(\tau_{k+1})$ est réinitialisée par N_{\max} .

L'algorithme 1 est appelé par l'algorithme 2 qui représente la procédure globale de commande d'un θ -RdPC, par le biais de la commande prédictive, jusqu'à l'atteinte de la configuration désirée.

Algorithme 1 : Commande prédictive (Entrées : $R_k, M_c(\tau_k), N_c, N_{\max}$ et $J(\tau_k)$; Sorties : $U_c(\tau_k)$ et N_c)

```

1: Initialisation des paramètres :
    $solution \leftarrow 0$ 
    $N_{limit}(\tau_k) \leftarrow N_{\max}$ 
2: Tant que  $solution = 0$ 
3:    $solution \leftarrow 1$ 
4:   Calculer  $U_c(\tau_k)$ , solution du problème d'optimisation (III.5)
   sous les contraintes (III.7)
5:    $e \leftarrow 1$ 
6:   Tant que  $e < N_c$ 
7:     Déterminer la région  $R_c(M_c(\tau_{k+e}))$  du marquage  $M_c(\tau_{k+e})$ 
8:     Si  $R_c(M_c(\tau_{k+e})) \neq R_k$ , alors
9:        $N_{limit}(\tau_k) \leftarrow e$ 
10:       $N_c \leftarrow e$ 
11:       $solution \leftarrow 0$ 
12:     Sinon
13:        $e \leftarrow e + 1$ 
14:     Fin si
15:   Fin tant que
16:   Si  $solution = 1$ , alors
17:     Calculer le critère  $J(\tau_{k+N_c})$ 
18:     Si  $(J(\tau_k) - J(\tau_{k+N_c}) \leq 0)$  et  $(N_c < N_{limit}(\tau_k))$ , alors
19:        $N_c \leftarrow N_c + 1$ 
20:        $solution \leftarrow 0$ 
21:     Fin si
22:   Fin si
23: Fin tant que
24: Retourner  $U_c(\tau_k)$  et  $N_c$ 

```

Algorithme 2 : Commande du θ -RdPC via la commande prédictive (Entrées : M_{0c} et N_0 ; Sorties : U_c , M_c , X_c et J)

- 1: Initialisation des paramètres :
 - $N_c \leftarrow N_0$
 - $fin \leftarrow 0$
 - $\tau_k \leftarrow 0$
 - $M_c(\tau_k) \leftarrow M_{0c}$
 - $U_c \leftarrow \emptyset$
 - 2: Calculer $X_c(\tau_k)$ et $J(\tau_k)$
 - 3: $M_c \leftarrow M_c(\tau_k)$, $X_c \leftarrow X_c(\tau_k)$, $J \leftarrow J(\tau_k)$
 - 4: **Tant que** $fin = 0$
 - 5: Déterminer R_k
 - 6: Calculer la commande $U_c(\tau_k)$ et déterminer N_c par le biais de l'algorithme 1
 - 7: Calculer $X(\tau_k)$, $M_c(\tau_{k+1})$ et $J(\tau_{k+1})$
 - 8: $M_c \leftarrow M_c \cup M_c(\tau_k)$
 - 9: $X_c \leftarrow X_c \cup X_c(\tau_k)$
 - 10: $J \leftarrow J \cup J(\tau_k)$
 - 11: $U_c \leftarrow U_c \cup \emptyset$
 - 12: **Si** $J(\tau_{k+1}) = 0$, **alors**
 - 13: $end \leftarrow 1$
 - 14: **Sinon**
 - 15: $\tau_k \leftarrow \tau_{k+1}$
 - 16: **Fin si**
 - 17: **Fin tant que**
 - 18: Retourner U_c , M_c , X_c et J
-

Des propriétés d'existence de la commande prédictive sont énoncées dans la proposition III.1.2.

Proposition III.1.2. *Soit $\langle \mathcal{N}_c, M_0, \lambda, \theta, T_c^{comm} \rangle$ un θ -RdPC tel que θ vérifie la condition (I.35) et que $T_c^{comm} = T$. Supposons que $M_{refc} > 0_{n \times 1}$ soit atteignable à partir de $M_{0c} > 0_{n \times 1}$ et que X_{refc} respecte les conditions (II.9) pour que la configuration désirée soit un état d'équilibre. Alors :*

- *Le problème d'optimisation quadratique (III.5) sous les contraintes (III.7) est toujours faisable.*
- *L'algorithme 2 assure la convergence du θ -RdPC vers la configuration désirée en un nombre fini de pas.*
- *Il existe une commande qui permet de maintenir M_{refc} .*

Démonstration.

- $\forall e = 1, \dots, N_c, \Lambda \cdot A_k \cdot B_k^{e-1} \cdot M_c(\tau_k) \geq 0_{q \times n}$ car $M_c(\tau_k) \geq 0_{n \times 1}$. Donc, le domaine défini par l'intersection des contraintes (a) et (b) (de l'équation (III.7)) est non-vidé; il existe toujours une commande U_c appartenant à cette intersection, en particulier, $U_c = 0_{q \times 1}$. Par conséquent, le problème d'optimisation donné par (III.5) sous les contraintes (III.7) est toujours faisable.
- Puisque $M_{0c} > 0_{n \times 1}, M_{refc} > 0_{n \times 1}$ et que θ vérifie la condition (I.35), alors d'après [MRTRS08], tous les marquages sont strictement positifs et le marquage désiré est atteint en un nombre fini de pas.
- Finalement, X_{refc} est choisi de telle sorte qu'il vérifie la condition (II.9) : $0 \leq \Lambda \cdot A_k \cdot M_{refc} - X_{refc} \leq \Lambda \cdot A_k \cdot M_{refc}$. Supposons que $U_c(\tau_k) = \Lambda \cdot A_k \cdot M_{refc} - X_{refc}$. D'après (II.8), $M_c(\tau_{k+1}) = B_k \cdot M_{refc} - \theta \cdot W \cdot U_c(\tau_k) = M_{refc} + \theta \cdot W \cdot (\Lambda \cdot A_k \cdot M_{refc} - U_{refc}) = M_{refc}$. La commande $U_c(\tau_k)$ permet de maintenir le marquage désiré.

□

Afin de mettre en valeur la complexité de calcul réduite de la commande prédictive constante, une étude de complexité est menée et une comparaison avec la méthode usuelle est effectuée dans la sous-section suivante.

III.1.3 Étude de la complexité de calcul

En vue de comparer l'approche proposée dans ce chapitre avec d'autres, en particulier l'approche implicite proposée dans [GMR⁺06], les tableaux III.1, III.2, III.3 et III.4 déterminent, pour chaque cas, l'effort de calcul engendré par une prédiction avec N_c pas de

calcul. Ces différents cas peuvent être distingués en fonction de la définition du critère J (forme standard (II.17) ou forme simplifiée (III.1)), de la prise en compte du changement de régions ou non durant la prédiction et finalement en fonction de la commande continue (constante ou variable).

La complexité est évaluée par rapport au nombre de variables, au nombre de contraintes et par rapport au nombre d'opérations élémentaires nécessaires pour calculer le critère J sur l'horizon de prédiction considéré.

TABLE III.1 – Complexité de la commande prédictive qui utilise le critère (III.1) sans tenir compte du changement de région

	U_c constante	U_c variable
Nombre de variables	q	$N_c \cdot q$
Complexité par rapport aux contraintes	$O(N_c \cdot q)$	$O(q \cdot (N_c))$
Complexité par rapport à J	$O(N_c \cdot (n + q))$	$O(N_c \cdot (n + q))$

TABLE III.2 – Complexité de la commande prédictive qui utilise le critère (III.1) en tenant compte du changement de région

	U_c constante	U_c variable
Nombre de variables	q	$N_c \cdot q$
Complexité par rapport aux contraintes	$O(K^{N_c-1} \cdot N_c \cdot q)$	$O(K^{N_c-1} \cdot q \cdot (N_c))$
Complexité par rapport à J	$O(K^{N_c-1} \cdot N_c \cdot (n + q))$	$O(K^{N_c-1} \cdot N_c \cdot (n + q))$

avec K le nombre total de régions.

TABLE III.3 – Complexité de la commande prédictive qui utilise le critère (II.17) sans tenir compte du changement de région

	U_c constante	U_c variable
Nombre de variables	q	$N_c \cdot q$
Complexité par rapport aux contraintes	$O(N_c \cdot q)$	$O(q \cdot (N_c))$
Complexité par rapport à J	$O(N_c^2 \cdot (n + q))$	$O(N_c^2 \cdot (n + q))$

TABLE III.4 – Complexité de la commande prédictive qui utilise le critère (II.17) en tenant compte du changement de région

	U_c constante	U_c variable
Nombre de variables	q	$N_c \cdot q$
Complexité par rapport aux contraintes	$O(K^{N-1} \cdot N \cdot q)$	$O(K^{N-1} \cdot q \cdot (N - 1))$
Complexité par rapport à J	$O(K^{N_c-1} \cdot N_c^2 \cdot (n + q))$	$O(K^{N_c-1} \cdot N_c^2 \cdot (n + q))$

La première et la deuxième colonne du tableau III.1 représentent respectivement les complexités de la commande prédictive proposée et de la commande prédictive usuelle présentées dans [GMR⁺06]. D'après ces tableaux, nous pouvons affirmer que les trois facteurs qui influencent explicitement la complexité de calcul, sont le type de la commande (constante ou variable), la commutation entre les régions et la prise en compte des marquages et des flux intermédiaires. Si la commande est variable, le nombre de variables à déterminer sera multiplié par un facteur N_c . Dans le cas d'un changement de région, la complexité devient proportionnelle au nombre de régions. Finalement, la complexité de la fonction coût est N_c fois plus grande si les points intermédiaires sont considérés dans le calcul. Pour ces raisons, la commande prédictive constante proposée présente la complexité de calcul la plus faible par rapport aux autres présentées. En outre, une propriété très utile caractérise cette commande : la complexité par rapport à J est polynomiale en fonction de N_c . Cette caractéristique permettra de déterminer, de façon systématique, l'horizon de prédiction maximal.

Dans ce qui suit, nous nous focaliserons sur les trois approches prédictives Ap_1 , Ap_2 et Ap_3 telles que :

* Ap_1 est l'approche prédictive proposée pour laquelle :

- Les flux et les marquage intermédiaires ne sont pas pris en compte dans le critère de performance (III.1).
- La commande est supposée être constante au cours de la prédiction.
- Le marquage ne change pas de région avant le dernier pas de prédiction.

La complexité de cette approche est donnée par la première colonne du tableau III.1.

* Ap_2 est une commande prédictive pour laquelle :

- Les flux et les marquage intermédiaires sont pas pris en compte dans le critère de performance (II.17).
- La commande est supposée être constante au cours de la prédiction.
- Le marquage ne change pas de région avant le dernier pas de prédiction.

La complexité de Ap_2 correspond à la première colonne du tableau III.3.

* Ap_3 est la commande prédictive standard telle que :

- Les flux et les marquage intermédiaires ne sont pas pris en compte dans le critère de performance (II.17).
- La commande est variable au cours de la prédiction.
- La commutation entre les régions est prise en compte au cours de la prédiction.

La complexité de Ap_3 est donnée par la deuxième colonne du tableau III.4) :

III.1.4 Détermination de l'horizon de prédiction maximal

Le paramètre N_{\max} est extrêmement important dans la commande prédictive. À cet égard, il doit être choisi avec prudence pour deux raisons. D'une part, il doit être suffisamment grand afin d'assurer la convergence dans le cas de présence d'une phase de "hill-climbing" dans la trajectoire. D'autre part, il ne doit pas être trop grand afin de limiter la profondeur de la prédiction, et de faire en sorte que la solution proposée soit réalisable. En particulier, le temps nécessaire pour résoudre le problème d'optimisation, à un instant donné, ne doit pas dépasser la période d'échantillonnage.

La proposition III.1.3 donne une expression analytique de la borne supérieure de N_{\max} .

Proposition III.1.3. *Soit $\langle \mathcal{N}_c, M_0, \lambda, \theta, T_c^{\text{comm}} \rangle$ un θ -RdPC tel que θ vérifie la condition (I.35) et que toutes les transitions soient commandables via la commande prédictive. Alors, pour garantir que la durée de calcul est inférieure à θ , N_{\max} doit vérifier la condition (III.17) :*

$$N_{\max} \leq \left\lfloor \frac{\theta}{\Psi(1)} \right\rfloor \quad (\text{III.17})$$

où $\Psi(1)$ représente le temps de calcul nécessaire pour résoudre le problème d'optimisation quadratique (III.5) sous les contraintes (III.7), pour un horizon de prédiction de taille $N_c = 1$.

Démonstration. Dans l'approche proposée, la complexité par rapport à J est linéaire en fonction de N_c . Par conséquent, le temps requis pour optimiser J à un instant donné avec l'algorithme 2 et pour un horizon de N_{\max} pas, est aussi linéaire en fonction de N_c . Il peut être exprimé par : $\Psi(N_{\max}) = \Psi(1) \cdot N_{\max}$. De plus, la commande prédictive est implémentable si cette durée est inférieure à θ . En conséquence, l'inégalité (III.17) est obtenue. \square

III.1.5 Exemple illustratif 1

Afin d'illustrer l'efficacité de la commande prédictive constante proposée, nous considérons le RdPC représenté dans la figure III.2.

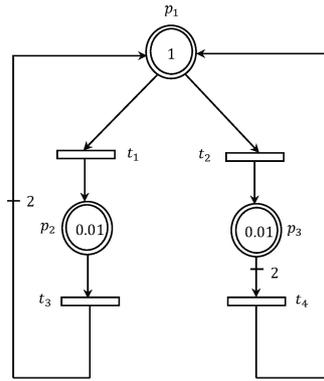
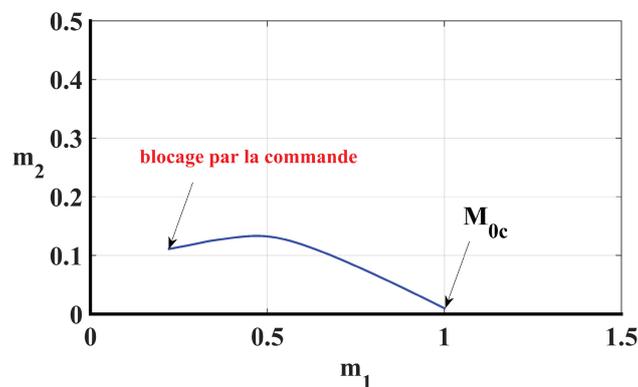


FIGURE III.2 – Exemple 1 de RdPC

Le marquage initial et les vitesses maximales de franchissement des transitions, qui caractérisent ce système, sont respectivement données par $M_{0c} = [1 \ 0.01 \ 0.01]^t$ et $\lambda = [1 \ 1 \ 1 \ 1]^t$ (par hypothèse $M_{0c} > 0_{n \times 1}$, ici $\forall i = 1, \dots, n, m_i \geq \epsilon$ avec $\epsilon = 0.01$). Toutes les transitions sont commandables. Toutes les simulations ont été effectuées telles que la configuration désirée est ($M_{refc} = [2 \ 0.5 \ 0.5]^t$, $X_{refc} = [0.25 \ 0.5 \ 0.25 \ 0.25]^t$), $\theta = 0.1$ U.T., $r_{M_c} = 1$, $r_{x_c} = 1$, $N_0 = 1$ et $N_{max} = 25$.

Les commandes prédictives Ap_1 , Ap_2 et Ap_3 avec N_c constant et égal à 1 ne parviennent pas à atteindre la configuration désirée. En effet, dans le but de diminuer le critère de performance, la commande prédictive a choisi de franchir les transitions t_3 et t_4 . Par conséquent, la valeur du critère J chute de 2.41 à 1.67. Ensuite, elle augmente à 3.52. Les commandes prédictives avec $N_c = 1$ ne prédisent pas suffisamment loin pour détecter que J va ensuite diminuer, le θ -RdPC reste donc bloqué dans le marquage $[0.22 \ 0.11 \ 0.22]^t$ (voir les figures III.3 et III.4).

FIGURE III.3 – Trajectoire du marquage dans le plan de projection (m_1, m_2) pour $N_c = 1$

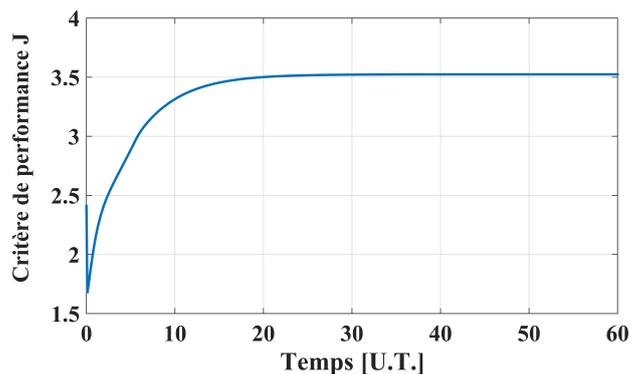


FIGURE III.4 – Évolution temporelle du critère de performance J pour $N_c = 1$

Pour faire décroître J et s'approcher de la configuration désirée, la transition t_1 devrait être franchie afin de produire des jetons. Par conséquent, une augmentation transitoire de J est primordiale.

Afin de montrer l'efficacité de la méthode proposée, l'algorithme 2 est appliqué au θ -RdPC. Il a été constaté que lorsque $\theta = 0.1$ U.T., l'horizon de prédiction doit augmenter jusqu'à $N_c = 16$ afin de dépasser le "hill-climbing".

La figure III.5 montre la trajectoire des approches Ap_1 , Ap_2 et Ap_3 pour $N_c = 16$.

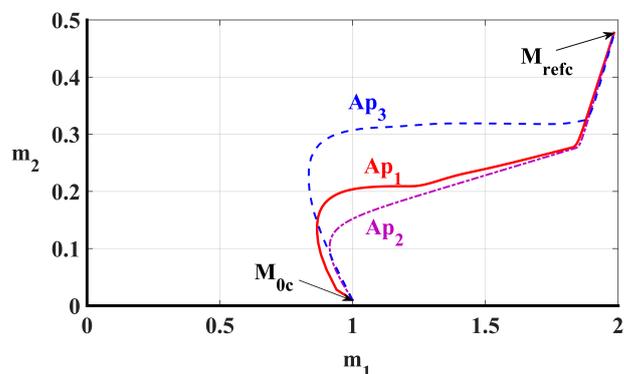


FIGURE III.5 – Trajectoires du marquage, dans le plan de projection (m_1, m_2) , des approches Ap_1 , Ap_2 et Ap_3 pour $N_c = 16$

Nous observons que pour $N_c = 16$, toutes les stratégies de commande réussissent à atteindre la configuration souhaitée. Cependant, quelques différences importantes doivent être mentionnées dans le but d'évaluer les performances de chaque approche. Pour ce faire, le tableau III.5 résume, pour chaque commande, l'erreur quadratique moyenne entre chaque configuration intermédiaire et celle souhaitée, le temps nécessaire pour atteindre la configuration désirée, le temps de calcul moyen et le temps de calcul maximal requis pour calculer l'action de commande à chaque étape.

TABLE III.5 – Les performances des approches Ap_1 , Ap_2 et Ap_3

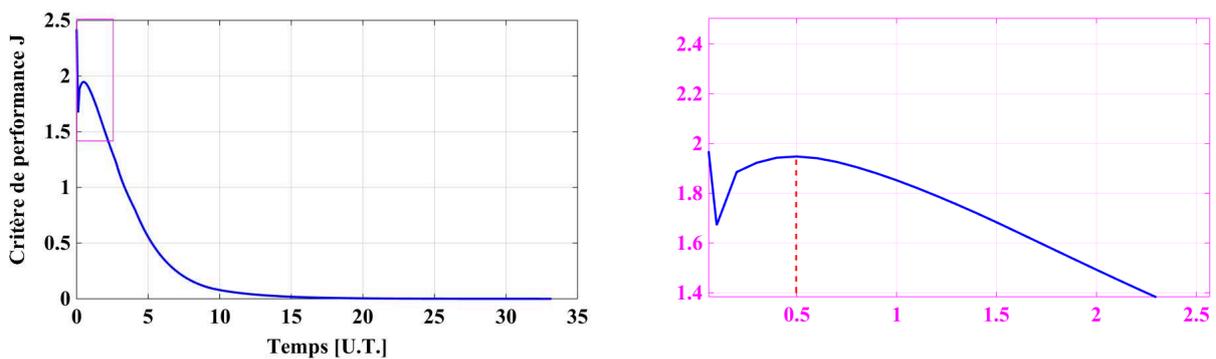
	Erreur quadratique	Temps de convergence	Temps de calcul moyen	Temps de calcul maximal
Ap_1	0.3087	25.5	0.0129	0.0355
Ap_2	0.3137	35.3	0.0178	0.0354
Ap_3	0.3714	16.00	0.0556	0.1853

Nous remarquons que les erreurs quadratiques moyennes sont plus ou moins identiques pour les trois trajectoires.

De point de vue rapidité, la trajectoire obtenue avec l'approche Ap_3 est la plus rapide par rapport aux deux autres, bien qu'elle présente une trajectoire moins directe vers la cible, l'erreur quadratique est la plus grande avec cette approche.

Concernant les temps de calcul moyens et maximaux requis pour calculer l'action de commande à chaque étape, ils sont très grands avec Ap_3 contrairement à Ap_2 et en particulier à Ap_1 . En moyenne, le temps de calcul a été divisé par 4 en utilisant la fonction de coût proposée (II.17) et les contraintes supplémentaires. Le temps de calcul maximal requis avec Ap_3 empêche l'implémentation de cette stratégie de commande en temps réel avec une période d'échantillonnage de 0.1 T.U. ce qui n'est pas le cas pour Ap_1 et Ap_2 .

Les résultats obtenus suite à l'application de l'algorithme 2 au θ -RdPC, sont présentés par les figures III.6 et III.7.

FIGURE III.6 – Évolution temporelle du critère de performance J pour N_c adaptatif

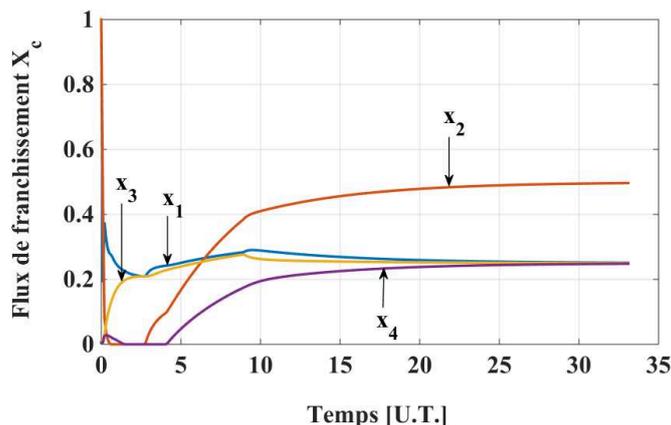


FIGURE III.7 – Évolution temporelle du flux de franchissement X_c pour N_c adaptatif

L'évolution du critère de performance illustre que le marquage doit d'abord s'éloigner du marquage désiré ce qui entraîne un "hill-climbing". Puis, à partir de la date $\tau_k \approx 0.5$ U.T., J commence à diminuer et le marquage et le flux se rapprochent des valeurs désirées. L'évolution de $J(\tau_{k+N_c})$ à la fin de l'horizon de prédiction reste toujours inférieure à la valeur courante de $J(\tau_k)$.

Afin de montrer expérimentalement la robustesse de la commande prédictive face aux perturbations, l'évolution du θ -RdPC (présenté par la figure III.2) commandé par la commande prédictive proposée, a été perturbé durant l'intervalle de temps $[9, 9.2]$ où la commande de la transition est franchie avec son flux maximale de franchissement (c-à-d $u_1 = 0, 9 \leq \tau_k \leq 9.2$) (voir figure III.8).

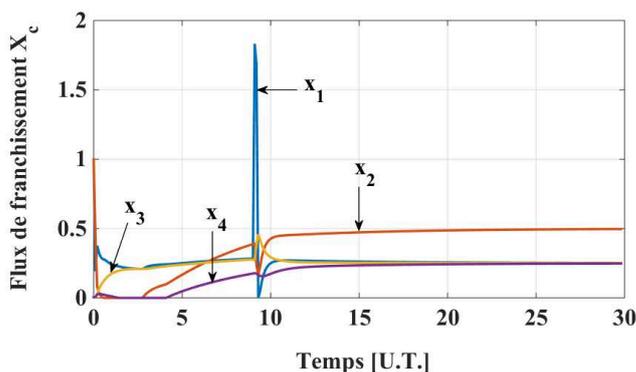


FIGURE III.8 – Évolution temporelle du flux de franchissement X_c en présence d'une perturbation

Les figures III.9 et montrent la trajectoire du marquage et l'évolution du J en présence de cette perturbation.

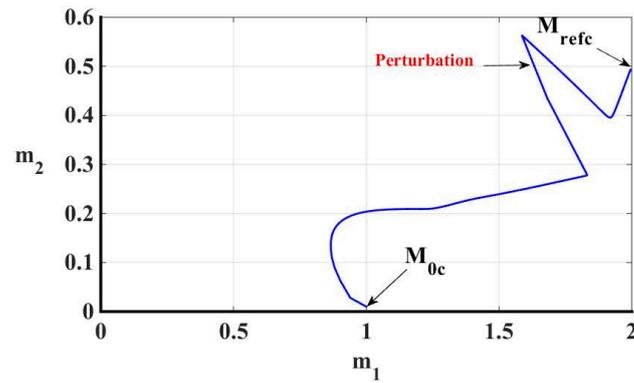


FIGURE III.9 – Trajectoire du marquage dans le plan de projection (m_1, m_2) en présence d'une perturbation

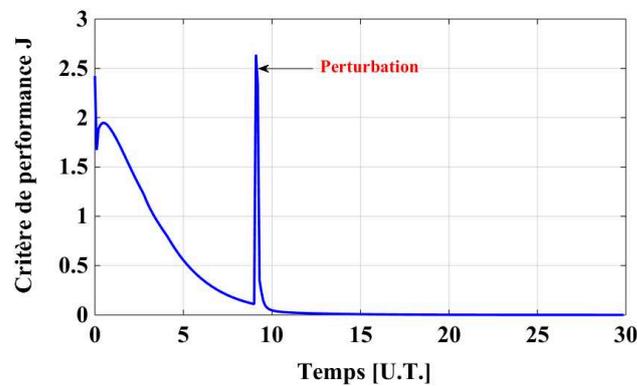


FIGURE III.10 – Évolution temporelle du critère de performance J en présence d'une perturbation

Nous remarquons que même en présence d'une perturbation qui écarte le marquage courant du marquage désiré, la commande prédictive détermine la commande continue qui permet au système de converger vers la configuration désirée. Nous déduisons que la commande prédictive est une commande robuste aux perturbations grâce à son processus d'optimisation en ligne. De plus, elle est compatible avec des contraintes temps réels [GMR⁺06]. Néanmoins, elle peut engendrer de fortes variations du flux, traduites dans la pratique par une sollicitation excessive des actionneurs du système commandé. Ce problème est critique car il représente l'une des causes de vieillissement précoce des composants. À cet égard, le but de la section suivante est de conserver les avantages de la commande prédictive et de limiter la sollicitation des actionneurs.

III.2 Limitation de la variation du flux

III.2.1 Principe de la commande prédictive avec limitation de la variation du flux

Afin de limiter la sollicitation excessive des actionneurs, un nouveau terme, qui prend en compte la variation du flux (c-à-d l'accélération), est ajouté au critère de performance global de la commande prédictive. Ce terme est désigné par $J_{dX_c}(\tau_k)$. Par conséquent, J est désormais exprimé par (III.18) :

$$J(\tau_{k+N_c}) = J_M(\tau_{k+N_c}) + J_X(\tau_{k+N_c-1}) + J_{dX_c}(\tau_k) \quad (\text{III.18})$$

tel que $J_{dX_c}(\tau_k)$ est défini par (III.19) :

$$J_{dX_c}(\tau_k) = \theta^2 \cdot A_{acc}(\tau_k)^t \cdot Q_{dX_c} \cdot A_{acc}(\tau_k) \quad (\text{III.19})$$

$Q_{dX_c} \in \mathbb{R}_{\geq 0}^{q \times q}$ est la matrice de pondération de J_{dX_c} . Par souci de simplicité, elle est choisie diagonale telle que $Q_{dX_c} = r_{dX_c} \cdot I_q$, où $r_{dX_c} \in \mathbb{R}_{\geq 0}$. $A_{acc}(\tau_k)$ dénote une estimation du vecteur d'accélération à la date τ_k , définie par (III.20) :

$$A_{acc}(\tau_k) = \frac{1}{\theta} \cdot (X_c(\tau_k) - X_c(\tau_{k-1})) \quad (\text{III.20})$$

Par conséquent, $J_{dX_c}(\tau_k)$ peut être simplement exprimé par (III.21) :

$$J_{dX_c}(\tau_k) = r_{dX_c} \cdot \theta^2 \cdot \|A_{acc}(\tau_k)\|^2 \quad (\text{III.21})$$

où $\|\cdot\|$ représente la norme euclidienne usuelle dans \mathbb{R}^q .

Aucune nouvelle contrainte n'est ajoutée au problème d'optimisation à résoudre.

Le problème d'optimisation est identique à celui donné par (III.4) sauf que désormais $J(\tau_{k+N_c})$ est donné par (III.18).

La proposition III.2.1 donne les nouvelles expressions du problème d'optimisation quadratique.

Proposition III.2.1. *Soit $\langle \mathcal{N}_c, M_0, \lambda, \theta, T_c^{comm} \rangle$ un θ -RdPC tel que θ vérifie la condition (I.35) et $T_c^{comm} = T$. Soit $N_c > 0$ l'horizon de prédiction continu. Supposons que $M_{refc} > 0_{n \times 1}$ soit atteignable à partir de $M_{0c} > 0_{n \times 1}$, que X_{refc} respecte les conditions (II.9) pour que la configuration désirée soit un état d'équilibre et que, durant la prédiction, les marquages $M(\tau_{k+e})$, $e = 1 \dots, N_c - 1$ restent dans la même région R_k . Alors, la commande continue $U_c(\tau_k)$ est la solution du problème d'optimisation quadratique (III.22)*

sous les contraintes (III.24) :

$$U_c(\tau_k) = \arg \min_{U_c} \{J_{U_c}(\tau_{k+N_c})\} \quad (\text{III.22})$$

tel que le critère $J_{U_c}(\tau_{k+N_c})$ s'écrit sous la forme (III.23) :

$$J_{U_c}(\tau_{k+N_c}) = \frac{1}{2} \cdot U_c(\tau_k)^t \cdot \Omega \cdot U_c(\tau_k) + \rho(\tau_k)^t \cdot U_c(\tau_k) \quad (\text{III.23})$$

et :

$$\forall e = 1, \dots, N_c \quad \begin{array}{l} \mathbf{a}/ S_{e-1} \cdot U_c(\tau_k) \geq 0_{q \times 1} \\ \mathbf{b}/ S_{e-1} \cdot U_c(\tau_k) \leq \Lambda \cdot A_k \cdot B_k^{e-1} \cdot M_c(\tau_k) \end{array} \quad (\text{III.24})$$

Les termes Ω et $\rho(\tau_k)$ sont respectivement définis par (III.25) tels que Σ_{N_c} et S_{N_c-1} sont respectivement définis par (III.10) et (III.11) :

$$\Omega = \theta^2 \cdot W^t \cdot \Sigma_{N_c}^t \cdot Q_{M_c} \cdot \Sigma_{N_c} \cdot W + S_{N_c-1}^t \cdot Q_{X_c} \cdot S_{N_c-1} + Q_{dX_c} \quad (\text{III.25})$$

$$\begin{aligned} \rho(\tau_k) = & \theta \cdot W^t \cdot \Sigma_{N_c}^t \cdot Q_{M_c} \cdot (M_{refc} - B_k^{N_c} \cdot M_c(\tau_k)) \\ & + S_{N_c-1}^t \cdot Q_{X_c} \cdot (X_{refc} - \Lambda \cdot A_k \cdot B_k^{N_c-1} \cdot M_c(\tau_k)) \\ & + Q_{dX_c} \cdot (X_c(\tau_{k-1}) - \Lambda \cdot A_k \cdot M_c(\tau_k)) \end{aligned} \quad (\text{III.26})$$

Démonstration. En supprimant tous les termes constants par rapport à $U_c(\tau_k)$, les formes réduites de $J_M(\tau_{k+N_c})$, $J_X(\tau_{k+N_c-1})$ et $J_{dX_c}(\tau_{k+N_c})$, désignées respectivement par $J_{M_{U_c}}(\tau_{k+N_c})$, $J_{X_{U_c}}(\tau_{k+N_c-1})$ et $J_{dX_{U_c}}(\tau_{k+N_c})$, sont données par (III.14), (III.15) et (III.27) :

$$J_{dX_{U_c}}(\tau_k) = U_c(\tau_k)^t \cdot Q_{dX_c} \cdot U_c(\tau_k) + 2 \cdot (X_c(\tau_{k-1}) - \Lambda \cdot A_k \cdot M_c(\tau_k))^t \cdot Q_{dX_c} \cdot U_c(\tau_k) \quad (\text{III.27})$$

En conséquence, $J_{U_c}(\tau_{k+N_c})$ est désormais exprimé par (III.28) :

$$J_{U_c}(\tau_{k+N_c}) = \frac{1}{2} \cdot (J_{M_{U_c}}(\tau_{k+N_c}) + J_{X_{U_c}}(\tau_{k+N_c-1}) + J_{dX_{U_c}}(\tau_k)) \quad (\text{III.28})$$

Cette dernière équation est réécrite sous la forme (III.23) où Ω et $\rho(\tau_k)$ sont respectivement définis (III.25) et (III.26). \square

Désormais, à chaque date τ_k , l'algorithme 2 commande le θ -RdPC et détermine, à l'aide de l'algorithme 1, $U_c(\tau_k)$ solution du nouveau problème d'optimisation (III.22) sous les contraintes (III.24).

Remarque 2. Signalons que le nouveau problème d'optimisation est faisable pour tout $M_c(\tau_k) > 0_{n \times 1}$.

III.2.2 Exemple illustratif 2

Afin de mettre en évidence la limitation de la variation du flux grâce à l'approche proposée et l'influence du paramètre r_{dX_c} , nous aurons recours à un exemple illustratif pour lequel un changement brusque de la configuration souhaitée aura lieu.

Le RdPC représenté par la figure III.11 est caractérisé par $M_{0c} = \begin{bmatrix} 0.99 & 0.01 & 0.01 & 0.99 \end{bmatrix}^t$ et $\lambda = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^t$:

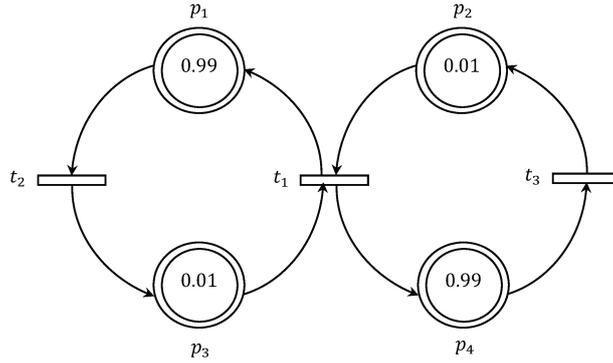


FIGURE III.11 – Exemple 2 de RdPC

La commande prédictive détermine la commande U_c qui permet au système d'atteindre la première configuration désirée définie par $(M_{refc1} = \begin{bmatrix} 0.2 & 0.6 & 0.8 & 0.4 \end{bmatrix}^t, X_{refc1} = \begin{bmatrix} 0.2 & 0.2 & 0.2 \end{bmatrix}^t)$. À la date $\tau_k = 3.5$ T.U., la cible est remplacée par la deuxième configuration souhaitée $(M_{refc2} = \begin{bmatrix} 0.3 & 0.3 & 0.7 & 0.7 \end{bmatrix}^t, X_{refc2} = \begin{bmatrix} 0.3 & 0.3 & 0.3 \end{bmatrix}^t)$. Toutes les simulations numériques sont effectuées telles que : $\theta = 0.01$ T.U., $N_0 = 1$, $N_{\max} = 5$, $r_{M_c} = 1$ et $r_{X_c} = 0.01$. Différentes valeurs de r_{dX_c} sont considérées.

Préalablement à la présentation et à la comparaison des résultats obtenus, il convient de définir A_C en tant que valeur cumulée du carré de la norme d'accélération multipliée par θ .

$A_C(\tau_k)$ sera utilisé dans ce qui suit comme un élément de comparaison. Son expression analytique, à la date τ_k , est donnée par (III.29) :

$$A_C = \theta \cdot \sum_{s=1}^k \|A_{acc}(\tau_s)\|^2 \quad (\text{III.29})$$

Les figures III.12 et III.13 présentent l'évolution temporelle de la norme d'accélération ainsi que les valeurs finales de A_C pour trois valeurs différentes de r_{dX_c} et montrent comment une valeur non-nulle permet de limiter la variation de flux.

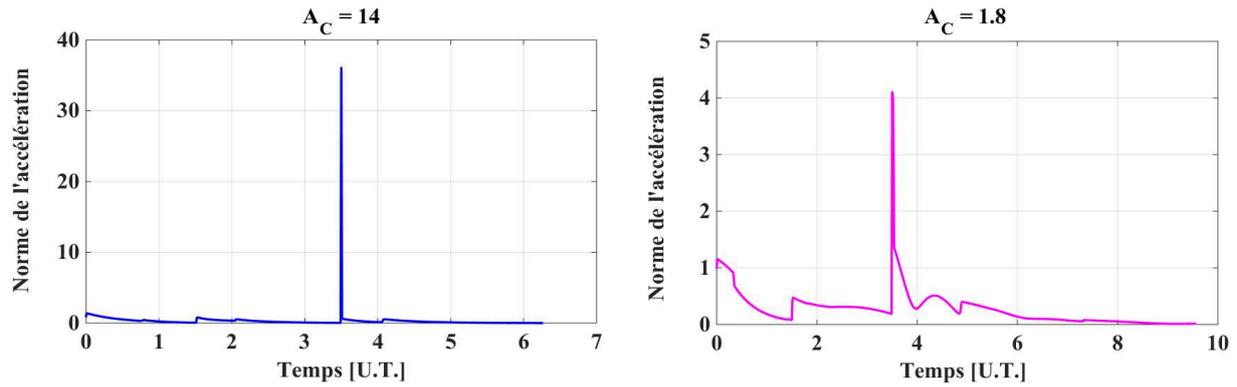


FIGURE III.12 – Norme de l'accélération et la valeur de A_C lorsque $r_{dX_c} = 0$ (gauche) et $r_{dX_c} = 1$ (droite)

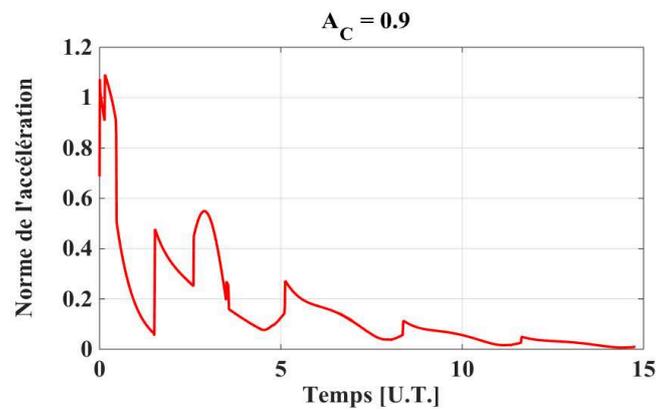
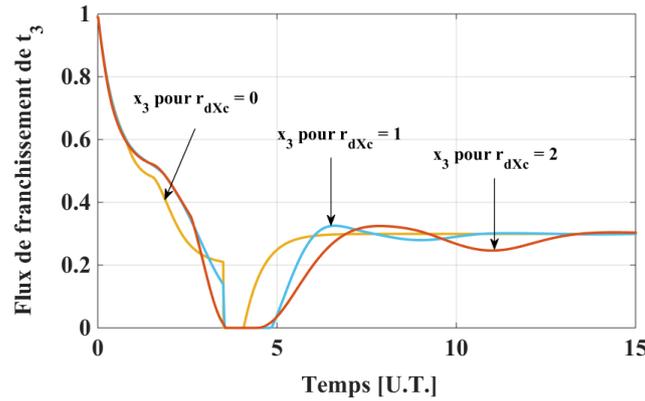
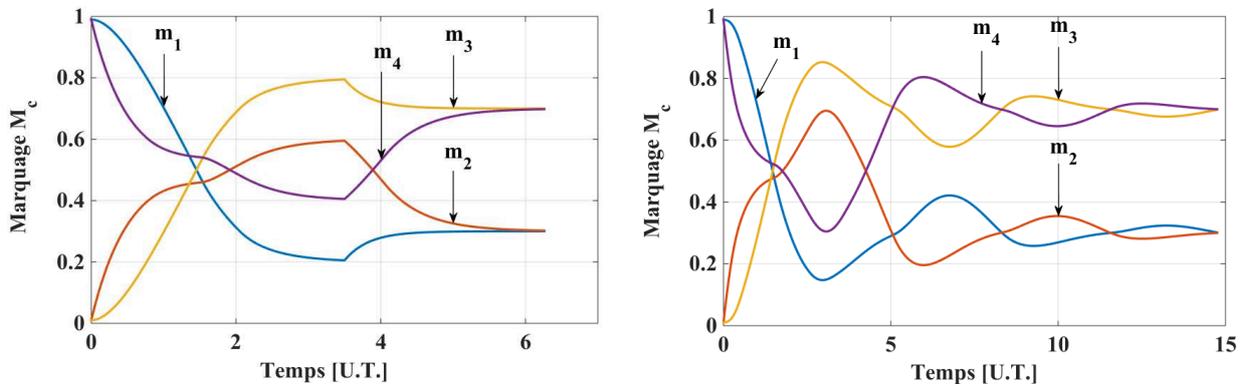


FIGURE III.13 – Norme de l'accélération et la valeur de A_C lorsque $r_{dX_c} = 2$

Ainsi, l'accélération est significativement réduite. Sa valeur maximale décroît de 36.1 à 1.1 lorsque r_{dX_c} passe de 0 à 2, ce qui explique la baisse de A_C de 14 à 0.9. La variation du flux diminue de 87% pour $r_{dX_c} = 1$ et de 93% pour $r_{dX_c} = 2$ par rapport à sa valeur initiale. Plus r_{dX_c} est important plus le poids de J_{dX_c} est important dans J et plus son influence est significative.

Essayons maintenant de voir le comportement d'un tel système pour les différentes valeurs de r_{dX_c} . Nous présentons dans la figure III.14 la variation du flux de la transition t_3 (les autres transitions possèdent le même comportement) et dans la figure III.15 l'évolution temporelle du marquage pour $r_{dX_c} = 0$ et $r_{dX_c} = 2$.

FIGURE III.14 – Variation du flux de la transition t_3 pour différentes valeurs de r_{dX_c} FIGURE III.15 – Évolution temporelle du marquage M_c pour $r_{dX_c} = 0$ (gauche) et $r_{dX_c} = 2$ (droite)

À ce stade, il s'avère important d'effectuer une analyse critique de la méthode proposée afin de l'améliorer. Une augmentation trop forte de r_{dX_c} provoque une dégradation dans les performances de la commande. En effet, une telle augmentation provoque un comportement oscillatoire autour de la configuration désirée ainsi qu'un ralentissement de la convergence. Effectivement, le temps de convergence évolue de 6.25 U.T. jusqu'à 9.54 T.U. pour $r_{dX_c} = 1$ et 14.74 T.U. pour $r_{dX_c} = 2$. Ce problème est dû à l'addition du terme J_{dX_c} qui impose au système de progresser très doucement vers la configuration désirée.

La déstabilisation du système est due au fait que la commande doit obéir à deux objectifs parfois opposés : varier le flux pour converger, et éviter en même temps les fortes variations. Ainsi, deux phénomènes se produisent de façon alternative : Si $J_{dX_c}(\tau_k)$ est plus grand par rapport à $J_M(\tau_{k+N_c})$ et $J_X(\tau_{k+N_c-1})$, l'optimisation favorise la trajectoire qui limite la variation, cette trajectoire n'est pas forcément celle qui s'approche de la configuration désirée. Par conséquent, la configuration courante $(M_c(\tau_{k+1}), X_c(\tau_k))$ s'éloigne des valeurs désirées. Puis, $J_M(\tau_{k+N_c})$ et $J_X(\tau_{k+N_c-1})$ vont l'emporter sur $J_{dX_c}(\tau_k)$, la commande est déterminée de telle sorte que le système se rapproche le plus de la configuration

désirée, ce qui va certainement provoquer une accélération plus importante, $J_{dX_c}(\tau_k)$ augmente et les deux phénomènes se répètent. Ainsi, comme illustré dans la figure III.14, l'amplitude des fluctuations est directement liée à la valeur de r_{dX_c} . Le seul paramètre sur lequel nous puissions agir pour compenser les faiblesses étudiées est r_{dX_c} . Une valeur raisonnable de ce facteur, qui répond aux exigences cherchées par rapport au taux de limitation de la sollicitation des actionneurs, à la rapidité du système et à l'amplitude des oscillations, peut être éventuellement déterminée graphiquement en traçant l'un de ces critères en fonction des différentes valeurs de r_{dX_c} . Le problème crucial auquel il faut faire face, est comment assurer la stabilité du système en présence de r_{dX_c} . Il est donc recommandé de déterminer graphiquement, sur toute la trajectoire, une valeur de r_{dX_c} qui provoque peu d'oscillations [TLL14b]. Selon l'amplitude des oscillations, $r_{dX_c} = 1$ semble être ici un bon choix. Or, cette méthode graphique n'est ni précise ni générique. La section suivante propose une méthode plus systématique qui se base sur une étude de stabilité afin de trouver la valeur optimale à chaque instant.

III.3 Étude de stabilité

Dans cette partie, une adaptation en ligne du terme r_{dX_c} est effectuée à chaque période d'échantillonnage. Par ailleurs, une étude de stabilité est menée dans le voisinage du marquage souhaité [TLL14a].

III.3.1 Modification de la fonction coût

L'objectif de l'adaptation de r_{dX_c} consiste à éviter les comportements oscillatoires et à accélérer la convergence. Si le critère J_M satisfait l'inégalité (III.30) (c-à-d J_M est strictement décroissante), cela signifie que le marquage courant tend vers le marquage désiré. Dans le cas contraire, le système oscille ou diverge.

$$J_M(\tau_k) < J_M(\tau_{k-1}) \quad (\text{III.30})$$

Par conséquent, en se basant sur (III.30), la fonction V_L , définie par (III.31), permet de détecter la présence des oscillations :

$$V_L(\tau_k) = \begin{cases} 1, & \text{si l'inégalité (III.30) est vérifiée} \\ 0, & \text{sinon} \end{cases} \quad (\text{III.31})$$

La deuxième phase de la méthode correspond à la correction en déterminant la valeur convenable de $r_{dX_c}(\tau_{k+1})$. Une solution consiste à modifier r_{dX_c} selon deux paramètres

multiplicatifs μ_+ et μ_- , respectivement pour augmenter ou diminuer la valeur de r_{dX_c} . Cette opération dépend de la vérification de la condition (III.30) et de la valeur de la variation de flux à la date τ_k . Cette idée présente l'intérêt d'avoir, d'une part, une variation douce de r_{dX_c} et d'autre part, d'affecter à r_{dX_c} des petites valeurs non-nulles, ce qui garantit la prise en compte de la variation de flux sur toute la trajectoire :

$$r_{dX_c}(\tau_{k+1}) = \begin{cases} \mu_+ \cdot r_{dX_c}(\tau_k), & \text{si } (V_L(\tau_k) = 1) \text{ et } (\|A_{acc}(\tau_k)\|^2 > \|A_{acc}(\tau_{k-1})\|^2) \\ \mu_- \cdot r_{dX_c}(\tau_k), & \text{si } (V_L(\tau_k) = 0) \\ r_{dX_c}(\tau_k), & \text{sinon} \end{cases} \quad (\text{III.32})$$

Cette adaptation permet de réduire les oscillations mais ne garantit pas la stabilité du système au voisinage de la configuration désirée. Pour cette raison, une étude de stabilité est proposée dans le paragraphe suivant.

III.3.2 Étude de la stabilité asymptotique

Commençons par définir la stabilité asymptotique dans le cas général. La stabilité de Lyapunov implique que tout mouvement issu d'un voisinage proche du point d'équilibre reste proche de ce point. La stabilité asymptotique inclut la stabilité de Lyapunov, mais exige en plus que le mouvement converge vers le point d'équilibre : un système est asymptotiquement stable ssi il a toujours tendance à revenir à sa position d'équilibre.

Dans le cas de la commande prédictive, la stabilité asymptotique peut être obtenue suite à l'ajout d'une contrainte terminale [MRRS00]. Quelques méthodes qui assurent la convergence du RdPC vers la configuration désirée ont notamment été présentées dans [MGR⁺08] : en particulier, forcer $M_c(\tau_{k+N_c})$ à appartenir à la droite qui joint $M_c(\tau_k)$ et M_{refc} . Analytiquement, la contrainte terminale équivalente est exprimée par (III.33) :

$$M_c(\tau_{k+N_c}) = M_c(\tau_k) + \alpha \cdot (M_{refc} - M_c(\tau_k)) \text{ tel que } 0 < \alpha \leq 1 \quad (\text{III.33})$$

Dans [MGR⁺08], il a été mentionné que la stabilité asymptotique, à l'aide de cette contrainte, ne peut être démontrée que pour $N_c = 1$. De plus, elle est imposée tout au long de la trajectoire. Cependant, dans le cas de la commande prédictive proposée, N_c doit pouvoir augmenter pour pallier le problème du "hill-climbing". De ce fait, cette contrainte terminale ne peut être directement appliquée. Par conséquent, la commande prédictive doit subir quelques modifications pour pouvoir appliquer une contrainte terminale qui garantira la stabilité.

Soit δ – *voisinage* une zone pour laquelle les marquages vérifient : $J_M \leq \delta$ avec $\delta > 0$. Dans ce voisinage, si J décroît, il n'est plus nécessaire de maintenir $N_c > 1$. De ce

fait, N_c devrait diminuer. Une fois que $N_c = 1$, le marquage atteint un β – voisinage ($\beta \leq \delta$). Uniquement dans cette dernière phase de la trajectoire la contrainte linéaire est introduite. De plus, à ce stade, r_{dX_c} n'est plus adapté, il prend une valeur fixe notée μ_0 . Les algorithmes 1 et 2 sont légèrement modifiés afin d'introduire l'adaptation de r_{dX_c} , la décrémentation de N_c et l'application de la contrainte terminale. Le paramètre CT (contrainte terminale) indique si la contrainte terminale sera utilisée pour calculer $U_c(\tau_k)$ ou non. Les nouvelles versions des algorithmes sont respectivement données par les algorithmes 3 et 4.

Algorithme 3 : Commande prédictive constante avec limitation de la variation du flux (Entrées : R_k , $M_c(\tau_k)$, N_c , N_{\max} , CT et $J(\tau_k)$; Sorties : $U_c(\tau_k)$ et N_c)

```

1: Initialisation des paramètres :
    $solution \leftarrow 0$ 
    $N_{limit}(\tau_k) \leftarrow N_{\max}$ 
2: Tant que  $solution = 0$ 
3:    $solution \leftarrow 1$ 
4:   Si  $CT = 0$ , alors
5:     Calculer  $U_c(\tau_k)$ , solution du problème d'optimisation (III.22)
     sous les contraintes (III.24)
6:      $e \leftarrow 1$ 
7:     Tant que  $e < N_c$ 
8:       Déterminer la région  $R_c(M_c(\tau_{k+e}))$  du marquage  $M_c(\tau_{k+e})$ 
9:       Si  $R_c(M_c(\tau_{k+e})) \neq R_k$ , alors
10:         $N_{limit}(\tau_k) \leftarrow e$ 
11:         $N_c \leftarrow e$ 
12:         $solution \leftarrow 0$ 
13:       Sinon
14:         $e \leftarrow e + 1$ 
15:       Fin si
16:     Fin tant que
17:     Si  $solution = 1$ , alors
18:       Calculer le critère  $J(\tau_{k+N_c})$ 
19:       Si  $(J(\tau_k) - J(\tau_{k+N_c}) \leq 0)$  et  $(N_c < N_{limit}(\tau_k))$ , alors
20:         $N_c \leftarrow N_c + 1$ 
21:         $solution \leftarrow 0$ 
22:       Fin si
23:     Fin si
24:     Calculer  $U_c(\tau_k)$ , solution du problème d'optimisation (III.22)
     sous les contraintes (III.24) et (III.34)
25:   Fin si
26: Fin tant que
27: Retourner  $U_c(\tau_k)$  et  $N_c$ 

```

Algorithme 4 : Commande du θ -RdPC via la commande prédictive avec limitation de la variation du flux (Entrées : M_{0c} , μ_0 et N_0 ; Sorties : U_c , M_c , X_c et J)

- 1: Initialisation des paramètres :
 - $N_c \leftarrow N_0$
 - $fin \leftarrow 0$
 - $TC \leftarrow 0$
 - $\tau_k \leftarrow 0$
 - $M_c(\tau_k) \leftarrow M_{0c}$
 - $U_c \leftarrow \emptyset$
 - 2: Calculer $X_c(\tau_k)$ et $J(\tau_k)$ et
 - 3: $M_c \leftarrow M_c(\tau_k)$, $X_c \leftarrow X_c(\tau_k)$, $J \leftarrow J(\tau_k)$
 - 4: **Tant que** $fin = 0$
 - 5: Déterminer R_k
 - 6: Calculer la commande $U_c(\tau_k)$ et déterminer N_c par le biais de l'algorithme 3
 - 7: Calculer $X(\tau_k)$, $M_c(\tau_{k+1})$ et $J(\tau_{k+1})$
 - 8: Calculer $rdX_c(\tau_{k+1})$
 - 9: $M_c \leftarrow M_c \cup M_c(\tau_k)$
 - 10: $X_c \leftarrow X_c \cup X_c(\tau_k)$
 - 11: $J \leftarrow J \cup J(\tau_k)$
 - 12: $U_c \leftarrow U_c \cup \emptyset$
 - 13: **Si** $J(\tau_{k+1}) = 0$, **alors**
 - 14: $fin \leftarrow 1$
 - 15: **Sinon**
 - 16: **Si** $J_M(\tau_{k+1}) \leq \delta$, **alors**
 - 17: **Si** $J(\tau_{k+1}) \leq J(\tau_k)$, **alors**
 - 18: $N_c \leftarrow \max(N_c - 1, 1)$
 - 19: **Fin si**
 - 20: **Si** $N_c = 1$, **alors**
 - 21: $CT \leftarrow 1$
 - 22: $rdX_c(\tau_{k+1}) \leftarrow \mu_0$
 - 23: **Fin si**
 - 24: $\tau_k \leftarrow \tau_{k+1}$
 - 25: **Fin si**
 - 26: **Fin tant que**
 - 27: Retourner U_c , M_c , X_c et J
-

Ayant à l'esprit l'expression de $M_c(\tau_{k+1})$ donnée par (II.4), l'équation (III.33) conduit à (III.34) :

$$\theta \cdot W \cdot U_c(\tau_k) + \alpha \cdot (M_{refc} - M_c(\tau_k)) = (B_k - I_n) \cdot M_c(\tau_k) \text{ tel que } 0 < \alpha \leq 1 \quad (\text{III.34})$$

Grâce à l'introduction de la contrainte terminale, la stabilité asymptotique peut être démontrée.

Soit Φ une fonction continue : $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}$, Φ est une fonction de Lyapunov si les conditions suivantes sont satisfaites :

- 1- $\Phi(M)$ est définie positive (c-à-d si $M = 0_{n \times 1}$, $\Phi(M) = 0$ et si $M \neq 0_{n \times 1}$, $\Phi(M) > 0$).
- 2- $\Phi(M(\tau_{k+1})) - \Phi(M(\tau_k)) \leq 0, \forall k \in \mathbb{N}$.

Si cette équation est satisfaite, le système est stable. De plus, si $\Phi(M)$ est strictement décroissante (c-à-d $\Phi(M(\tau_{k+1})) - \Phi(M(\tau_k)) < 0$), le système est asymptotiquement stable.

Proposition III.3.1. *Soit $\langle \mathcal{N}_c, M_0, \lambda, \theta, T_c^{comm} \rangle$ un θ -RdPC tel que θ vérifie la condition (I.35) et que $T_c^{comm} = T$. Supposons que $M_{refc} > 0_{n \times 1}$ soit atteignable à partir de $M_{0c} > 0_{n \times 1}$ et qu'il existe un instant τ_k à partir duquel le β - voisinage est atteint, alors le système en boucle fermée est asymptotiquement stable.*

Démonstration. Si $N_c = 1$, la contrainte terminale est appliquée. L'équation (III.33) est réécrite sous la forme $M_c(\tau_{k+1}) - M_{refc} = M_c(\tau_k) + \alpha \cdot (M_{refc} - M_c(\tau_k)) - M_{refc}$ ce qui implique $M_c(\tau_{k+1}) - M_{refc} = M_c(\tau_k) + (1 - \alpha) \cdot (M_c(\tau_k) - M_{refc})$. En conséquence, $J_M(\tau_{k+1})$ est reformulé suivant (III.35) :

$$\begin{aligned} J_M(\tau_{k+1}) &= (1 - \alpha) \cdot (M_c(\tau_k) - M_{refc})^t \cdot Q_{M_c} \cdot (1 - \alpha) \cdot (M_c(\tau_k) - M_{refc}) \\ &= (1 - \alpha)^2 \cdot (M_c(\tau_k) - M_{refc})^t \cdot Q_{M_c} \cdot (M_c(\tau_k) - M_{refc}) \\ &= (1 - \alpha)^2 \cdot J_M(\tau_k) \end{aligned} \quad (\text{III.35})$$

Dans la mesure où $0 < \alpha \leq 1$, J_M peut être utilisé comme une fonction de Lyapunov, puisqu'il est défini positif et strictement décroissant ($(1 - \alpha)^2 < 1$). \square

III.3.3 Exemple illustratif 3

Considérons à nouveau le RdPC donnée par la figure III.11 auquel l'algorithme 4 est appliqué. Rappelons que la première configuration désirée est $(M_{refc1} = \begin{bmatrix} 0.2 & 0.6 & 0.8 & 0.4 \end{bmatrix}^t, X_{refc1} = \begin{bmatrix} 0.2 & 0.2 & 0.2 \end{bmatrix}^t)$ et qu'à la date $\tau_k = 3.5$ T.U., la cible est remplacée par $(M_{refc2} = \begin{bmatrix} 0.3 & 0.3 & 0.7 & 0.7 \end{bmatrix}^t, X_{refc2} = \begin{bmatrix} 0.3 & 0.3 & 0.3 \end{bmatrix}^t)$.

Les paramètres de simulations sont ceux de l'exemple III.2.2. Les nouveaux paramètres sont $\mu_0 = 0.1$, $\mu_+ = 1.1$ et $\mu_- = 0.9$.

Les figures III.16 et III.17 présentent respectivement l'évolution de la valeur adaptative de r_{dX_c} et de l'horizon de prédiction en fonction du temps.

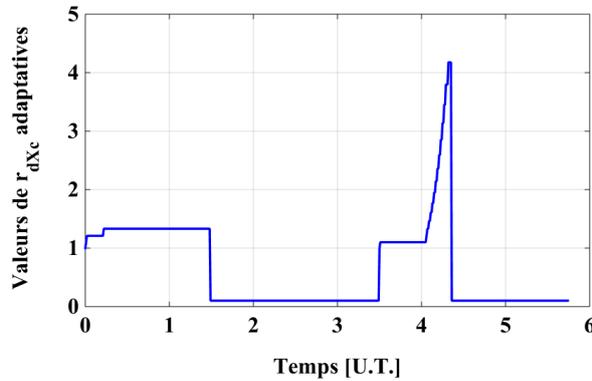


FIGURE III.16 – Évolution temporelle de r_{dX_c} adapté avec (III.32)

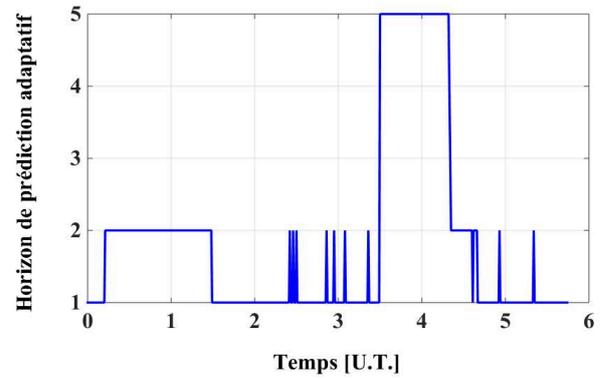


FIGURE III.17 – Évolution temporelle de l'horizon de prédiction N_c

La figure III.18 présente une comparaison entre le comportement du système lorsque $r_{dX_c} = 1$ et r_{dX_c} est adaptatif.

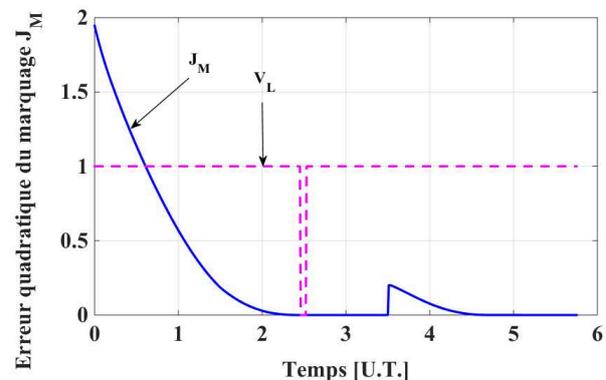
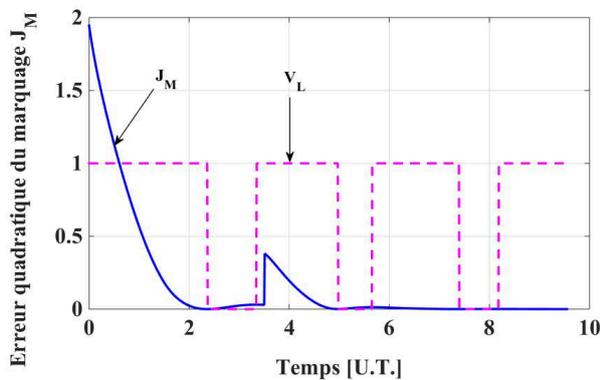


FIGURE III.18 – Évolution de l'erreur quadratique du marquage lorsque $r_{dX_c} = 1$ (gauche) et r_{dX_c} est adaptatif (droite)

Comme mentionné précédemment, à cause des oscillations qui surviennent autour de la configuration désirée suite à une valeur constante non-nulle de r_{dX_c} , la condition (III.30) n'est pas vérifiée durant les intervalles pour lesquels le marquage s'éloigne temporairement du marquage désiré (figure III.18 (gauche)). Grâce à la méthode proposée, les oscillations sont presque éliminées (les oscillations qui restent sont pratiquement négligeables en durée et en amplitude par rapport à celles obtenues pour $r_{dX_c} = 1$).

La figure III.19 montre l'évolution du marquage lorsque r_{dX_c} est adaptatif.

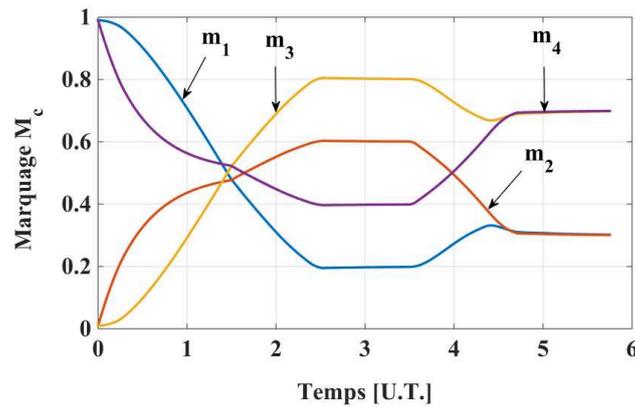


FIGURE III.19 – Évolution du marquage lorsque r_{dX_c} est adaptatif

L'accélération maximale et la valeur de A_C restent également faibles. Elles sont respectivement de l'ordre de 11.17 et 4.7. Par conséquent, par rapport à l'optimisation sans r_{dX_c} , elles ont diminué d'environ 70%.

La trajectoire du marquage projetée dans le plan (m_1, m_2) est donnée par la figure III.20. Une fois que le marquage atteint le β -voisinage, la contrainte terminale est appliquée ce qui explique bien la trajectoire rectiligne du marquage au voisinage du marquage désiré (figure III.20 (droite)).

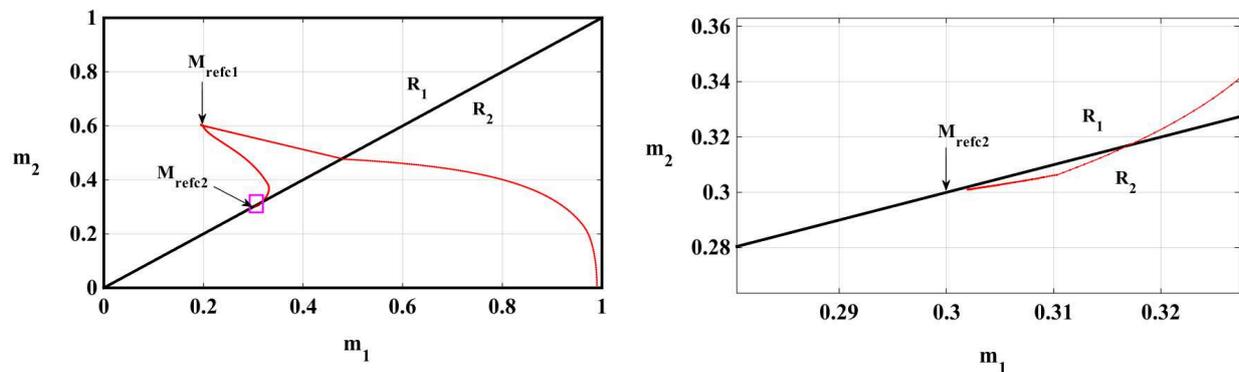


FIGURE III.20 – Trajectoire du marquage continu dans le plan de projection (m_1, m_2) lorsque r_{dX_c} est adaptatif

D'après ces résultats, nous pouvons dire que l'approche proposée assure que l'objectif est atteint avec moins de variations de flux.

III.4 Application de la commande prédictive aux réseaux de Petri discrets T-temporisés

Dans cette section nous montrons de quelle façon la commande proposée permet de commander des RdPD. La motivation de cette partie est de déterminer une séquence de commandes discrètes, qui permet à un RdPD T-temporisé d'atteindre un marquage désiré de durée quasi-minimale, sans passer par l'exploration du graphe d'atteignabilité.

Le principe de la méthode proposée est d'utiliser la fluidification pour commander le RdPD. Il s'agit de fluidifier un RdPD afin d'obtenir un θ -RdPC, commander ce dernier par le biais de la commande prédictive et déterminer la séquence de commandes discrètes à partir de la commande continue obtenue. Cette approche est adaptée pour résoudre les problèmes d'ordonnancement, en particulier lorsque l'optimisation des temps de cycle est concernée [KSN07].

III.4.1 Principe de la commande prédictive du RdPD T-temporisé

Cette méthode de commande concerne uniquement les RdPD T-temporisés avec des transitions non-immédiates ($d_{min\ j} \neq 0, j = 1, \dots, q$) et sans blocage. Toutes les transitions discrètes sont supposées être commandables. Notons que dans cette partie la limitation de la variation du flux n'est pas prise en compte tandis que la contrainte terminale est bien considérée.

Soit $\langle \mathcal{N}_d, D_{min}, M_{0d} \rangle$ un RdPD T-temporisé et $\langle \mathcal{N}_c, M_{0c}, \lambda, \theta \rangle$ le θ -RdPC correspondant. Les deux réseaux possèdent la même structure statique $\mathcal{N}_d = \mathcal{N}_c$, le même marquage initial $M_{0d} = M_{0c} \in \mathbb{N}_{\geq 0}^n$ et les mêmes spécifications temporelles $\lambda_j = \frac{1}{d_{min\ j}}, t_j \in T$. Supposons que le marquage désiré discret M_{refd} soit atteignable à partir de M_{0d} . L'idée de base est d'imposer le même objectif de commande au θ -RdPC, de commander ce dernier via la commande prédictive, proposée dans les sections précédentes, pour atteindre la configuration désirée (M_{refc}, X_{refc}) et de se référer à la commande continue générée pour déduire la séquence de commandes discrètes.

Le flux de franchissement permet de déterminer la quantité de jetons qui traverse chaque transition. Soit $X_{cum}(\tau_k) = (x_{cum\ j}(\tau_k))_{j=1, \dots, q}$ le flux cumulé, jusqu'à la date τ_k , multiplié par θ . Ce dernier représente le nombre de jetons qui se sont déplacés, jusqu'à la date τ_k , à travers chaque transition continue t_j . Il est défini par (III.36) :

$$X_{cum}(\tau_k) = \theta \cdot \sum_{s=0}^k X_c(\tau_s) \quad (\text{III.36})$$

Analytiquement, la commande du RdPD est décrite dans l'algorithme 6. Elle consiste à déterminer, à chaque date τ_k , l'action de commande $U_c(\tau_k)$ à l'aide de l'algorithme 1, puis à calculer $X_c(\tau_k)$ et $X_{cum}(\tau_k)$. Dès que $X_{cum}(\tau_k)$ atteint la valeur 1 pour une transitions continue t_j , un jeton au moins a traversé la même transition discrète t_j dans le RdPD. Deux cas peuvent se distinguer ; soit t_j n'est pas validée par le marquage discret courant et la méthode échoue, soit t_j est validée. Après avoir franchi cette transition, il faut vérifier si le franchissement de cette transition, parmi les autres validées, conduit au marquage discret le plus proche du marquage continu obtenu à cet instant. Pour ce faire, tous les marquages discrets, qui résultent du franchissement de toutes les transitions discrètes validées, sont calculés. L'écart quadratique $\Phi(M_d)$, défini par (III.37), est calculé pour chaque marquage discret M_d obtenu.

$$\Phi(M_d) = (M_d - M_c)^t \cdot (M_d - M_c) \quad (\text{III.37})$$

Soit t^{opt} la transition discrète qui minimise Φ . Deux autres cas peuvent se présenter : si t^{opt} correspond à la transition t_j sélectionnée, alors seule la composante $x_{cum\ j}(\tau_k)$ est remise à zéro. Sinon, la trajectoire continue s'arrête au niveau de M_c , le RdPC est réinitialisé avec M_d (c-à-d $M_{0c} = M_d$) et le vecteur $X_{cum}(\tau_k)$ est remis à zéro.

Il convient de mentionner que, parfois, il y a plus d'une transition pour lesquelles $X_{cum}(\tau_k)$ atteint 1. La transition franchie est celle validée et qui minimise $\Phi(M_d)$ par rapport aux autres sélectionnées.

En conclusion, l'approche ainsi décrite fait en sorte que le marquage discret soit aussi proche que possible de la trajectoire continue, comme détaillée dans la proposition III.4.1.

Notons que les dates de franchissement des transitions discrètes $t_j \in \sigma_d$ dans le RdPD T-temporisé sont déterminées en ligne au fur et à mesure que la transition à franchir est sélectionnée. Ces dates sont calculées de telle sorte que le franchissement des transitions discrètes a lieu le plus tôt possible, tout en respectant les contraintes temporelles discrètes données par les durées minimales de franchissement. La sous-routine qui calcule ces dates est détaillée dans l'algorithme 5. Cet algorithme est basé sur la construction d'un échancier $DCH(h)$, $h = 1, \dots, H$. Ce dernier associe, à chaque transition validée par $M_d^{tot}(h-1)$, sa date de franchissement au plus tôt notée $date(t_j)$. Puis, $DCH(h)$ est mis à jour.

- Le couple $(\sigma_d(h), date(\sigma_d(h)))$ est supprimé de $DCH(h)$.
- Les transitions mémorisées dans $DCH(h)$ mais qui sont invalidées par $M_d^{tot}(h-1)$ sont également retirées de $DCH(h)$.
- Les dates de franchissement des transitions t_j retenues dans $DCH(h)$ sont mises à

jour : $date(t_j) \leftarrow \max(date(\sigma_d(h)), date(t_j))$.

- Les dates de franchissement des transitions t_j nouvellement validées sont calculées et mémorisées dans $DCH(h)$: $date(t_j) \leftarrow date(\sigma_d(h)) + d_{min\ j}$.

L'échéancier DCH est construit à l'étape initiale (c-à-d $h = 0$) en affectant à chaque transition validée par M_{0d} , sa durée minimale de franchissement et le nombre d'occurrences d'une transition t_j dans l'échéancier dépend de son degré de validation.

Algorithme 5 : Calcul de la date de franchissement au plus tôt des transitions discrètes dans le RdPD (Entrées : $DCH(h-1)$, $M_d^{tot}(h-1)$ et $\sigma_d(h)$; Sortie : $date(\sigma_d(h))$)

- 1: Déterminer, dans $DCH(h-1)$, $date(\sigma_d(h))$
 - 2: $DCH(h) \leftarrow DCH(h-1)$
 - 3: Supprimer $(\sigma_d(h), date(\sigma_d(h)))$ de $DCH(h)$
 - 4: Supprimer les transitions discrètes qui ne sont plus validées par $M_d^{tot}(h)$ de $DCH(h)$
 - 5: Mettre à jour la date des transitions discrètes déjà présentes dans $DCH(h)$
 - 6: Déterminer l'ensemble des transitions validées par $M_d^{tot}(h)$
 - 7: Calculer la date de franchissement des transitions nouvellement validées et les mémoriser dans $DCH(h)$
 - 8: Retourner $date(\sigma_d(h))$
-

Les dates de franchissement calculées sont mémorisées dans une séquence temporelle notée σ_d^{temps} . L'algorithme 6 donne alors en sortie deux séquences de commande σ_d^{temps} et σ_d de longueur $H = |\sigma_d|$ ainsi que deux séquences M_d^{tot} et M_c^{tot} des marquages discrets et continus.

Remarque 3. La séquence de commandes discrètes σ_d conduit, la plupart du temps, au marquage désiré. En effet, $\forall h = 1, \dots, H$, le marquage continu peut s'écrire sous la forme $M_c^{tot}(h) = M_{0c} + W \cdot X_{cum}$ où M_{0c} (ou M_{0d}) et X_{cum} représentent respectivement le dernier marquage qui a initialisé la trajectoire continue et la somme cumulée depuis la dernière réinitialisation. X_{cum} peut être exprimé par : $X_{cum} = \lfloor X_{cum} \rfloor + \{X_{cum}\}$ tel que $\lfloor \cdot \rfloor$ et $\{ \cdot \}$ désignent respectivement les parties entière et décimale de (\cdot) . Notez que $\lfloor X_{cum} \rfloor$ peut être vu comme le vecteur de franchissements des transitions dans le RdPD. Par conséquent, les expressions suivantes sont obtenues : $M_d^{tot}(h) = M_{0d} + W \cdot \lfloor X_{cum} \rfloor = M_c^{tot}(h) - W \cdot \{X_{cum}\}$. Lorsque le θ -RdPC atteint la configuration désirée, $M_c^{tot}(h) = M_{refc}$, $\{X_{cum}\} = X_{refc}$ et par la suite l'équation fondamentale du RdPD est donnée par : $M_d^{tot}(h) = M_{refc} - W \cdot X_{refc}$. Ainsi, d'après la définition de X_{refc} , $W \cdot \{X_{cum}\} = 0_{n \times 1}$. Par conséquent, $M_d^{tot}(h) = M_{refc}$.

Algorithme 6 : Commande du RdPD via la commande prédictive constante (Entrées : M_{0d} et N_0 ; Sorties : σ_d , σ_d^{temps} , H , M_d^{tot} et M_c^{tot})

```

1: Initialisation des paramètres :
    $N_c \leftarrow N_0, \tau_k \leftarrow 0, h \leftarrow 0, M_c(\tau_k) \leftarrow M_{0d}, X_{cum}(\tau_k) \leftarrow 0_{q \times 1}, \sigma_d \leftarrow \emptyset, \sigma_d^{temps} \leftarrow \emptyset,$ 
    $N_c \leftarrow N_0, drap \leftarrow 1, fin \leftarrow 1, M_d^{tot}(h) \leftarrow M_{0d}, M_c^{tot}(h) \leftarrow M_{0d}, CT \leftarrow 1$ 
2: Construire l'échéancier  $DCH(h)$  et calculer  $X_c(\tau_k)$  et  $J(\tau_k)$ 
3: Tant que  $fin = 0$  et  $drap = 1$ 
4:   Calculer la commande  $U_c(\tau_k)$  par le biais de l'algorithme 4
5:   Calculer  $X_c(\tau_k), X_{cum}(\tau_k), M_c(\tau_{k+1})$  et  $J(\tau_{k+1})$ 
6:   Si  $\exists j$  tel que  $x_{cum\ j}(\tau_k) \geq 1$ , alors
7:      $h \leftarrow h + 1$ 
8:     Si  $M_d^{tot}(h - 1)[t_j >$ , alors
9:       Calculer le marquage discret  $M_d$ 
10:       $M_d^{tot} \leftarrow M_d^{tot} \cup M_d$ 
11:       $drap \leftarrow 1$ 
12:      Déterminer  $t^{opt}$ 
13:      Si  $t_j = t^{opt}$ , alors
14:         $x_{cum\ j} \leftarrow 0$ 
15:         $M_c^{tot} \leftarrow M_c^{tot} \cup M_c(\tau_{k+1})$ 
16:      Sinon
17:        Réinitialiser la trajectoire continue par  $M_d^{tot}(h)$ 
18:         $M_c^{tot} \leftarrow M_c^{tot} \cup M_d^{tot}(h)$ 
19:         $X_{cum}(\tau_k) \leftarrow 0_{q \times 1}$ 
20:      Fin si
21:    Sinon
22:       $drap \leftarrow 0$ 
23:    Fin si
24:  Fin si
25:   $\sigma_d \leftarrow \sigma_d \cup t_j$ 
26:  Déterminer  $date(\sigma_d(h))$  à l'aide de l'algorithme 5
27:   $\sigma_d^{temps} \leftarrow \sigma_d^{temps} \cup (\sigma_d(h), date(\sigma_d(h)))$ 
28:  Calculer  $J(\tau_{k+1})$ 
29:  Si  $J(\tau_{k+1}) = 0$ , alors
30:     $fin \leftarrow 1$ 
31:  Sinon
32:     $\tau_k \leftarrow \tau_{k+1}$ 
33:    Si  $J_M(\tau_{k+1}) \leq \delta$ , alors
34:      Si  $J(\tau_{k+1}) < J(\tau_k)$ , alors
35:         $N_c \leftarrow \max(N_c - 1, 1)$ 
36:      Fin si
37:      Si  $N_c = 1$ , alors
38:         $CT \leftarrow 1$ 
39:      Fin si
40:    Fin si
41:  Fin si
42: Fin tant que
43: Si  $drap = 1$ , alors
44:    $H \leftarrow h$ 
45: Fin si

```

La proposition III.4.1 garantit la proximité entre la trajectoire discrète et la trajectoire continue.

Proposition III.4.1. *Soit $\langle \mathcal{N}_d, M_{0d}, D_{min}, T_d^{comm} \rangle$ un RdPD T-temporisé sans blocage tel que toutes les transitions soient commandables ($T_d^{comm} = T$) et non-immédiates et soit $\langle \mathcal{N}_c, M_{0c}, \lambda, \theta, T_c^{comm} \rangle$ le θ -RdPC correspondant tel que $T_c^{comm} = T$. Supposons que M_{refd} soit atteignable en discret et que (M_{refd}, X_{refc}) soit la configuration désirée pour le θ -RdPC. Si l'algorithme 6 retourne une séquence M_d^{tot} de marquages discrets qui convergent vers M_{refd} , alors les propriétés suivantes sont obtenues :*

- $\forall h = 1, \dots, H$, $M_d^{tot}(h)$ est atteignable à partir de $M_d^{tot}(h-1)$ et la trajectoire discrète $M_{0d}[\sigma_d(1) > M_d^{tot}(1), \dots, M_d^{tot}(H-1)[\sigma_d(H) > M_d^{tot}(H)]$ est faisable à partir de M_{0d} .
- $\forall h = 1, \dots, H$, $M_d^{tot}(h)$ est le marquage discret le plus proche du marquage continu.

Démonstration.

- $\forall h = 1, \dots, H$, $\sigma_d(h)$ est validée par $M_d^{tot}(h-1)$. Par conséquent, la trajectoire discrète est faisable par construction.
- Selon l'algorithme 6, $\sigma_d(h)$ correspond à la transition t_j pour laquelle $x_{cumj} \geq 1$. Si t_j minimise Φ ($t_j = t^{opt}$), alors la propriété est vérifiée. Dans le cas contraire, la trajectoire continue est réinitialisée par $M_d^{tot}(h)$ et par conséquent, la propriété est également satisfaite.

□

Un exemple explicatif permet de mettre en valeur les résultats obtenus.

III.4.2 Exemple illustratif 4

Afin d'illustrer la méthode proposée, considérons le RdPD T-temporisé présenté dans la figure III.21 avec $M_{0d} = \begin{bmatrix} 6 & 4 & 8 & 2 \end{bmatrix}^t$, $D_{min} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^t$. Par conséquent, dans le RdPC correspondant, λ est défini par $\lambda = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^t$. Les simulations numériques ont été effectuées telles que : $M_{refd} = \begin{bmatrix} 6 & 8 & 4 & 18 \end{bmatrix}^t$, $X_{refc} = \begin{bmatrix} 3 & 3 & 3 \end{bmatrix}^t$, $r_{M_c} = 1$, $r_{X_c} = 0.1$, $\theta = 0.1$ T.U., $N_0 = 1$ et $N_{max} = 10$. Notons que X_{refc} est choisi tel que la configuration continue désirée soit un point d'équilibre, il vérifie la condition (II.9).

L'évolution de X_{cum} et de l'erreur quadratique du marquage continu sont respectivement présentées dans les figures III.22 et III.23.

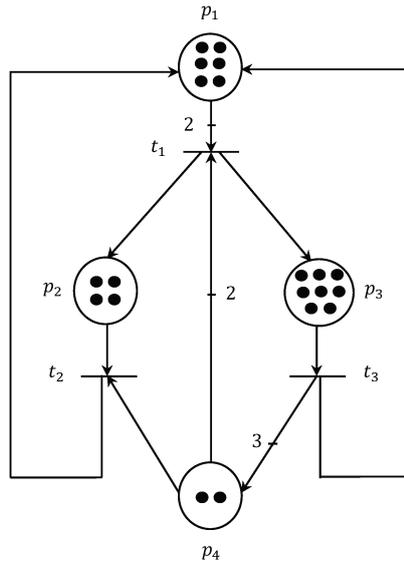


FIGURE III.21 – Exemple d’un RdPD T-temporisé sans blocage

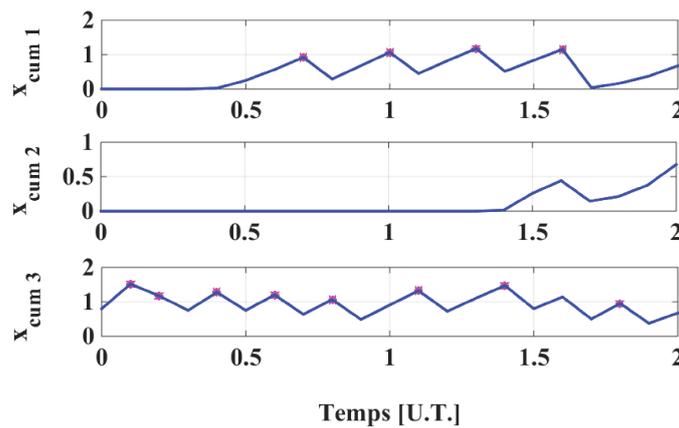


FIGURE III.22 – Évolution temporelle du flux cumulé X_{cum}

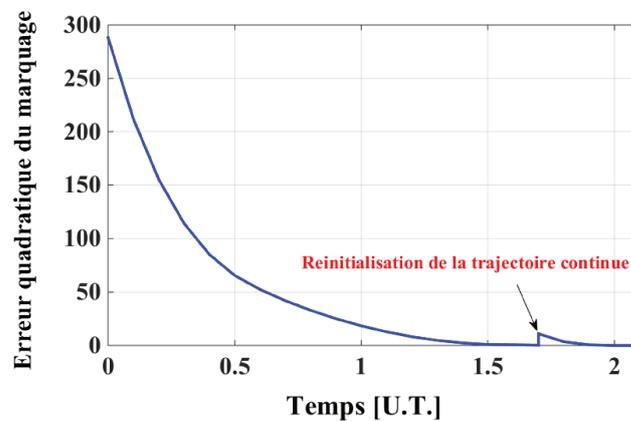


FIGURE III.23 – Évolution temporelle de l’erreur quadratique du marquage continu J_M

La figure III.24 récapitule les résultats obtenus. Elle présente les trajectoires du marquage continu et discret qui sont respectivement représentées par des lignes pleines et des traits discontinus. En outre, elle illustre la séquence de commandes temporisée identifiée par des disques.

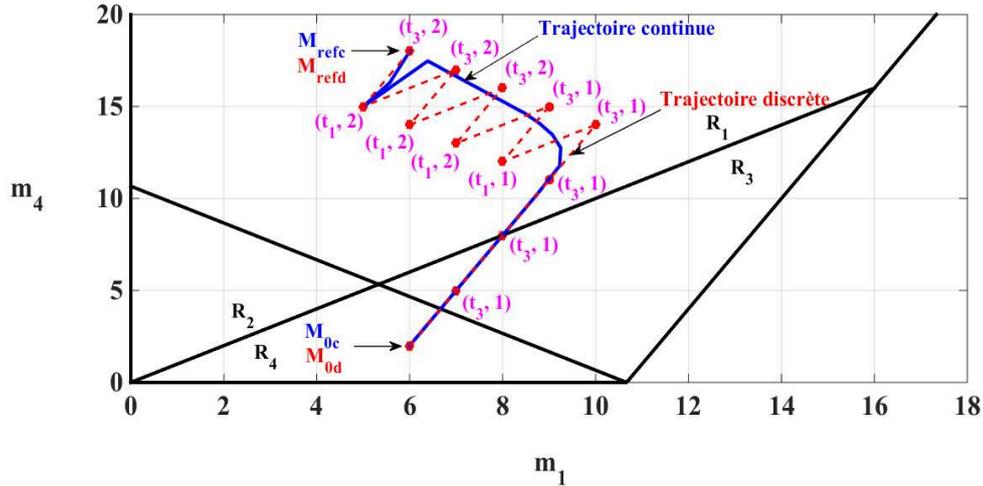


FIGURE III.24 – Trajectoire du marquage continu et discret dans le plan de projection (m_1, m_4)

Nous remarquons que la trajectoire continue a été réinitialisée une fois, ce qui explique la discontinuité de l'écart quadratique du marquage (voir figure III.23 à la date $\tau_k = 1.7$ U.T).

Le processus de détermination de σ_d , pour cet exemple, est le suivant :

Tout d'abord, puisque $\eta_d(t_1, M_{0d}) = 1$, $\eta_d(t_2, M_{0d}) = 2$ et $\eta_d(t_3, M_{0d}) = 8$, $DCH(0)$ est construit comme suit : $DCH(0) = \{(t_1, 1)(t_2, 1)(t_2, 1)(t_3, 1)(t_3, 1)(t_3, 1)(t_3, 1)(t_3, 1)(t_3, 1)(t_3, 1)(t_3, 1)\}$. À la date $\tau_k = 0.1$ U.T., $x_{cum\ 3} \approx 1.52$, alors t_3 est la première transition continue sélectionnée (voir figure III.22). Cette transition est validée dans le RdPD par le marquage M_{0d} . En conséquence, $h = 1$, $\sigma_d(h) = t_3$ et $M_d^{tot}(1) = \begin{bmatrix} 7 & 4 & 7 & 5 \end{bmatrix}^t$. De plus, t_3 minimise Φ (c-à-d $t^{opt} = t_3$), alors $x_{cum\ 3} \leftarrow 0$. D'après $DCH(0)$, la date de franchissement au plus tôt de t_3 est 1 U.T., alors $\sigma_d^{temps}(h) = (t_3, 1)$. Une fois t_3 franchie, $\eta_d(t_1, M_d^{tot}(1)) = 2$, $\eta_d(t_2, M_d^{tot}(1)) = 2$ et $\eta_d(t_3, M_d^{tot}(1)) = 7$, la transition t_1 est nouvellement validée, donc la transition t_1 apparaîtra deux fois dans $DCH(1) = \{(t_1, 1)(t_1, 2)(t_2, 1)(t_2, 1)(t_3, 1)(t_3, 1)(t_3, 1)(t_3, 1)(t_3, 1)(t_3, 1)\}$. La même procédure est répétée aux étapes suivantes. Pendant la 11^{ème} étape, t_1 et t_3 sont sélectionnées ($x_{cum\ 1} \approx 1$ et $x_{cum\ 3} \approx 1$) et toutes les deux sont validées par $M_d^{tot}(10) = \begin{bmatrix} 7 & 7 & 4 & 17 \end{bmatrix}^t$. En revanche, puisqu'on ne peut franchir qu'une seule transition à la fois, la transition t_1 est choisie car elle présente un coût minimal de Φ par rapport à t_3 . D'un autre côté, cette transition ne minimise pas Φ (celle qui minimise ce coût est t_2 , transition dis-

crète qui permet d'obtenir le marquage discret le plus proche de la trajectoire continue). Pour cette raison, la trajectoire continue est réinitialisée par le marquage discret $M_d^{tot}(11) = \begin{bmatrix} 5 & 8 & 5 & 15 \end{bmatrix}^t$ qui résulte du franchissement de t_1 (voir figures III.23 et III.24), $\sigma_d(11) = t_1$, X_{cum} est remis à zéro pour toutes les transitions. Sachant que $DCH(10) = \{(t_1, 2)(t_1, 2)(t_1, 3)(t_2, 2)(t_2, 2)(t_2, 2)(t_2, 2)(t_2, 2)(t_2, 3)(t_2, 3)(t_3, 2)(t_3, 2)(t_3, 3)(t_3, 3)\}$. Alors, $\sigma_d^{temps}(11) = (t_1, 2)$. Le problème d'optimisation est répété. À la fin de la simulation, les séquences de commande discrètes non-temporisée et temporisée sont respectivement données par $\sigma_d = t_3 t_3 t_3 t_3 t_1 t_3 t_1 t_3 t_1 t_3 t_1 t_3$ et $\sigma_d^{temps} = (t_3, 1)(t_3, 1)(t_3, 1)(t_3, 1)(t_1, 1)(t_3, 1)(t_3, 1)(t_3, 1)(t_1, 2)(t_3, 2)(t_1, 2)(t_3, 2)(t_1, 2)(t_3, 2)\}$ telles que :

$$\begin{array}{cccccccc}
 \begin{bmatrix} 6 \\ 4 \\ 8 \\ 2 \end{bmatrix} & \xrightarrow{(t_{3,1})} & \begin{bmatrix} 7 \\ 4 \\ 7 \\ 5 \end{bmatrix} & \xrightarrow{(t_{3,1})} & \begin{bmatrix} 8 \\ 4 \\ 6 \\ 8 \end{bmatrix} & \xrightarrow{(t_{3,1})} & \begin{bmatrix} 9 \\ 4 \\ 5 \\ 11 \end{bmatrix} & \xrightarrow{(t_{3,1})} & \begin{bmatrix} 10 \\ 4 \\ 4 \\ 14 \end{bmatrix} & \xrightarrow{(t_{1,1})} & \begin{bmatrix} 8 \\ 5 \\ 5 \\ 12 \end{bmatrix} & \xrightarrow{(t_{3,1})} & \begin{bmatrix} 9 \\ 5 \\ 4 \\ 15 \end{bmatrix} & \xrightarrow{(t_{1,2})} & \begin{bmatrix} 7 \\ 6 \\ 5 \\ 13 \end{bmatrix} \\
 \begin{bmatrix} 7 \\ 6 \\ 5 \\ 13 \end{bmatrix} & \xrightarrow{(t_{3,2})} & \begin{bmatrix} 8 \\ 6 \\ 4 \\ 16 \end{bmatrix} & \xrightarrow{(t_{1,2})} & \begin{bmatrix} 6 \\ 7 \\ 5 \\ 14 \end{bmatrix} & \xrightarrow{(t_{3,2})} & \begin{bmatrix} 7 \\ 7 \\ 4 \\ 17 \end{bmatrix} & \xrightarrow{(t_{1,2})} & \begin{bmatrix} 5 \\ 8 \\ 5 \\ 15 \end{bmatrix} & \xrightarrow{(t_{3,2})} & \begin{bmatrix} 6 \\ 8 \\ 4 \\ 18 \end{bmatrix} & & & &
 \end{array} \quad (III.38)$$

Conclusion

Dans ce chapitre, une commande prédictive est proposée pour les RdPC en vue de limiter la complexité de calcul, de résoudre le problème de “hill-climbing” et de réduire la variation du flux. Quatre contributions ont été présentées. Dans un première temps, une commande prédictive est proposée. Cette commande présente les caractéristiques suivantes : au cours de la prédiction, la commande est constante, le marquage reste dans la même région et l'horizon de prédiction est adaptatif, il augmente tant que le critère de performance n'a pas convenablement baissé et il diminue dans le cas contraire. L'avantage principal de la commande proposée est sa complexité calculatoire réduite qui permet de l'utiliser pour des systèmes de grande taille et/ou avec des horizons de prédictions plus larges. Les résultats de simulation illustrent clairement que l'horizon de prédiction adaptatif permet de prédire suffisamment loin afin d'éviter le blocage du système lorsque ce dernier présente une phase de “hill-climbing”.

En dépit de son efficacité, cette commande peut engendrer une forte variation du flux qui se manifeste en pratique par une sollicitation excessive des actionneurs. Pour résoudre ce problème, dans la deuxième partie de ce chapitre, un nouveau terme qui prend en compte la variation du flux est ajouté au critère de performance. Les résultats de simula-

tions montrent l'importance de ce terme, mais des oscillations surviennent et par la suite un problème de stabilité se présente. Pour limiter cet inconvénient, une étude de stabilité a été menée ; une nouvelle formule est proposée pour calculer le facteur de pondération à chaque période d'échantillonnage et une contrainte terminale est ajoutée afin de garantir la stabilité asymptotique du système autour du marquage désiré.

Pour éviter l'exploration de l'espace d'atteignabilité, la dernière partie du chapitre présente une méthode qui consiste à déterminer la séquence de commandes discrètes du RdPD à partir du RdPC correspondant commandé via la commande prédictive proposée.

Le chapitre suivant s'inspire de la commande prédictive présentée dans ce chapitre afin de proposer une commande prédictive pour les RdPH.

CHAPITRE IV

COMMANDE PRÉDICTIVE DES RÉSEAUX DE PETRI HYBRIDES ÉLÉMENTAIRES

Sommaire

Introduction	116
IV.1 Commande prédictive hybride des RdpHE avec des transi- tions discrètes immédiates	117
IV.1.1 Principe général	117
IV.1.2 Commande prédictive discrète	118
IV.1.3 Commande prédictive continue	120
IV.1.4 Exemple illustratif 1	124
IV.1.5 Commande prédictive hybride avec limitation de la variation du flux	129
IV.1.6 Exemple illustratif 2	131
IV.2 Commande prédictive hybride des RdpHE avec des transi- tions discrètes temporisées	134
IV.2.1 Principe de la commande prédictive hybride simplifiée	135
IV.2.2 Exemple illustratif 3	137
IV.2.3 Principe de la commande prédictive hybride complète	141
IV.2.4 Exemple illustratif 4	147
Conclusion	149

Introduction

L'objectif de la commande des SDH (Systèmes dynamiques hybrides) modélisés à l'aide des RdPHE munis de transitions discrètes immédiates et/ou temporisées, consiste à déterminer une commande hybride dans le but d'atteindre un marquage désiré à partir d'un marquage initial. Étant donné que le RdPH combine à la fois des processus continus et discrets en interaction, la commande hybride doit être composée d'une commande continue et d'une commande discrète. Ces deux commandes étant interdépendantes. Dans le cadre de cette étude, la partie discrète est composée soit de transitions immédiates soit de transitions temporisées. Concernant ce dernier cas, la temporisation discrète doit être prise en compte dans le calcul de la commande continue ce qui rend plus délicat la détermination de la commande hybride. Par ailleurs, une sollicitation excessive des actionneurs continus risque d'être engendrée par la commande.

Pour pallier tous ces problèmes, le présent chapitre propose une commande prédictive hybride. Cette commande est inspirée de la commande prédictive et la commande prédictive avec limitation de la variation de flux présentée dans le deuxième et le troisième chapitres. Dans un premier temps, afin de commander le RdPHE et de limiter la variation du flux, une commande prédictive discrète suivie d'une commande continue vont être mises en œuvre en considérant uniquement des transitions discrètes immédiates. Par la suite, une extension de cette commande prendra en considération des transitions discrètes non-immédiates.

La section IV.1 présente le principe de la commande prédictive hybride proposée dans le cas du RdPHE (ou RdPHDE) temporisé avec des transitions discrètes immédiates. Cette méthode est mise en œuvre grâce à une proposition garantissant l'optimalité par rapport à la fonction coût choisie. Cette section souligne également l'importance de la limitation de la variation de flux et propose une solution qui permet de limiter la sollicitation excessive des actionneurs continus. Dans la section IV.2, deux extensions de la commande prédictive hybride sont établies afin de prendre en compte les transitions discrètes temporisées. La première ne retient que les séquences de commandes immédiatement franchissables à l'instant courant et la deuxième exige une prise en compte des franchissements discrets au cours de la prédiction continue. Une comparaison entre ces deux approches sera détaillée par le biais d'un exemple illustratif.

IV.1 Commande prédictive hybride des RdPHE avec des transitions discrètes immédiates

Dans cette partie, les transitions discrètes sont supposées être immédiates ($D_{min} = 0_{q_d \times 1}$). Une généralisation de cette méthode sera présentée ultérieurement dans ce chapitre.

IV.1.1 Principe général

Comme son nom l'indique, cette commande est inspirée de la commande prédictive (la commande prédictive standard et la commande continue constante durant l'horizon de prédiction présentée dans les chapitres précédents). Par ailleurs, puisqu'elle est destinée aux SDH qui combinent les deux processus continus et discrets, la stratégie proposée intègre deux étapes : une commande prédictive discrète (pour la partie discrète) et une autre continue (pour la partie continue). La prédiction de la partie continue du RdPHE est réalisée sur un horizon de N_c périodes d'échantillonnage ($N_c \in \mathbb{N}_{>0}$). Alors que la prédiction de la partie discrète est réalisée sur un horizon de N_d événements discrets intermédiaires ($N_d \in \mathbb{N}_{>0}$).

À chaque date τ_k , la commande prédictive discrète prédit le comportement de la partie discrète du RdPHE résultant d'un certain nombre de franchissements. Un ensemble fini de toutes les séquences de commandes discrètes réalisables est déterminé. Chaque séquence est composée d'un nombre limité de transitions discrètes immédiates validées. Pour chacune de ces séquences, la commande prédictive continue calcule la commande continue associée qui minimise un critère de performance donné. Finalement, un ensemble d'actions de commandes hybrides est obtenu. La combinaison hybride optimale sera sélectionnée parmi les commandes déterminées et sera appliquée au système à partir de l'état courant. La figure IV.1 donne un aperçu général de la commande prédictive hybride :

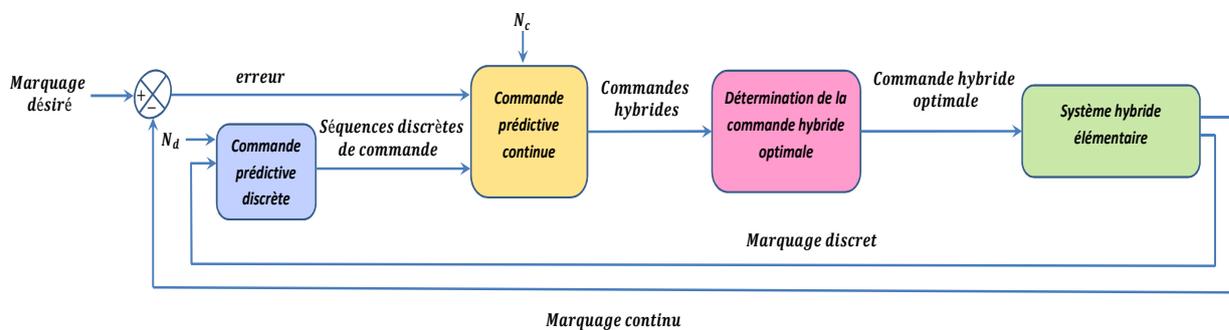


FIGURE IV.1 – Principe de la commande prédictive hybride

IV.1.2 Commande prédictive discrète

La commande discrète proposée dans cette sous-section repose principalement sur une commande prédictive discrète des RdPD non-temporisés présentée dans le deuxième chapitre. Elle s'appuie sur une exploration partielle du graphe d'atteignabilité du RdPD en vue de déterminer les séquences de commandes discrètes possibles de taille minimale ou quasi-minimale, méthode utilisée lors de la commande prédictive discrète dans [LL15]. La séquence de commandes discrètes, qui minimise un critère de performance discret (erreur de marquage discret), est retenue.

L'objectif de la commande prédictive discrète proposée est de déterminer, à chaque date τ_k , l'ensemble E_k de toutes les séquences de commandes discrètes réalisables. Les séquences discrètes ne sont composées que des transitions discrètes validées franchissables à partir du marquage discret courant noté $M_d(\tau_k)$. La longueur de ces séquences reste limitée par l'horizon de prédiction discret N_d .

La procédure de détermination des séquences de commandes est détaillée dans l'algorithme 7. Ce dernier est initialisé par le marquage discret courant et par l'horizon de prédiction. Soit \mathcal{N}_h^{d-comm} la structure de la partie discrète du RdPH en ne considérant que des transitions discrètes immédiates commandables. Notons par $R_d(\mathcal{N}_h^{d-comm}, M_d(\tau_k))$ l'ensemble de tous les marquages discrets atteignables à partir du marquage $M_d(\tau_k)$. Cet ensemble est construit de façon itérative au fur et à mesure que l'algorithme 7 détermine les séquences de commandes possibles de taille maximale N_d . Soit $E_{M'_d}$ l'ensemble des séquences discrètes qui permettent d'atteindre un marquage $M'_d \in R_d(\mathcal{N}_h^{d-comm}, M_d(\tau_k))$ à partir de $M_d(\tau_k)$. Chaque séquence est constituée d'au plus N_d transitions discrètes commandables validées. L'union de tous les ensembles $E_{M'_d}$, $M'_d \in R_d(\mathcal{N}_h^{d-comm}, M_d(\tau_k))$, constitue l'ensemble E_k . D'une manière générale, ce dernier peut être exprimé par $E_k = \{\sigma_d \in (T_d^{comm})^* \text{ tel que } |\sigma_d| \leq N_d \text{ et } M_d(\tau_k) [\sigma_d >]\}$.

Remarque 4. *Dans le cadre de cette étude, la notion de perturbations est liée aux franchissements imprévus de transitions discrètes non-commandables.*

Algorithme 7 : Construction de l'ensemble E_k (Entrées : $M_d(\tau_k)$ et N_d ; Sortie : E_k)

1: Initialisation des paramètres :

$h \leftarrow 0$

$M'_d \leftarrow M_d(\tau_k)$

$R_d(\mathcal{N}_h^{d-comm}, M_d(\tau_k)) \leftarrow M_d(\tau_k)$

$E_{M'_d} \leftarrow \emptyset$

$H_{M'_d} \leftarrow 0$

2: **Tant que** $h \leq N_d$

3: **Pour** chaque marquage $M'_d \in R_d(\mathcal{N}_h^{d-comm}, M_d(\tau_k))$ tel que $H_{M'_d} = h$

4: **Pour** chaque transition $t_j \in T_d^{comm}$ telle que $M'_d[t_j >$

5: Calculer $M'_d[t_j > M''_d$

6: **Si** $M''_d \notin R_d(\mathcal{N}_h^{d-comm}, M_d(\tau_k))$, **alors**

7: $R_d(\mathcal{N}_h^{d-comm}, M_d(\tau_k)) \leftarrow R_d(\mathcal{N}_h^{d-comm}, M_d(\tau_k)) \cup \{M''_d\}$

8: $E_{M''_d(\tau_k)} \leftarrow \emptyset$

9: $H_{M''_d} \leftarrow H_{M'_d} + 1$

10: **Fin si**

11: **Pour** chaque séquence $\sigma_d \in E_{M'_d}$

12: $E_{M''_d} \leftarrow E_{M''_d} \cup \{\sigma_d t_j\}$

13: **Fin pour**

14: **Fin pour**

15: **Fin pour**

16: $h \leftarrow h + 1$

17: **Fin tant que**

18: $E_k \leftarrow \{\sigma_d \in (T_d^{comm})^* \text{ tel que } |\sigma_d| \leq N_d \text{ et } M_d(\tau_k)[\sigma_d >\}$

19: Retourner E_k

IV.1.3 Commande prédictive continue

Avant d'entamer la détermination de la commande hybride, précisons tout d'abord que cette étude porte sur une classe particulière de SDH, à savoir les systèmes dynamiques continus munis d'actionneurs continus et discrets. Par conséquent, le marquage désiré ne concerne que les places continues. Le critère de performance considéré par la commande prédictive hybride, qui permet l'évolution du RdPHE du marquage initial au marquage désiré, est défini par l'erreur quadratique entre le marquage continu obtenu après N_c pas d'échantillonnage (c-à-d $M_c(\tau_{k+N_c})$) et le marquage continu désiré M_{refc} . Il est donné par l'équation (IV.1) :

$$J_M(\tau_{k+N_c}) = (M_c(\tau_{k+N_c}) - M_{refc})^t \cdot Q_{M_c} \cdot (M_c(\tau_{k+N_c}) - M_{refc}) \quad (IV.1)$$

Une fois l'ensemble E_k déterminé. La commande continue (c-à-d $U_c(\tau_{k+})$) est déterminée par le biais de la commande prédictive continue pour chacune des séquences de commandes discrètes. Cette commande minimise le critère $J_M(\tau_{k+N_c})$ sous les contraintes liées aux flux de franchissements des transitions continues.

Même si le critère de performance ne tient compte ni des marquages ni des flux intermédiaires, ceux-ci sont pris en compte dans les contraintes d'optimisation. En effet, $\forall e = 1, \dots, N_c$, le critère de performance (IV.1) est minimisé sous les contraintes (IV.2) :

$$\begin{aligned} \mathbf{a/} & \text{diag}(V_c^{ncomm}) \cdot U_c(\tau_{k+e}) = 0_{q_c \times 1} \\ \mathbf{b/} & X_c(\tau_{k+e-1}) \leq F_c(\tau_{k+e-1}) \\ \mathbf{c/} & 0_{q_c \times 1} \leq X_c(\tau_{k+e-1}) \end{aligned} \quad (IV.2)$$

Par conséquent, le couple $(X_d(\tau_k), U_c(\tau_{k+}))$ définit une commande hybride réalisable.

La commande proposée ici utilise les principes de la commande prédictive présentée dans le chapitre précédent. Afin de limiter la complexité algorithmique liée à l'augmentation de l'espace de recherche et donc des contraintes qui en découlent, la commande continue $U_c(\tau_{k+})$ est supposée être constante durant la totalité de la prédiction continue. De plus, les marquages prédits doivent rester dans la même région jusqu'au dernier pas de prédiction. Par ailleurs, puisque les transitions discrètes qui constituent les séquences de commandes sont immédiates, elles sont franchies en début de chaque période d'échantillonnage, et aucun événement discret ne se produit durant la prédiction continue. Cette hypothèse permet de simplifier le problème d'optimisation et d'éviter l'explosion combinatoire du nombre de cas possibles. Cette idée est illustrée par la figure (IV.2).

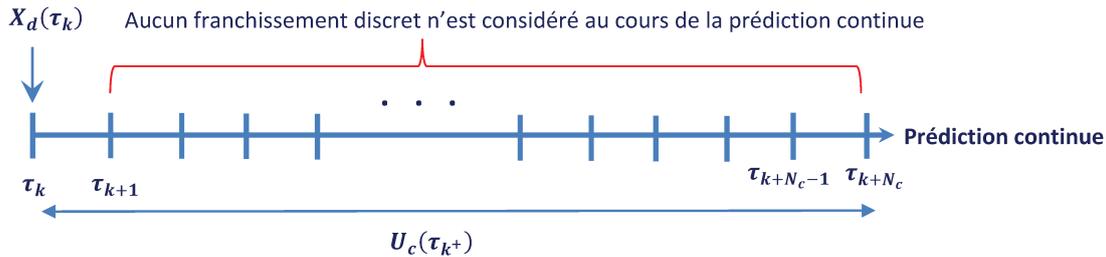


FIGURE IV.2 – Principe de la commande prédictive continue constante

La proposition IV.1.1 montre que le processus d'optimisation du critère de performance (IV.1) sous les contraintes (IV.2), peut être présenté sous la forme d'un problème d'optimisation quadratique à résoudre.

Proposition IV.1.1. *Soit $\langle \mathcal{N}_h, M_{0h}, \lambda, \theta, T^{comm} \rangle$ un θ -RdPHE avec des transitions discrètes immédiates. Soient $N_c > 0$ et $N_d > 0$ respectivement les horizons de prédiction des dynamiques continue et discrète. Supposons que tous les événements discrets soient synchronisés en début de chaque période d'échantillonnage, qu'aucun événement ne se déclenche durant la prédiction continue (c-à-d pendant l'intervalle de temps $]\tau_k, \tau_{k+N_c}]$) et que le marquage continu ne change pas de région avant le dernier pas de prédiction. Alors, la commande continue $U_c(\tau_{k+})$ associée au vecteur de franchissements discrets X_d (caractérisant la séquence de commandes $\sigma_d \in E_k$), minimisant le critère de performance (IV.1), sous les contraintes (IV.2), est la solution du problème d'optimisation quadratique (IV.3) sous les contraintes (IV.5) :*

$$U_c(\tau_{k+}) = \arg \min_{U_c} \{J_{M_{U_c}}(\tau_{k+N_c})\}, \sigma_d \in E_k \quad (IV.3)$$

tel que le critère $J_{M_{U_c}}(\tau_{k+N_c})$, $\sigma_d \in E_k$, s'écrit sous la forme (IV.4) :

$$J_{M_{U_c}}(\tau_{k+N_c}) = \frac{1}{2} \cdot U_c(\tau_{k+})^t \cdot \Omega \cdot U_c(\tau_{k+}) + \rho(X_d(\tau_k))^t \cdot U_c(\tau_{k+}) \quad (IV.4)$$

et :

$$\begin{aligned} \mathbf{a}/ & \text{diag}(V_c^{ncomm}) \cdot U_c(\tau_{k+}) = 0_{q_c \times 1} \\ \mathbf{b}/ & \forall e = 1, \dots, N_c, S_{e-1} \cdot U_c(\tau_{k+}) \geq 0_{q_c \times 1} \\ \mathbf{c}/ & S_{e-1} \cdot U_c(\tau_{k+}) \leq \Lambda \cdot A_k^{cc} \cdot ((G_k^c)^{e-1} \cdot M_c(\tau_k) + \Sigma_{e-1} \cdot G_k^d \cdot M_d(\tau_k)) \\ & + \Lambda \cdot (A_k^{cc} \cdot \Sigma_{e-1} \cdot G_k^d \cdot W^{dd} \cdot X_d(\tau_k) + A_k^{cd} \cdot (M_d(\tau_k) + W^{dd} \cdot X_d(\tau_k))) \end{aligned} \quad (IV.5)$$

Les termes Ω et $\rho(X_d(\tau_k))$ sont respectivement donnés par (IV.6) et (IV.7) :

$$\Omega = \theta^2 \cdot (W^{cc})^t \cdot (\Sigma_{N_c})^t \cdot Q_{M_c} \cdot (\Sigma_{N_c}) \cdot (W^{cc}) \quad (IV.6)$$

$$\begin{aligned} \rho(X_d(\tau_k)) &= \theta \cdot (W^{cc})^t \cdot (\Sigma_{N_c})^t \cdot Q_{M_c} \cdot (M_{refc} - (G_k^c)^{N_c} \cdot M_c(\tau_k)) \\ &\quad - \theta \cdot (W^{cc})^t \cdot (\Sigma_{N_c})^t \cdot Q_{M_c} \cdot \Sigma_{N_c} \cdot G_k^d \cdot (M_d(\tau_k) + W^{dd} \cdot X_d(\tau_k)) \end{aligned} \quad (IV.7)$$

Les constantes G_k^c et G_k^d , et le vecteur V_c^{ncomm} sont respectivement donnés par (I.39) et (II.23), l'ensemble E_k est déterminé à l'aide de l'algorithme 7 alors que les termes Σ_e et S_e , $e = 0, \dots, N_c$, sont définis par (IV.8) :

$$\begin{cases} \Sigma_e = \sum_{j=0}^{e-1} (G_k^c)^j, \Sigma_0 = 0_{n_c} \\ S_e = I_{q_c} + \theta \cdot \Lambda \cdot A_k^{cc} \cdot \Sigma_e \cdot W^{cc} \end{cases} \quad (IV.8)$$

Démonstration. En supposant que durant la prédiction la commande continue est constante, que le marquage continu ne change pas de région et qu'aucun franchissement discret ne se produira, alors le flux en régime forcé $X_c(\tau_{k+e-1})$ ainsi que le marquage continu $M_c(\tau_{k+e})$, $e = 1, \dots, N_c$, sont obtenus de façon itérative à partir de (II.21) et de (II.22). Ils sont respectivement exprimés par (IV.9) et (IV.10) :

$$\begin{aligned} X_c(\tau_{k+e-1}) &= \Lambda \cdot A_k^{cc} \cdot ((G_k^c)^{e-1} \cdot M_c(\tau_k) + \Sigma_{e-1} \cdot G_k^d \cdot (M_d(\tau_k) + W^{dd} \cdot X_d(\tau_k))) \\ &\quad + \Lambda \cdot A_k^{cd} \cdot (M_d(\tau_k) + W^{dd} \cdot X_d(\tau_k)) - S_{e-1} \cdot U_c(\tau_{k+}) \end{aligned} \quad (IV.9)$$

$$\begin{aligned} M_c(\tau_{k+e}) &= (G_k^c)^e \cdot M_c(\tau_k) + \Sigma_e \cdot G_k^d \cdot (M_d(\tau_k) + W^{dd} \cdot X_d(\tau_k)) \\ &\quad - \theta \cdot \Sigma_e \cdot W^{cc} \cdot U_c(\tau_{k+}) \end{aligned} \quad (IV.10)$$

L'expression (IV.10) est utilisée pour réécrire le critère $J_M(\tau_{k+N_c})$ donnée par (IV.1) et dans lequel tous les termes constants par rapport à la commande $U_c(\tau_{k+})$ sont supprimés. $J_M(\tau_{k+N_c})$ est alors réduite à $J_{M_{U_c}}(\tau_{k+N_c})$ défini par (IV.11) :

$$\begin{aligned} J_{M_{U_c}}(\tau_{k+N_c}) &= \frac{\theta^2}{2} \cdot U_c(\tau_{k+})^t \cdot (W^{cc})^t \cdot (\Sigma_{N_c})^t \cdot Q_{M_c} \cdot (\Sigma_{N_c}) \cdot (W^{cc}) \cdot U_c(\tau_{k+}) \\ &\quad + \theta \cdot (M_{refc} - (G_k^c)^{N_c} \cdot M_c(\tau_k))^t \cdot Q_{M_c} \cdot (\Sigma_{N_c}) \cdot (W^{cc}) \cdot U_c(\tau_{k+}) \\ &\quad - \theta \cdot (\Sigma_{N_c} \cdot G_k^d \cdot (M_d(\tau_k) + W^{dd} \cdot X_d(\tau_k)))^t \cdot Q_{M_c} \cdot (\Sigma_{N_c}) \cdot (W^{cc}) \cdot U_c(\tau_{k+}) \end{aligned} \quad (IV.11)$$

Sachant que $X_c(\tau_{k+e-1})$, $\forall e = 1, \dots, N_c$, est défini par (IV.9) et que le flux en régime libre (c-à-d $F_c(\tau_{k+e-1})$) est exprimé par (IV.12), alors les équations (IV.2) traduisant les

contraintes sur $U_c(\tau_{k+})$ peuvent être réécrites sous la forme données (IV.5).

$$F_c(\tau_{k+e-1}) = \Lambda \cdot A_k^{cc} \cdot \left((G_k^c)^{e-1} \cdot M_c(\tau_k) + \Sigma_{e-1} \cdot G_k^d \cdot (M_d(\tau_k) + W^{dd} \cdot X_d(\tau_k)) \right) \\ + \Lambda \cdot A_k^{cd} \cdot (M_d(\tau_k) + W^{dd} \cdot X_d(\tau_k)) \quad (\text{IV.12})$$

□

Remarque 5. *L'horizon de prédiction discret influence fortement les résultats obtenus. Le choix de cet horizon dépend d'un compromis : une valeur élevée permet, d'un coté, de faire une bonne prédiction de la partie discrète et de mieux explorer l'espace d'atteignabilité, mais d'autre coté, il génère une augmentation du nombre de séquences de commandes possibles. En conséquence, le nombre de problèmes d'optimisation quadratique à résoudre devient plus important.*

La proposition IV.1.1 énonce la forme générale de la solution du problème d'optimisation quadratique d'un système continu muni d'actionneurs continus et discrets. Les corollaires suivants donnent les deux cas particuliers qui peuvent exister.

Corollaire IV.1.2. *Si $\langle \mathcal{N}_h, M_{0h}, \lambda, \theta, T^{comm} \rangle$ modélise un système continu tel que les transitions discrètes ne sont pas commandables et seules les transitions continues peuvent être commandables (c -à- d $T_d^{comm} = \emptyset$, $T^{comm} = T_c^{comm}$), alors $X_d(\tau_k) = 0_{q_d \times 1}$ et la commande continue constante $U_c(\tau_k)$, qui minimise (IV.1) sous les contraintes (IV.5), est la solution du problème d'optimisation quadratique exprimé par (IV.13) sous les contraintes (IV.2) :*

$$U_c(\tau_k) = \arg \min_{U_c} \left\{ \frac{1}{2} \cdot U_c(\tau_k)^t \cdot \Omega \cdot U_c(\tau_k) + \rho(\tau_k)^t \cdot U_c(\tau_k) \right\} \quad (\text{IV.13})$$

tel que Ω et $\rho(\tau_k)$ sont respectivement définis par (IV.14) et (IV.15) :

$$\Omega = \theta^2 \cdot (W^{cc})^t \cdot (\Sigma_{N_c})^t \cdot Q_{M_c} \cdot (\Sigma_{N_c}) \cdot (W^{cc}) \quad (\text{IV.14})$$

$$\rho(\tau_k) = \theta \cdot (W^{cc})^t \cdot (\Sigma_{N_c})^t \cdot Q_{M_c} \cdot (M_{refc} - (G_k^c)^{N_c} \cdot M_c(\tau_k) - \Sigma_{N_c} \cdot G_k^d \cdot M_d(\tau_k)) \quad (\text{IV.15})$$

Notons qu'étant donné que les transitions discrètes sont non-commandables, $M_d(\tau_k) = M_{0d}$.

Corollaire IV.1.3. Si $\langle \mathcal{N}_h, M_{0h}, \lambda, \theta, T^{comm} \rangle$ modélise un système continu tel que les transitions continues ne sont pas commandables et seules les transitions discrètes sont commandables (c-à-d $T_c^{comm} = \emptyset$, $T^{comm} = T_d^{comm}$), le vecteur de franchissements discret $X_d(\tau_k)$, qui minimise le critère (IV.1), est la solution du problème d'optimisation quadratique exprimé par (IV.16) :

$$X_d(\tau_k) = \arg \min_{X_d} \left\{ \frac{1}{2} \cdot X_d(\tau_k)^t \cdot \Omega \cdot X_d(\tau_k) + \rho(\tau_k)^t \cdot X_d(\tau_k), \sigma_d \in E_k \right\} \quad (IV.16)$$

avec Ω et $\rho(\tau_k)$ respectivement définis par (IV.17) et (IV.18).

$$\Omega = (W^{dd})^t \cdot (G_k^d)^t \cdot (\Sigma_{N_c})^t \cdot Q_{M_c} \cdot (\Sigma_{N_c}) \cdot (G_k^d) \cdot (W^{dd}) \quad (IV.17)$$

$$\rho(\tau_k) = (W^{dd})^t \cdot (G_k^d)^t \cdot (\Sigma_{N_c})^t \cdot Q_{M_c} \cdot ((G_k^c)^{N_c} \cdot M_c(\tau_k) + (\Sigma_{N_c}) \cdot (G_k^d) \cdot M_d(\tau_k) - M_{refc}) \quad (IV.18)$$

La commande prédictive hybride proposée est détaillée dans l'algorithme 8.

Après avoir déterminé l'ensemble E_k , la proposition IV.1.1 est utilisée de manière itérative, à chaque date τ_k , afin de déterminer toutes les commandes hybrides possibles $(X_d(\tau_k), U_c(\tau_{k+}))$, $\sigma_d \in E_k$. Un critère de sélection est ensuite utilisé afin de choisir la commande hybride optimale. Cette dernière est celle qui minimise le critère (IV.1) parmi l'ensemble des commandes hybrides. Cette commande sera appliquée au système à partir du point courant. Le processus est répété jusqu'à l'obtention du marquage continu désiré.

À cette étape et avant d'entamer les autres parties de ce chapitre, nous allons illustrer l'application de la commande sur un exemple.

IV.1.4 Exemple illustratif 1

Afin de prouver l'efficacité de la commande prédictive proposée, le système de trois réservoirs connectés, illustré par la figure IV.3 [GA07], est considéré :

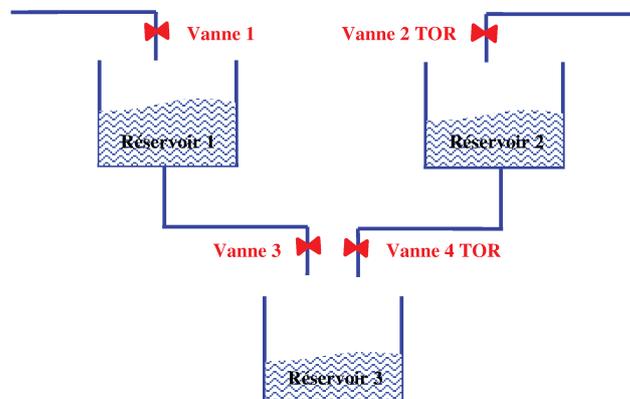


FIGURE IV.3 – Système de trois réservoirs connectés

Algorithme 8 : Commande prédictive hybride du θ -RdPHE avec des transitions discrètes immédiates (Entrées : M_{0h} , N_c et M_{refc} ; Sorties : σ_d^{tot} et U_c^{tot})

- 1: Initialisation des paramètres :
 - $\tau_k \leftarrow 0$
 - $M_d(\tau_k) \leftarrow M_{0d}$
 - $M_c(\tau_k) \leftarrow M_{0c}$
 - $\sigma_d^{tot} \leftarrow \emptyset$
 - $U_c^{tot} \leftarrow \emptyset$
 - 2: **Tant que** $M_c(\tau_k) \neq M_{refc}$
 - 3: Déterminer l'ensemble E_k à l'aide de l'algorithme 7
 - 4: **Pour** chaque séquence de commandes discrètes $\sigma_d \in E_k$
 - 5: Déterminer le vecteur de franchissements $X_d(\tau_k)$
 - 6: Calculer la commande continue constante $U_c(\tau_{k+})$, solution du problème d'optimisation (IV.3) sous les contraintes (IV.5)
 - 7: **Fin pour**
 - 8: Déterminer la commande hybride optimale qui minimise le critère $J_M(\tau_{k+N_c})$
 - 9: Calculer $M_d(\tau_{k+1})$, $X_c(\tau_{k+})$ et $M_c(\tau_{k+1})$
 - 10: $\sigma_d^{tot} \leftarrow \sigma_d^{tot} \cup \sigma_d(\tau_k)$
 - 11: $U_c^{tot} \leftarrow U_c^{tot} \cup U_c(\tau_{k+})$
 - 12: $\tau_k \leftarrow \tau_{k+1}$
 - 13: **Fin tant que**
 - 14: Retourner σ_d^{tot} et U_c^{tot}
-

Le processus d'alimentation des trois réservoirs est le suivant : les deux premières vannes alimentent les réservoirs 1 et 2. Ces derniers servent à remplir le troisième réservoir à travers les vannes 3 et 4. Ce système sera modélisé à l'aide d'un RdPH. Contrairement à [GA07], la sémantique serveurs infinis est adoptée plutôt que la sémantique serveurs finis, pour traduire le flux de franchissement des transitions continues. Les deux vannes 2 et 4 sont de type TOR (tout ou rien) alors que l'ouverture des vannes 1 et 3 est progressive. Dans cette section, nous supposons que le changement d'état (l'ouverture et la fermeture) de ces deux vannes est instantané. En conséquence, les transitions discrètes qui modéliseront ce comportement sont immédiates.

Le RdPHDE représenté dans la figure IV.4 est une modélisation possible du système étudié. La représentation adoptée dans les chapitres précédents, à savoir la représentation des places et des transitions continues par un double trait et celles discrètes par un trait simple, reste toujours valable pour le RdPH.

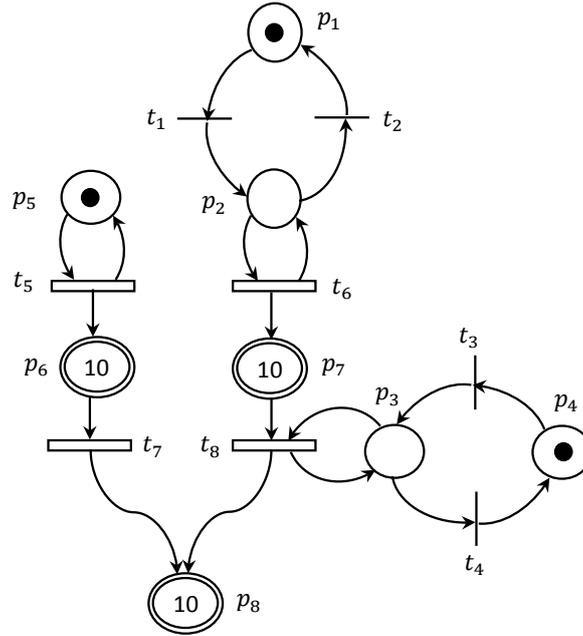


FIGURE IV.4 – RdPHD qui modélise le système de trois réservoirs connectés

Les places et les transitions continues, $P_c = \{p_6, p_7, p_8\}$ et $T_c = \{t_5, t_6, t_7, t_8\}$, représentent respectivement les trois réservoirs et les quatre vannes. En effet, le marquage continu correspond à la quantité de liquide dans chaque réservoir et le flux de franchissement des transitions correspond à la quantité du liquide qui coule à travers chaque vanne [GA07]. Les places et les transitions discrètes, $P_d = \{p_1, p_2, p_3, p_4, p_5\}$ et $T_d = \{t_1, t_2, t_3, t_4\}$, assurent le changement d'état des vannes 2 et 4. Toutes les transitions discrètes sont supposées être commandables (c-à-d $T_d^{comm} = T_d$) et les transitions continues t_5 et t_7 sont également commandables (c-à-d $T_c^{comm} = \{t_5, t_7\}$). Les autres transitions continues, qui ne sont pas commandables directement par la commande prédictive, le sont par le biais des sous-réseaux discrets. Elles sont franchies avec leur flux maximal de franchissement.

Nous admettons que les vannes 2 et 4 sont initialement fermées, ce qui implique que le marquage discret initial est donné par $M_{0d} = [1 \ 0 \ 0 \ 1 \ 1]^t$. De plus, nous supposons que les quantités de liquide dans les trois réservoirs sont initialement identiques. Elles sont de 10 U.V (Unité de Volume) chacune. Le marquage continu initial est donc donné par $M_{0c} = [10 \ 10 \ 10]^t$. La vitesse maximale de franchissement des transitions continues est $\lambda = [1 \ 1 \ 1 \ 1]^t$. Par conséquent, le RdPHDE remplit toutes les conditions requises dans cette partie. Ainsi, la commande prédictive hybride peut être appliquée. Nous choisissons comme objectif de commande le marquage continu suivant : $M_{refc} = [20 \ 20 \ 20]^t$. La simulation numérique a été effectuée en utilisant les paramètres suivants : $\theta = 0.2$ U.T., $r_{M_c} = 1$ et $N_d = 2$. Lorsque $N_c = 10$, la séquence de commandes discrètes globale σ_d^{tot}

(telle que $\sigma_d^{tot} = t_j(\tau_k)$ avec $t_j \in T_d^{comm}$ et τ_k sa date de franchissement) est donnée par $\sigma_d^{tot} = t_1(0) t_3(1.6) t_4(1.8) t_3(2) t_4(2.2) t_3(2.6) t_4(2.8) t_3(3.2) t_4(3.4) t_3(3.8) t_4(4) t_3(4.4) t_4(4.6) t_3(5) t_4(5.2) t_3(5.6) t_4(5.8) t_3(6.2) t_4(6.4) t_3(6.8) t_4(7) t_3(7.4) t_4(7.6) t_3(8) t_4(8.2) t_3(8.6) t_4(8.8) t_3(9.2) t_4(9.4) t_3(9.8) t_4(10) t_3(10.4) t_4(10.6) t_3(11) t_4(11.2) t_3(11.6) t_4(11.8) t_3(12.2) t_4(12.4) t_3(12.6) t_4(12.8) t_3(13) t_4(13.2) t_2(13.4) t_1(13.8) t_2(14) t_1(14.4) t_2(14.6) t_1(15) t_3(15.2) t_2(15.4) t_4(15.4) t_1(15.6) t_3(15.6) t_4(15.8) t_2(16)$. La commande continue qui correspond à cette séquence de commandes est donnée par la figure IV.5 :

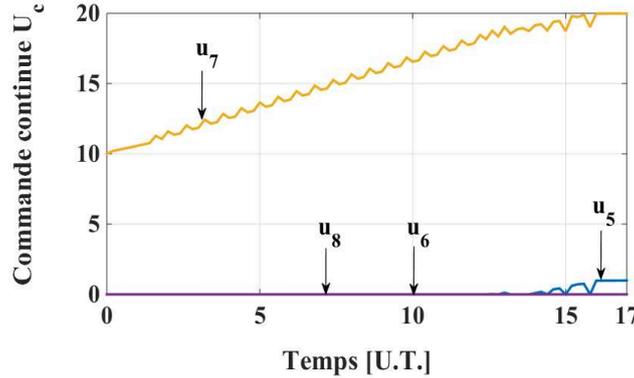


FIGURE IV.5 – Commande continue du RdPHDE

À l'instant initial, les séquences discrètes réalisables sont données par $E_0 = \{(\emptyset) (t_1 t_2) (t_3 t_4) (t_1) (t_3) (t_1 t_3)\}$. La commande hybride, composée de la transition t_1 et de la commande continue associée, est celle pour laquelle la valeur du critère $J_M(\tau_{k+N_c})$ est minimale. La transition t_1 est donc franchie, ce qui a entraîné l'ouverture de la deuxième vanne et le remplissage du deuxième réservoir. À partir de cet instant et jusqu'à la date $\tau_k = 1.4$ U.T., il faut garder la vanne 2 ouverte et la vanne 4 fermée afin de s'approcher du niveau de liquide désiré. Ainsi, aucune transition discrète n'est franchie et la commande optimale est donc constituée uniquement de la commande continue. À la date $\tau_k = 1.6$ U.T., $E_8 = \{(\emptyset) (t_2 t_1) (t_3 t_4) (t_2) (t_3) (t_3 t_2)\}$, la transition t_3 et son action de commande continuent de minimiser le critère de performance, la vanne 4 est alors ouverte afin de remplir le troisième réservoir à partir du deuxième. Cependant, à la date $\tau_k = 1.8$ U.T., si aucun franchissement n'a eu lieu, la valeur du critère de performance est de 169.35 alors que si la transition t_4 est franchie, la valeur de $J_M(\tau_{k+N_c})$ est de 168.96. En conséquence, la commande prédictive choisit de fermer la vanne 4 (en franchissant t_4). En revanche, au pas de calcul suivant, cette vanne doit s'ouvrir (en franchissant t_4) pour s'approcher davantage de la configuration désirée puisque $J_M(\tau_{k+N_c}) = 163.21$ et $J_M(\tau_{k+N_c}) = 163.26$ lorsque la vanne 4 reste fermée. Jusqu'à la date $\tau_k = 13.2$ U.T., il y a une succession d'ouvertures et de fermetures de la vanne 4 (franchissements successifs de t_3 et t_4 qui apparaissent dans la séquence totale de commandes). Étant donné que les places continues ont les mêmes

marquages initiaux et désirés, l'objectif de cette succession est de faire évoluer le système vers le marquage désiré tout en gardant un écart minimal entre les marquages des différentes places continues.

Une fois le niveau souhaité obtenu, toutes les vannes sont fermées ; les transitions continues commandables sont bloquées grâce à la commande continue et le flux à travers les transitions continues non-commandables est stoppé suite aux franchissements des transitions t_2 et t_4 aux instants 14.6 U.T. et 15.8 U.T. Les figures IV.6 et IV.7 donnent respectivement l'évolution temporelle du marquage continu et du flux de franchissement des transitions continues.

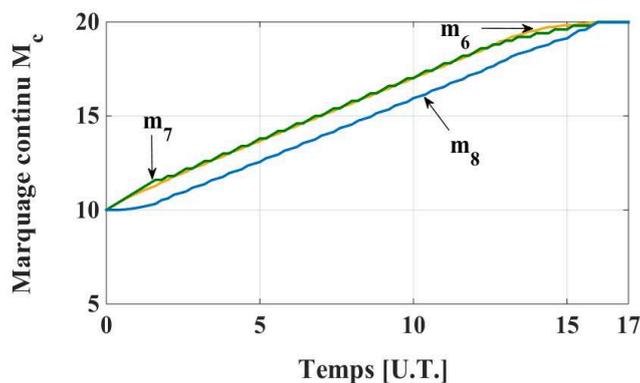


FIGURE IV.6 – Évolution temporelle du marquage continu M_c du RdPHDE

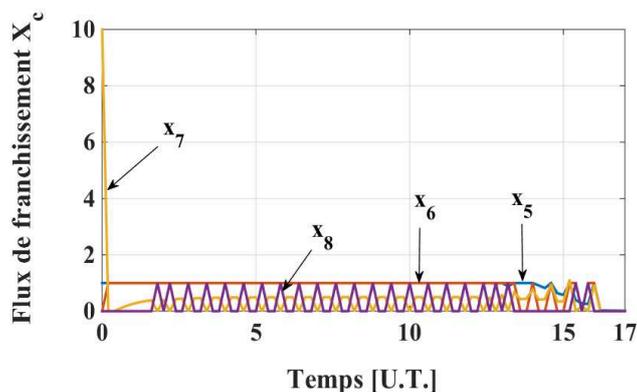


FIGURE IV.7 – Évolution temporelle du flux continu X_c du RdPHDE

En se référant à la figure IV.6, nous pouvons affirmer que la commande prédictive hybride a atteint avec succès le but envisagé. Néanmoins, d'après la figure IV.7, nous pouvons observer des fortes variations de flux notamment pour la transition t_7 ce qui se traduit par une sollicitation excessive des actionneurs continus.

La commande prédictive permet d'absorber les perturbations car l'optimisation est faite en ligne [GMR⁺06]. Néanmoins, elle peut donner naissance à de fortes variations du flux [TLL14b, TLL14a]. Ce problème a été considéré dans le chapitre précédent dans le cas d'un RdPC. La solution développée dans le chapitre précédent, sera étendue dans le paragraphe suivant pour le cas de la commande prédictive hybride.

IV.1.5 Commande prédictive hybride avec limitation de la variation du flux

Dans le cadre de cette étude, les systèmes à flux continu avec des actionneurs continus et discrets modélisés à l'aide du RdPHE sont considérés. Parmi les problèmes rencontrés dans ces systèmes, on peut citer le vieillissement des composants ou des actionneurs. Ces problèmes sont dus à plusieurs facteurs internes et externes parmi lesquels la sollicitation excessive des actionneurs. Sachant que les actionneurs continus sont modélisés par des transitions continues dans le RdPHE, la sollicitation excessive des actionneurs est équivalente à la forte variation du flux de franchissement des transitions continues. Afin de faire face à ce problème, un nouveau critère, désigné par $J_{dX_c}(\tau_{k+})$ et qui a pour vocation de réduire cette forte variation, est ajouté au critère $J_M(\tau_{k+N_c})$. En conséquence, le critère de performance global $J(\tau_{k+N_c})$ à minimiser, est exprimé par (IV.19) :

$$J(\tau_{k+N_c}) = J_M(\tau_{k+N_c}) + J_{dX_c}(\tau_{k+}) \quad (\text{IV.19})$$

Les deux critères $J_M(\tau_{k+N_c})$ et $J_{dX_c}(\tau_{k+})$ sont définis respectivement par (IV.1) et (IV.20) :

$$J_{dX_c}(\tau_{k+}) = \theta^2 \cdot A_{acc}(\tau_{k+})^t \cdot Q_{dX_c} \cdot A_{acc}(\tau_{k+}) \quad (\text{IV.20})$$

où Q_{dX_c} est une matrice positive qui définit la pondération de J_{dX_c} dans le critère $J(\tau_{k+N_c})$. $A_{acc}(\tau_{k+})$ correspond au vecteur d'accélération défini par l'équation (III.20).

Le calcul de la commande continue constante associée à chaque séquence de commandes $\sigma_d \in E_k$ repose sur la résolution d'un problème d'optimisation quadratique du nouveau critère $J(\tau_{k+N_c})$, ce dernier sera également réécrit sous la forme (IV.4) et sera minimisé sous des contraintes sur $U_c(\tau_{k+})$.

Proposition IV.1.4. Soit $\langle \mathcal{N}_h, M_{0h}, \lambda, \theta, T^{comm} \rangle$ un θ -RdPHE avec des transitions discrètes immédiates. Soient $N_c > 0$ et $N_d > 0$ respectivement les horizons de prédiction des dynamiques continue et discrète. Supposons que tous les événements discrets soient synchronisés en début de chaque période d'échantillonnage, qu'aucun événement ne se déclenche durant la prédiction continue (c-à-d pendant l'intervalle de temps $]\tau_k, \tau_{k+N_c}]$) et que le marquage continu ne change pas de région avant le dernier pas de prédiction. Alors, la commande continue $U_c(\tau_{k+})$ associée au vecteur de franchissements discret (caractérisant la séquence de commandes $\sigma_d \in E_k$), minimisant le critère de performance (IV.19), sous les contraintes (IV.2), est la solution du problème d'optimisation quadratique (IV.21) sous les contraintes (IV.5) :

$$U_c(\tau_{k+}) = \arg \min_{U_c} \{J_{U_c}(\tau_{k+N_c})\}, \forall \sigma_d \in E_k \quad (IV.21)$$

tel que le critère $J_{U_c}(\tau_{k+N_c})$ s'écrit sous la forme (IV.22) :

$$J_{U_c}(\tau_{k+N_c}) = \frac{1}{2} \cdot U_c(\tau_{k+})^t \cdot \Omega \cdot U_c(\tau_{k+}) + \rho(X_d(\tau_k))^t \cdot U_c(\tau_{k+}) \quad (IV.22)$$

Les deux paramètres Ω et $\rho(X_d(\tau_k))$ sont respectivement exprimés par (IV.23) et (IV.24) :

$$\Omega = \theta^2 \cdot (W^{cc})^t \cdot (\Sigma_{N_c})^t \cdot Q_{M_c} \cdot (\Sigma_{N_c}) \cdot (W^{cc}) + Q_{dX_c} \quad (IV.23)$$

$$\begin{aligned} \rho(X_d(\tau_k)) = & \theta \cdot (W^{cc})^t \cdot (\Sigma_{N_c})^t \cdot Q_{M_c} \cdot (M_{refc} - (G_k^c)^{N_c} \cdot M_c(\tau_k)) \\ & - \theta \cdot (W^{cc})^t \cdot (\Sigma_{N_c})^t \cdot Q_{M_c} \cdot \Sigma_{N_c} \cdot G_k^d \cdot (M_d(\tau_k) + W^{dd} \cdot X_d(\tau_k)) \\ & + Q_{dX_c} \cdot (X_c(\tau_{k-1}) - \Lambda \cdot A_k^{cc} \cdot M_c(\tau_k) - \Lambda \cdot A_k^{cd} \cdot (M_d(\tau_k) + W^{dd} \cdot X_d(\tau_k))) \end{aligned} \quad (IV.24)$$

où Σ_{N_c} est défini par (IV.8).

Démonstration. Le marquage et le flux sont remplacés dans le critère $J(\tau_{k+N_c})$ par leurs expressions analytiques données par (IV.10) et (IV.9). Seuls les termes qui dépendent de $U_c(\tau_{k+})$ sont retenus. De ce fait, $J(\tau_{k+N_c})$ est réduit à $J_{U_c}(\tau_{k+N_c})$ donné par (IV.25) :

$$\begin{aligned} J_{U_c}(\tau_{k+N_c}) = & \frac{\theta^2}{2} \cdot U_c(\tau_{k+})^t \cdot (W^{cc})^t \cdot (\Sigma_{N_c})^t \cdot Q_{M_c} \cdot (\Sigma_{N_c}) \cdot (W^{cc}) \cdot U_c(\tau_{k+}) \\ & + \theta \cdot (M_{refc} - (G_k^c)^{N_c} \cdot M_c(\tau_k))^t \cdot Q_{M_c} \cdot (\Sigma_{N_c}) \cdot (W^{cc}) \cdot U_c(\tau_{k+}) \\ & - \theta \cdot (\Sigma_{N_c} \cdot G_k^d \cdot (M_d(\tau_k) + W^{dd} \cdot X_d(\tau_k)))^t \cdot Q_{M_c} \cdot (\Sigma_{N_c}) \cdot (W^{cc}) \cdot U_c(\tau_{k+}) \end{aligned} \quad (IV.25)$$

$J_{U_c}(\tau_{k+N_c})$ est reformulé selon (IV.22). Les deux termes Ω et $\rho(X_d(\tau_k))$ sont respectivement définis par (IV.23) et (IV.24). \square

Désormais, à chaque période d'échantillonnage, la proposition IV.1.4 sera utilisée de façon itérative afin de déterminer toutes les commandes hybrides réalisables.

Passons maintenant à l'étape principale de la commande prédictive hybride qui réside dans la détermination de la commande hybride optimale. Rappelons que l'objectif de la commande prédictive hybride est d'atteindre un marquage désiré et supposons que le critère $J_{dX_c}(\tau_{k+})$ rentre dans le choix de la commande hybride optimale. Deux cas de figures peuvent se présenter : d'une part, une commande hybride qui donne un marquage continu proche du marquage désiré mais qui présente une variation significative du flux par rapport aux autres, et d'autre part, une commande dont le marquage résultant n'est pas assez proche du marquage désiré, par rapport à la première commande, mais dont la variation de flux est quasiment négligeable. Théoriquement, la commande optimale devra correspondre à cette deuxième commande. En conséquence, le critère de sélection de la commande optimale doit favoriser la commande qui présente la plus faible variation de flux même si elle ne converge pas rapidement vers l'objectif. Le critère $J_{dX_c}(\tau_{k+})$ a été déterminé de telle sorte à réduire les variations des flux de franchissement des transitions, mais ces variations sont généralement nécessaires pour atteindre le marquage désiré. En l'absence de ces variations, soit le marquage met plus de temps pour converger vers l'objectif souhaité, soit il reste bloqué dans un minimum local.

Pour toutes ces raisons, $J_{dX_c}(\tau_{k+})$ n'est pas pris en compte dans la sélection de la commande optimale. Il n'est considéré que dans le calcul de la commande continue constante pour chaque séquence de commandes. La commande hybride optimale est celle qui minimise, parmi les autres commandes, le critère $J_M(\tau_{k+N_c})$. Elle sera appliquée au θ -RdPHE et le processus d'optimisation sera répété jusqu'à atteindre le marquage désiré. La commande prédictive hybride avec limitation de la variation du flux du θ -RdPHE avec des transitions discrètes immédiates est détaillée dans l'algorithme 9.

Cet algorithme est identique à l'algorithme 8. Le seul changement réside dans le problème d'optimisation (IV.21) au lieu de (IV.3).

IV.1.6 Exemple illustratif 2

Dans cette sous-section, nous reprenons l'exemple du système de trois réservoirs et sa modélisation par un RdPHE (figures IV.3 et IV.4) afin de mettre en évidence l'efficacité de l'ajout du terme J_{dX_c} dans la limitation de la variation du flux. Pour une valeur fixée à $r_{dX_c} = 0.5$, la commande hybride obtenue est : $\sigma_d^{tot} = t_1(0) t_3(1.8) t_4(2) t_3(2.2) t_4(2.4) t_3(2.8) t_4(3) t_3(3.2) t_4(3.4) t_3(3.8) t_4(4) t_3(4.4) t_4(4.6) t_3(5) t_4(5.2) t_3(5.6) t_4(5.8) t_3(6.2) t_4(6.4) t_3(6.8) t_4(7) t_3(7.4) t_4(7.6) t_3(8) t_4(8.2) t_3(8.6) t_4(8.8) t_3(9.2) t_4(9.4) t_3(9.8) t_4(10) t_3(10.4) t_4(10.6) t_3(11) t_4(11.2) t_3(11.6) t_4(11.8) t_3(12.2) t_4(12.4) t_3(12.6) t_4(12.8) t_3(13) t_4(13.2) t_3(13.4) t_4(13.6) t_2(13.8) t_1(14.2) t_2(14.4) t_1(15.2) t_2(15.4) t_1(17) t_2(17.2)$.

Algorithme 9 : Commande prédictive hybride avec limitation de la variation du flux, du θ -RdPHE (Entrées : M_{0h} , N_c et M_{refc} ; Sorties : σ_d^{tot} et U_c^{tot})

1: Initialisation des paramètres :

$$\begin{aligned}\tau_k &\leftarrow 0 \\ M_d(\tau_k) &\leftarrow M_{0d} \\ M_c(\tau_k) &\leftarrow M_{0c} \\ \sigma_d^{tot} &\leftarrow \emptyset \\ U_c^{tot} &\leftarrow \emptyset\end{aligned}$$

2: **Tant que** $M_c(\tau_k) \neq M_{refc}$

3: Déterminer l'ensemble E_k à l'aide de l'algorithme 7

4: **Pour** chaque séquence de commandes discrètes $\sigma_d \in E_k$

5: Déterminer le vecteur de franchissements $X_d(\tau_k)$

6: Calculer la commande continue constante $U_c(\tau_{k+})$, solution du problème d'optimisation (IV.21) sous les contraintes (IV.5)

7: **Fin pour**

8: Déterminer la commande hybride optimale qui minimise le critère $J_M(\tau_{k+N_c})$

9: Calculer $M_d(\tau_{k+1})$, $X_c(\tau_{k+})$ et $M_c(\tau_{k+1})$

10: $\sigma_d^{tot} \leftarrow \sigma_d^{tot} \cup \sigma_d(\tau_k)$

11: $U_c^{tot} \leftarrow U_c^{tot} \cup U_c(\tau_{k+})$

12: $\tau_k \leftarrow \tau_{k+1}$

13: **Fin tant que**

14: Retourner σ_d^{tot} et U_c^{tot}

La commande continue, qui correspond à cette séquence de commandes, est présentée dans la figure IV.8 :

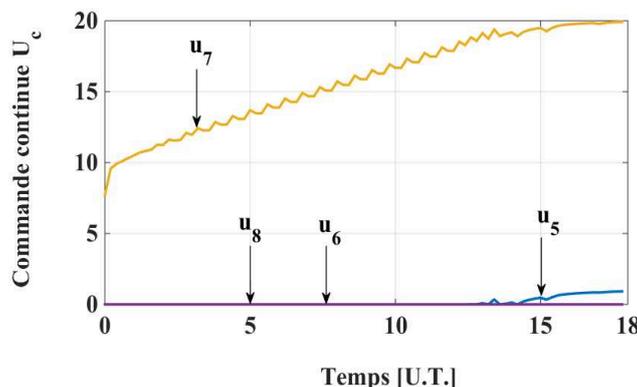


FIGURE IV.8 – Commande continue du RdPHDE lorsque $r_{dX_c} = 0.5$

Au premier pas de calcul, les séquences discrètes réalisables sont données par $E_0 = \{(\emptyset) (t_1 t_2) (t_3 t_4) (t_1) (t_3) (t_1 t_3)\}$. La commande hybride, composée de la transition t_1 et de la commande continue associée, minimise le critère $J_M(\tau_{k+N_c})$, le franchissement de cette transition a entraîné l'ouverture de la deuxième vanne et le remplissage du deuxième réservoir. À la date $\tau_k = 1.8$ U.T., $E_9 = \{(\emptyset) (t_2 t_1) (t_3 t_4) (t_2) (t_3) (t_3 t_2)\}$, la transition t_3 et son action de commande continue minimisent le critère de performance, la vanne 4

est alors ouverte afin de remplir le troisième réservoir à partir du deuxième. Or, au pas de calcul suivant, la vanne 4 a été fermée suite au franchissement de la transition t_4 , car $J_M(\tau_{k+N_c}) = 169.18$ et $J_M(\tau_{k+N_c}) = 169.66$ lorsque la vanne 4 reste ouverte. Jusqu'à la date $\tau_k = 13.6$ U.T., une succession de franchissements de t_3 et t_4 apparaît dans σ_d^{tot} (tel est le cas de l'exemple IV.1.4). Ceci est lié au fait que la commande hybride permet de faire évoluer le système vers la configuration désirée de telle sorte que tous les marquages aient toujours des valeurs proches.

Une fois le niveau souhaité obtenu, toutes les vannes sont fermées ; les transitions continues commandables sont bloquées grâce à la commande continue et le flux à travers les transitions continues non-commandables est stoppé suite aux franchissements des transitions t_2 et t_4 aux instants 13.6 U.T. et 17.2 U.T.

L'évolution du marquage continu M_c est présentée par la figure IV.9 :

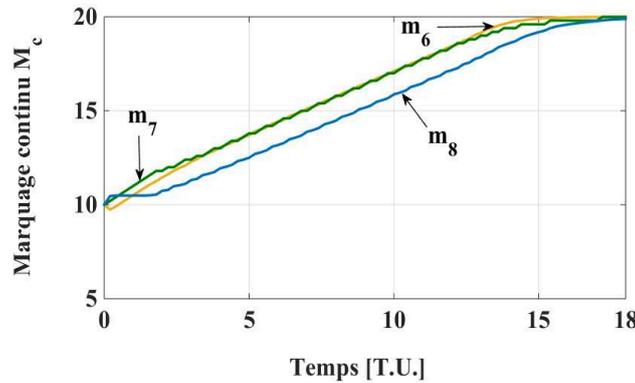


FIGURE IV.9 – Évolution temporelle du marquage continu M_c du RdPHDE lorsque $r_{dX_c} = 0.5$

En ce qui concerne la convergence, on peut constater une similitude entre la convergence du système avec et sans la présence de la limitation de la variation du flux. Pour ces deux cas, le système atteint le marquage désiré. En d'autres termes, l'ajout du facteur r_{dX_c} n'a ni influencé la convergence du système ni dégradé la qualité de la commande. Passons maintenant à l'objectif principal de cette partie. La figure IV.10 compare la norme de l'accélération obtenue quand $r_{dX_c} = 0$ à celle obtenue quand $r_{dX_c} = 0.5$. Afin de mettre l'accent sur l'importance de l'ajout de J_{dX_c} , il convient d'utiliser la somme cumulée du carré de la norme d'accélération $A_{acc}(\tau_{k+})$ multipliée par θ (c-à-d $A_C(\tau_{k+})$), définie par l'équation (III.29), comme un paramètre de comparaison.

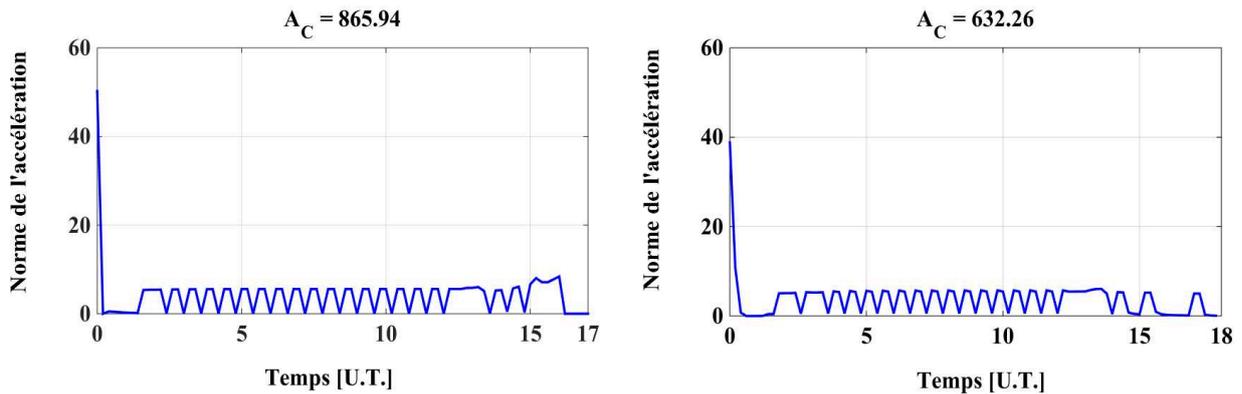


FIGURE IV.10 – Norme de l'accélération du RdPHDE lorsque $r_{dX_c} = 0$ (gauche) et $r_{dX_c} = 0.5$ (droite)

La figure IV.10 illustre une limitation modérée de la variation du flux et donc de la sollicitation des actionneurs continus qui résulte d'une valeur non-nulle du facteur r_{dX_c} . En fait, la valeur maximale de la norme d'accélération diminue de 50.25 à 38.83 lorsque r_{dX_c} passe de 0 à 0.5. En outre, sur les mêmes valeurs de r_{dX_c} , A_C chute de 865.94 jusqu'à 632.26. La norme de l'accélération et A_C ont perdu environ 25% de leur valeur initiale lors de la limitation de la variation de flux. Suite à ces comparaisons, nous pouvons conclure que les simulations coïncident avec les résultats attendus et que l'ajout de J_{dX_c} améliore tant la commande prédictive hybride que la commande prédictive.

Nous avons vu dans cette section l'intérêt de la limitation du flux par ajout du critère J_{dX_c} , appliqué à des θ -RdPH munis de transitions discrètes immédiates. L'extension de cette méthode avec le critère permettant de limiter le flux, en considérant des transitions discrètes temporisées, est présentée dans la section suivante.

IV.2 Commande prédictive hybride des RdPHE avec des transitions discrètes temporisées

Dans cette section, les transitions discrètes étudiées sont temporisées. Sachant que la séquence discrète influence la détermination de la commande continue, les franchissements discrets au cours de la prédiction continue doivent être pris en compte. D'autre part, les instants des franchissements discrets au cours de la prédiction continue sont a priori inconnus; ces franchissements peuvent être déclenchés à n'importe quel moment durant la prédiction continue. De multiples solutions doivent être explorées ce qui engendre une explosion de la complexité combinatoire et une augmentation du temps de calcul de la commande hybride. Afin d'éviter tous ces problèmes, une première commande prédictive hybride simplifiée est proposée. À chaque date τ_k , seules les séquences de commandes

discrètes immédiatement franchissables sont retenues. Les temporisations discrètes et la prédiction continue seront combinées dans une seule approche sans pour autant augmenter la complexité combinatoire. Finalement, une étude comparative des deux méthodes sera menée.

IV.2.1 Principe de la commande prédictive hybride simplifiée

L'idée clé de cette méthode de commande réside, en premier lieu, dans le calcul a priori des dates de franchissement de toutes les séquences qui forment l'ensemble E_k et, en second lieu, dans la mise à jour de l'ensemble E_k de telle sorte qu'il ne contienne que les séquences de commandes discrètes dont les dates de franchissement au plus tôt coïncident avec l'instant courant, ceci afin de vérifier les hypothèses des propositions IV.1.1 et IV.1.4. Les dates de franchissement des séquences de commandes sont sujettes à des contraintes temporelles données par les durées minimales de franchissement. Pour chaque séquence de commandes possible, la date de franchissement au plus tôt sera donc déterminée, à chaque période d'échantillonnage, à l'aide de deux échéanciers temporels mémorisant les franchissements des différentes transitions concernées. Cette méthode de commande repose principalement sur trois étapes :

- La première étape se rapporte à la détermination de l'ensemble E_k de toutes les séquences de commandes réalisables de telle sorte que toutes les transitions discrètes sont retenues même si elles ne sont pas immédiates. Cette phase est réalisée à l'aide de l'algorithme 7.
- La deuxième étape concerne le calcul des dates de franchissement au plus tôt des séquences, qui composent l'ensemble E_k , et l'actualisation de ce dernier. Cette phase s'appuie sur la construction de deux échéanciers temporels à savoir TCH et ECH . À chaque date τ_k , l'échéancier TCH mémorise les durées de franchissement restantes de toutes les transitions discrètes commandables qui sont validées par le marquage discret courant $M_d(\tau_k)$. Chacune de ces transitions peut apparaître une ou plusieurs fois dans l'échéancier tout dépend de son degré de validation $\eta_d(t_j, M_d(\tau_k))$. TCH est réactualisé de telle sorte que les transitions franchies à l'instant τ_{k-1} et celles qui ne sont plus validées à l'instant courant, sont supprimées ; d'un autre côté, s'il existe des transitions validées par le marquage discret courant qui ne sont pas enregistrées dans TCH , alors leurs durées minimales de franchissement seront mémorisées. Pour terminer, les durées restantes de franchissement des transitions enregistrées au préalable sont mises à jour en les décrémentant de la valeur θ (mises à zéro si elles deviennent négatives).

À partir de TCH , un deuxième échéancier, ECH , est construit afin de calculer

les dates de franchissement au plus tôt des différentes séquences de commandes réalisables et afin de vérifier si elles sont immédiatement franchissables ou non. ECH est initialisé à partir des durées restantes mémorisées dans TCH auxquelles est ajouté l'instant courant τ_k . Chaque séquence $\sigma_d \in E_k$ est parcourue en déterminant les marquages intermédiaires et l'on effectue la mise à jour de ECH . La transition récemment franchie ainsi que celles qui ne sont plus validées par le marquage intermédiaire qui en résulte, sont supprimées. La date qui correspond à ce dernier franchissement, à laquelle sont ajoutées les durées minimales données par D_{min} , est affectée aux transitions nouvellement validées. En définitive, la date de chaque transition validée et anciennement stockée dans ECH , est actualisée de telle sorte que la durée corresponde à la valeur maximale entre la date actuelle et celle de la dernière transition franchie. Si la date de la transition que l'on va franchir ne correspond pas à τ_k , cette séquence ne peut pas être immédiatement franchie, elle est donc supprimée de l'ensemble E_k . Dans le cas contraire on poursuit le parcours de la séquence. Il en résulte un ensemble E_k ne contenant que les séquences de commandes immédiatement franchissables. Cette étape est détaillée dans l'algorithme 10.

- Lors de la troisième étape, la détermination des commandes hybrides sera achevée suite au calcul de la commande continue constante associée à chacune des séquences de commandes discrète retenues.

La limitation de la variation de flux est considérée dans cette méthode, ce qui revient à dire que la commande continue résulte de la résolution du problème d'optimisation quadratique par le biais de la proposition IV.1.4.

Comme il est expliqué précédemment, la commande hybride optimale sera celle qui minimisera le critère $J_M(\tau_{k+N_c})$.

Ces trois étapes sont répétées, à chaque date τ_k , jusqu'à l'obtention du marquage désiré. La structure complète de la présente méthode de commande est détaillée dans l'algorithme 11.

Algorithme 10 : Calcul des dates de franchissement au plus tôt et mise à jour de l'ensemble E_k (Entrées : τ_k , E_k , $M_d(\tau_k)$ et TCH ; Sortie : E_k)

```

1: Pour chaque séquence de commandes  $\sigma_d \in E_k$ 
2:    $ECH \leftarrow TCH + \tau_k$ 
3:    $M'_d \leftarrow M_d(\tau_k)$ 
4:    $s \leftarrow 1$ 
5:    $immediate \leftarrow 1$ 
6:   Tant que  $s \leq |\sigma_d|$  et  $immediate = 1$ 
7:     Supprimer les transitions qui ne sont pas validées par  $M'_d$ 
8:     Insérer dans  $ECH$  les transitions nouvellement validées
9:     Si la date de franchissement de la transition  $\sigma_d(s)$  ne correspond pas à  $\tau_k$ , alors
10:      Supprimer la séquence  $\sigma_d$  de  $E_k$ 
11:       $immediate \leftarrow 0$ 
12:     Sinon
13:       Calculer le marquage intermédiaire  $M'_d$  résultant du franchissement de  $\sigma_d(s)$ 
14:       Supprimer  $\sigma_d(s)$  de  $ECH$ 
15:       Mettre à jour les dates des transitions enregistrées dans  $ECH$ 
16:        $s \leftarrow s + 1$ 
17:     Fin si
18:   Fin tant que
19: Fin pour
20: Retourner  $E_k$ 

```

Dans le paragraphe suivant, la commande prédictive hybride simplifiée est illustrée par un exemple.

IV.2.2 Exemple illustratif 3

Revenons à l'exemple précédent des trois réservoirs (figure IV.3) et sa modélisation sous forme d'un RdPHDE (figure IV.3). En supposant que les durées d'ouverture et de fermeture des vannes 2 et 4 ne peuvent pas être négligées, les transitions discrètes commandables t_1 , t_2 , t_3 et t_4 , qui assurent ces deux actions, sont désormais temporisées. Soit $D_{min} = \begin{bmatrix} 0.3 & 0.3 & 0.3 & 0.3 \end{bmatrix}^t$. Le marquage continu désiré ainsi que les autres paramètres de simulation sont les mêmes que pour l'exemple IV.1.4 et IV.1.6. La séquence de commandes discrètes temporisée obtenue à la fin de la simulation est $\sigma_d^{tot} = t_1(0.4) t_3(2.4) t_4(3) t_3(4) t_4(4.6) t_3(5.8) t_4(6.4) t_3(7.6) t_4(8.2) t_3(9.4) t_4(10) t_3(11.2) t_4(11.8) t_3(12.8) t_4(13.4) t_2(14) t_1(14.6) t_2(15.2)$.

La commande continue qui correspond à cette séquence est représentée par la figure IV.11.

Algorithme 11 : Commande prédictive hybride simplifiée du θ -RdPHE avec des transitions discrètes temporisées (Entrées : M_{0h} , N_c et M_{refc} ; Sorties : σ_d^{tot} et U_c^{tot})

1: Initialisation des paramètres :

$$\begin{aligned}\tau_k &\leftarrow 0 \\ M_d(\tau_k) &\leftarrow M_{0d} \\ M_c(\tau_k) &\leftarrow M_{0c} \\ \sigma_d^{tot} &\leftarrow \emptyset \\ U_c^{tot} &\leftarrow \emptyset\end{aligned}$$

2: **Tant que** $M_c(\tau_k) \neq M_{refc}$

3: Mettre à jour l'échéancier TCH

4: Déterminer l'ensemble E_k à l'aide de l'algorithme 7

5: Déterminer les séquences immédiatement franchissables et actualiser E_k à l'aide de l'algorithme 10

6: **Pour** chaque séquence de commandes $\sigma_d \in E_k$

7: Déterminer le vecteur de franchissements $X_d(\tau_k)$

8: Calculer la commande continue constante $U_c(\tau_{k+})$ solution du problème d'optimisation (IV.21) sous les contraintes (IV.5)

9: **Fin pour**

10: Déterminer la commande hybride optimale minimise le critère $J_M(\tau_{k+N_c})$

11: Calculer $M_d(\tau_{k+1})$, $X_c(\tau_{k+})$ et $M_c(\tau_{k+1})$

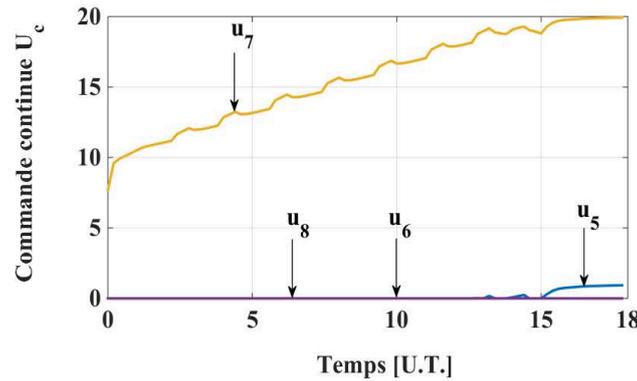
12: $\sigma_d^{tot} \leftarrow \sigma_d^{tot} \cup \sigma_d(\tau_k)$

13: $U_c^{tot} \leftarrow U_c^{tot} \cup U_c(\tau_{k+})$

14: $\tau_k \leftarrow \tau_{k+1}$

15: **Fin tant que**

16: Retourner σ_d^{tot} et U_c^{tot}

FIGURE IV.11 – Commande continue du RdPHDE lorsque $r_{dX_c} = 0.5$

Contrairement aux exemples précédents où les transitions sont supposées être immédiates, la première transition de la séquence (t_1) n'est franchie qu'à l'instant $\tau_k = 0.4$ U.T. En effet, puisqu'à l'instant initial les durées minimales de toutes les séquences de franchissements possibles sont non-nulles, aucune ne peut être prise en compte ni dans E_0 (c-à-d $E_0 = \{(\emptyset)\}$) ni dans dans l'optimisation de J . Les transitions t_1 et t_3 restent validées pendant deux périodes d'échantillonnage. Leurs durées minimales de franchissement sont écoulées, et à partir de l'instant $\tau_k = 0.4$ U.T., ces transitions sont présentes dans E_2 et peuvent être prises en compte par la commande prédictive. Le couple constitué de la transition t_1 et de la commande continue associée minimise le critère de performance. Nous appliquons donc ces commandes au système, puis t_1 est supprimé de l'échéancier. De nouvelles transitions validées apparaissent et la procédure est réitérée jusqu'à l'obtention du marquage désiré.

Nous remarquons que cet exemple présente moins d'oscillations entre les transitions t_3 et t_4 qu'avec les exemples IV.1.4 et IV.1.6 (la taille de la séquence σ_d^{tot} est presque divisée par trois). Ceci est dû à la temporisation, plus précisément, 0.3 U.T. doivent au moins séparer les franchissements de t_3 et t_4 (de t_1 et t_2 également). Pendant ce temps, aucune transition n'est franchie, la commande hybride détermine la commande continue qui permet au système de converger vers le marquage désiré.

L'évolution du marquage continu en fonction du temps, est représentée par la figure IV.12 :

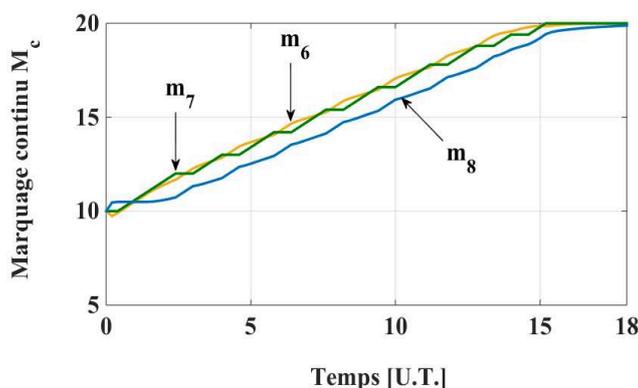


FIGURE IV.12 – Évolution temporelle du marquage continu du RdPHDE lorsque $r_{dX_c} = 0.5$

Une comparaison de la norme de l'accélération et de A_C entre $r_{dX_c} = 0$ et $r_{dX_c} = 0.5$ est donnée par la figure IV.13 :

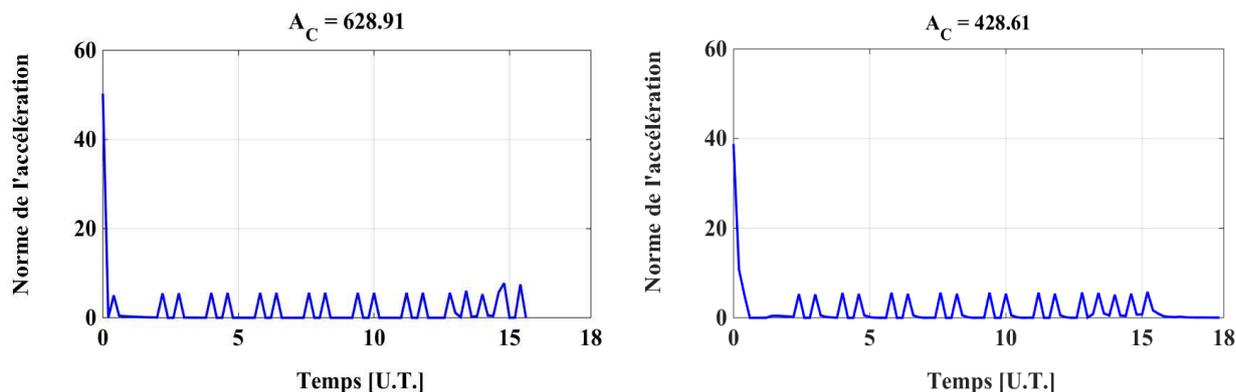


FIGURE IV.13 – Norme de l'accélération lorsque $r_{dX_c} = 0$ (gauche) et $r_{dX_c} = 0.5$ (droite)

La valeur maximale de la norme de l'accélération ainsi que celle de l'accélération cumulée A_C ont diminué d'une manière modérée, d'environ 23%. Elles baissent respectivement de 50 et 628.91 lorsque $r_{dX_c} = 0$ à 38.51 et 428.61 lorsque $r_{dX_c} = 0.5$.

Cette procédure est simple mais elle présente l'inconvénient de ne pas prendre en compte d'autres franchissements discrets hormis ceux effectués en début de période d'échantillonnage. Les transitions discrètes ne sont donc pas prises en compte. Une commande plus complète, sans être plus complexe, doit donc être proposée afin de considérer des événements discrets retardés et ainsi de prévoir l'influence d'un franchissement discret au cours de la prédiction sur le marquage $M_c(\tau_{k+N_c})$.

IV.2.3 Principe de la commande prédictive hybride complète

Le principe de cette méthode est de prendre en compte les franchissements discrets temporisés au cours de la prédiction continue. L'idée consiste à déterminer a priori les dates de franchissement au plus tôt de toutes les transitions qui composent chaque séquence $\sigma_d \in E_k$. Les franchissements seront ainsi pris en considération lors de la prédiction.

À l'instar de la première méthode de commande, cette approche est constituée de trois étapes.

La première concerne la détermination de l'ensemble E_k . Le but de la deuxième est de calculer les dates de franchissement au plus tôt à l'aide des échéanciers TCH et ECH . En conséquence, à partir d'une séquence de commandes non-temporisée, une séquence temporisée σ_d^{temp} est obtenue. Par la suite, les transitions d'une même séquence ayant la même date de franchissement seront regroupées ensemble. Nous désignons par Z le nombre des dates de franchissement distinctes différentes de τ_k . Le vecteur de franchissements de chaque séquence de commandes peut être décomposé comme suit : $X_d = [X_{d0}, \dots, X_{dz}, \dots, X_{dz}]$. Les vecteurs X_{d0} et X_{dz} spécifient respectivement le vecteur de franchissements des transitions dont les durées minimales de franchissement sont écoulées et le vecteur de franchissements des transitions qui vont se déclencher à la $z^{\text{ème}}$ date. De façon similaire, l'intervalle de prédiction $[\tau_k, \tau_{k+N_c}]$ sera subdivisé en plusieurs sous-intervalles comme expliqué par la figure IV.14 :

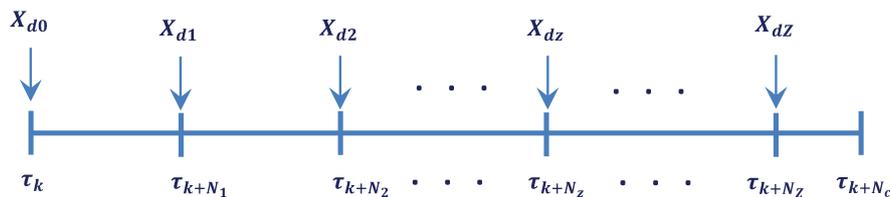


FIGURE IV.14 – Décomposition de l'intervalle de la prédiction continue

En conséquence, cet intervalle est représenté par l'union des sous-intervalles :

$$[\tau_k, \tau_{k+N_c}] = [\tau_k, \tau_{k+N_1} \cup [\tau_{k+N_1}, \tau_{k+N_2} \cup \dots \cup [\tau_{k+N_{z-1}}, \tau_{k+N_z} \cup \dots \cup [\tau_{k+N_Z}, \tau_{k+N_c}]]].$$

En fait, $\forall z = 0, \dots, Z$, l'horizon de prédiction intermédiaire N_z correspond au nombre de pas nécessaires pour que le vecteur X_{dz} soit pris en compte dans la prédiction continue. Il est calculé suivant la formule suivante : $N_z = \lceil \frac{\text{date de franchissement au plus tôt} - \tau_k}{\theta} \rceil$ sachant que $N_0 = 0$ et $N_{Z+1} = N_c$. Les transitions représentées par le vecteur X_{dz} , $z = 0, \dots, Z$, ne seront franchies qu'aux dates τ_{k+N_z} qui correspondent aux débuts de chaque sous-intervalle. Autrement dit, aucun franchissement discret commandable ne se produit en dehors de ces dates. Cette étape est détaillée dans l'algorithme 12.

Algorithme 12 : Détermination des horizons de prédiction intermédiaires (Entrées : M_{0c} , $M_d(\tau_k)$, σ_d , TCH et τ_k ; Sorties : N_1, \dots, N_Z et X_{d0}, \dots, X_{dZ})

- 1: Initialisation des paramètres :
 - $ECH \leftarrow TCH + \tau_k$
 - $M'_d \leftarrow M_d(\tau_k)$
 - $M_c(\tau_k) \leftarrow M_{0c}$
 - $s \leftarrow 1$
 - $\sigma_d^{temp} \leftarrow \emptyset$
 - 2: **Tant que** $s < |\sigma_d|$
 - 3: Supprimer de ECH les transitions qui ne sont pas validées par M'_d
 - 4: Insérer dans ECH les transitions nouvellement validées par M'_d
 - 5: Déterminer, à partir de ECH , la date de franchissement au plus tôt de $\sigma_d(s)$
 - 6: Calculer le marquage intermédiaire M'_d résultant du franchissement de $\sigma_d(s)$
 - 7: Supprimer $\sigma_d(s)$ de ECH
 - 8: Mettre à jour les dates des transitions enregistrées dans ECH
 - 9: Mettre à jour la séquence temporisée σ_d^{temp}
 - 10: $s \leftarrow s + 1$
 - 11: **Fin tant que**
 - 12: Déterminer Z à partir de σ_d^{temp}
 - 13: Décomposer le vecteur de franchissements en X_{d0}, \dots, X_{dZ}
 - 14: Calculer les horizons de prédiction intermédiaires N_1, \dots, N_Z
 - 15: Décomposer l'intervalle de prédiction
 - 16: Retourner N_1, \dots, N_Z et X_{d0}, \dots, X_{dZ}
-

Durant la troisième étape, la commande continue associée à chaque séquence de commandes est calculée. Le comportement de la partie continue peut être modifié suite aux franchissements discrets au cours de la prédiction continue ; l'action de commande continue ne peut donc pas être maintenue constante tout au long de la prédiction continue. Elle est supposée être constante au cours de la prédiction tant qu'il n'y a pas de franchissements discrets. Une nouvelle commande continue constante est ainsi calculée après chaque franchissement discret, ce qui signifie qu'une commande continue constante, désignée par U_{cz} , est associée à chaque sous-intervalle $[\tau_{k+N_z}, \tau_{k+N_{z+1}}]$, $z = 0, \dots, Z$. En définitive, une commande continue constante par morceaux est associée à chaque séquence comme l'indique la figure IV.15 :

La fonction coût à minimiser pour chaque intervalle $[\tau_{k+N_z}, \tau_{k+N_{z+1}}]$, $z = 0, \dots, Z$, est donnée par (IV.26) :

$$J(\tau_{k+N_{z+1}}) = J_M(\tau_{k+N_{z+1}}) + J_{dX_c}(\tau_{k+N_z^+}) \quad (\text{IV.26})$$

tel que $J_M(\tau_{k+N_{z+1}}) = (M_c(\tau_{k+N_{z+1}}) - M_{refc})^t \cdot Q_{M_c} \cdot (\tau_{k+N_{z+1}} - M_{refc})$
 et $J_{dX_c}(\tau_{k+N_z^+}) = (X_c(\tau_{k+N_z^+}) - X_c(\tau_{k-1}))^t \cdot Q_{dX_c} \cdot (X_c(\tau_{k+N_z^+}) - X_c(\tau_{k-1}))$.

De plus, à chaque pas de prédiction continue $e = N_z + 1, \dots, N_{z+1}$, la commande continue

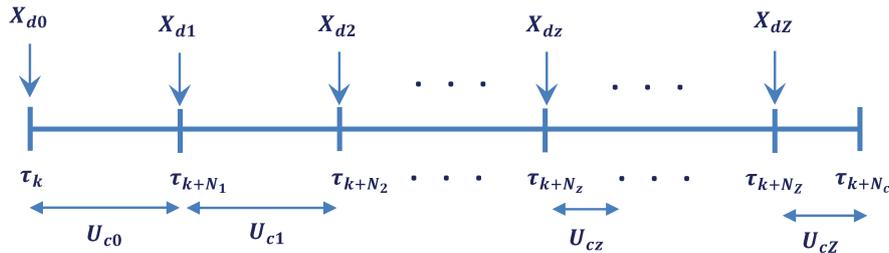


FIGURE IV.15 – Commande continue constante par sous-intervalle

et le flux forcé $X_c(\tau_{k+e-1})$ doivent vérifier les contraintes suivantes (IV.27) :

$$\begin{aligned}
 \mathbf{a/} \quad & \text{diag}(V_c^{ncomm}) \cdot U_{cz} = 0_{q_c \times 1} \\
 \mathbf{b/} \quad & X_c(\tau_{k+e-1}) \leq F_c(\tau_{k+e-1}) \\
 \mathbf{c/} \quad & 0_{q_c \times 1} \leq X_c(\tau_{k+e-1})
 \end{aligned} \tag{IV.27}$$

Il faut noter que les transitions sont franchies à la date τ_{k+N_z} et qu'aucun événement discret ne peut se déclencher durant l'intervalle $]\tau_{k+N_z}, \tau_{k+N_{z+1}}[$.

En résumé, le processus de détermination possible d'une commande hybride revient à appliquer à chaque intervalle $[\tau_{k+N_z}, \tau_{k+N_{z+1}}]$, la proposition IV.1.1 en prenant comme valeurs initiales celles obtenues par la commande précédente. La proposition IV.2.1 formalise l'optimisation quadratique de la fonction $J(\tau_{k+N_{z+1}})$ sous les contraintes (IV.27). Cette proposition est basée sur le principe de la proposition IV.1.1.

Proposition IV.2.1. *Soit $\langle \mathcal{N}_h, M_{0h}, D_{min}, \lambda, \theta, T^{comm} \rangle$ un θ -RdPHE avec des transitions discrètes temporisées. Soient $N_c > 0$ et $N_d > 0$ respectivement les horizons de prédiction des dynamiques continue et discrète. Supposons que pour la séquence de commandes $\sigma_d \in E_k$, les actions discrètes se produisent aux dates τ_{k+N_z} , $z = 0, \dots, Z$, qu'aucun événement discret ne se déclenche durant l'intervalle de temps $]\tau_{k+N_z}, \tau_{k+N_{z+1}}[$ et que le marquage continue ne change pas de région R_k (c-à-d la région du marquage $M_c(\tau_k)$). Alors, la commande continue U_{cz} associée au vecteur de franchissements discret X_{dz} , minimisant le critère $J(\tau_{k+N_{z+1}})$ sous les contraintes (IV.27), est la solution du problème d'optimisation quadratique (IV.28) sous les contraintes (IV.30) :*

$$U_{cz} = \arg \min_{U_{cz}} \{J_{U_c}(\tau_{k+N_{z+1}})\}, z = 0, \dots, Z \tag{IV.28}$$

tel que le critère $J_{U_c}(\tau_{k+N_z+1})$ est donné par (IV.29) :

$$J_{U_c}(\tau_{k+N_z+1}) = \frac{1}{2} \cdot U_{cz}^t \cdot \Omega \cdot U_{cz} + \rho(X_{dz})^t \cdot U_{cz} \quad (IV.29)$$

et :

$$\begin{aligned} \mathbf{a}/ \text{diag}(V_c^{ncomm}) \cdot U_{cz} &= 0_{q_c \times 1} \\ \mathbf{b}/ \forall e = N_z + 1, \dots, N_z + 1, S_{(e-N_z-1)} \cdot U_{cz} &\geq 0_{q_c \times 1} \\ \mathbf{c}/ S_{(e-N_z-1)} \cdot U_{cz} &\leq \Lambda \cdot A_k^{cc} \cdot (G_k^c)^{(e-N_z-1)} \cdot M_c(\tau_{k+N_z}) \\ &+ \Lambda \cdot A_k^{cc} \cdot \Sigma_{(e-N_z-1)} \cdot G_k^d \cdot M_d(\tau_{k+N_z}) \\ &+ \Lambda \cdot A_k^{cc} \cdot \Sigma_{(e-N_z-1)} \cdot G_k^d \cdot W^{dd} \cdot X_{dz} \\ &+ \Lambda \cdot A_k^{cd} \cdot (M_d(\tau_{k+N_z}) + W^{dd} \cdot X_{dz}) \end{aligned} \quad (IV.30)$$

Les termes Ω et $\rho(X_{dz})$ sont respectivement définis par (IV.31) et (IV.32) :

$$\Omega = \theta^2 \cdot (W^{cc})^t \cdot (\Sigma_{N_c})^t \cdot Q_{M_c} \cdot (\Sigma_{N_c}) \cdot (W^{cc}) + Q_{dX_c} \quad (IV.31)$$

$$\begin{aligned} \rho(X_{dz}) &= \theta \cdot (W^{cc})^t \cdot (\Sigma_{N_c})^t \cdot Q_{M_c} \cdot (M_{refc} - (G_k^c)^{(N_z+1-N_z)} \cdot M_c(\tau_{k+N_z})) \\ &- \theta \cdot (W^{cc})^t \cdot (\Sigma_{N_c})^t \cdot Q_{M_c} \cdot \Sigma_{(N_z+1-N_z)} \cdot G_k^d \cdot (M_d(\tau_{k+N_z}) + W^{dd} \cdot X_{dz}) \\ &+ Q_{dX_c} \cdot (X_c(\tau_{k-1}) - \Lambda \cdot A_k^{cc} \cdot M_c(\tau_{k+N_z}) - \Lambda \cdot A_k^{cd} \cdot (M_d(\tau_{k+N_z}) + W^{dd} \cdot X_{dz})) \end{aligned} \quad (IV.32)$$

Démonstration. Sous l'hypothèse que le marquage ne change pas de région, que la commande continue reste constante durant l'intervalle $[\tau_{k+N_z}, \tau_{k+N_z+1}]$ et que tous les franchissements discrets se produisent uniquement aux dates τ_{k+N_z} , $z = 0, \dots, Z$, alors $M_d(\tau_{k+N_z})$ est obtenu en franchissant toutes les transitions précédentes : $M_d(\tau_{k+N_z}) = M_d(\tau_{k+N_z-1}) + W^{dd} \cdot X_{dz-1}$. En outre, $M_c(\tau_{k+N_z+1})$, $X_c(\tau_{k+N_z+1-1})$ et $X_c(\tau_{k+N_z}^+)$ sont respectivement donnés par (IV.33), (IV.34) et (IV.35) avec Σ_e et S_e , $e = 0, \dots, N_c$, définis précédemment par (IV.8) :

$$\begin{aligned} M_c(\tau_{k+N_z+1}) &= (G_k^c)^{(N_z+1-N_z)} \cdot M_c(\tau_{k+N_z}) + \Sigma_{(N_z+1-N_z)} \cdot G_k^d \cdot (M_d(\tau_{k+N_z}) + W^{dd} \cdot X_{dz}) \\ &- \theta \cdot \Sigma_{(N_z+1-N_z)} \cdot W^{cc} \cdot U_{cz} \end{aligned} \quad (IV.33)$$

$$\begin{aligned} X_c(\tau_{k+N_z+1-1}) &= \Lambda \cdot A_k^{cc} \cdot (G_k^c)^{(N_z+1-N_z-1)} \cdot M_c(\tau_{k+N_z}) \\ &+ \Lambda \cdot A_k^{cc} \cdot \Sigma_{(N_z+1-N_z-1)} \cdot G_k^d \cdot (M_d(\tau_{k+N_z}) + W^{dd} \cdot X_{dz}) \\ &+ \Lambda \cdot A_k^{cd} \cdot (M_d(\tau_{k+N_z}) + W^{dd} \cdot X_{dz}) - S_{(N_z+1-N_z-1)} \cdot U_{cz} \end{aligned} \quad (IV.34)$$

$$X_c(\tau_{k+N_z}^+) = \Lambda \cdot (A_k^{cc} \cdot M_c(\tau_{k+N_z}) + A_k^{cd} \cdot (M_d(\tau_{k+N_z}) + W^{dd} \cdot X_{dz})) - U_{cz} \quad (IV.35)$$

Ainsi, $M_c(\tau_{k+N_z+1})$ et $X_c(\tau_{k+N_z+1})$ sont remplacés par leur expression mathématique dans $J(\tau_{k+N_z+1})$. En supprimant tous les termes constants par rapport à U_{cz} , le critère de performance $J(\tau_{k+N_z+1})$ est réduit à $J_U(\tau_{k+N_z+1})$ défini par (IV.36) :

$$\begin{aligned}
J_{U_c}(\tau_{k+N_z+1}) &= \theta^2 \cdot U_{cz}^t \cdot (W^{cc})^t \cdot (\Sigma_{N_c})^t \cdot Q_{M_c} \cdot (\Sigma_{N_c}) \cdot (W^{cc}) \cdot U_{cz} + U_{cz}^t \cdot Q_{dX_c} \cdot U_{cz} \\
&+ \theta \cdot (M_{refc} - (G_k^c)^{N_z+1-N_z} \cdot M_c(\tau_{k+N_z}))^t \cdot Q_{M_c} \cdot (\Sigma_{N_c}) \cdot (W^{cc}) \cdot U_{cz} \\
&- \theta \cdot (\Sigma_{N_z+1-N_z} \cdot G_k^d \cdot (M_d(\tau_{k+N_z}) + W^{dd} \cdot X_{dz}))^t \cdot Q_{M_c} \cdot (\Sigma_{N_c}) \cdot (W^{cc}) \cdot U_{cz} \\
&+ (X_c(\tau_{k-1}) - \Lambda \cdot A_k^{cc} \cdot M_c(\tau_{k+N_z}))^t \cdot Q_{dX_c} \cdot U_{cz} \\
&- (\Lambda \cdot A_k^{cd} \cdot (M_d(\tau_{k+N_z}) + W^{dd} \cdot X_{dz}))^t \cdot Q_{dX_c} \cdot U_{cz}
\end{aligned} \tag{IV.36}$$

Ce dernier peut être réécrit sous la forme (IV.29) telle que les deux termes Ω et $\rho(X_{dz})$ sont respectivement définis par (IV.31) et (IV.32). Sachant que $\forall e = N_z + 1, \dots, N_z+1$, le flux maximal de franchissement $F_c(\tau_{k+e-1})$ est exprimé selon (IV.37) :

$$\begin{aligned}
F_c(\tau_{k+e-1}) &= \Lambda \cdot A_k^{cc} \cdot ((G_k^c)^{(e-N_z-1)} \cdot M_c(\tau_{k+N_z}) + \Sigma_{(e-N_z-1)} \cdot G_k^d \cdot M_d(\tau_{k+N_z})) \\
&+ \Lambda \cdot (A_k^{cc} \cdot \Sigma_{(e-N_z-1)} \cdot G_k^d \cdot W^{dd} \cdot X_{dz} + A_k^{cd} \cdot (M_d(\tau_{k+N_z}) + W^{dd} \cdot X_{dz}))
\end{aligned} \tag{IV.37}$$

Par conséquent, le système d'équations (IV.27) implique les contraintes sur U_{cz} données par (IV.30). \square

Pour toute séquence $\sigma_d \in E_k$, la proposition IV.2.1 sera réitérée $Z + 1$ fois afin de déterminer la totalité de la commande U_c constante par sous-intervalle.

La commande hybride totale (X_d, U_c) , qui minimise le critère $J_M(\tau_{k+N_c})$, est optimale par rapport au critère (IV.29). Seuls les premiers vecteurs, à savoir X_{d0} et U_{c0} , seront appliqués à l'état courant du système. Le problème d'optimisation sera répété tant que le marquage désiré n'est pas atteint. La structure générale de la commande prédictive hybride, décrite tout le long de cette partie est détaillée dans l'algorithme 13.

Algorithme 13 : Commande prédictive hybride complète du θ -RdPHE avec des transitions discrètes temporisées (Entrées : M_{0h} , M_{refc} et N_c ; Sorties : σ_d^{tot} et U_c^{tot})

1: Initialisation des paramètres :

$$\begin{aligned}\tau_k &\leftarrow 0 \\ M_d(\tau_k) &\leftarrow M_{0d} \\ M_c(\tau_k) &\leftarrow M_{0c} \\ TCH &\leftarrow \emptyset \\ \sigma_d^{tot} &\leftarrow \emptyset \\ U_c^{tot} &\leftarrow \emptyset\end{aligned}$$

2: **Tant que** $M_c(\tau_k) \neq M_{refc}$

3: Mettre à jour l'échéancier TCH

4: Déterminer l'ensemble E_k par le biais de l'algorithme 7

5: **Pour** chaque séquence de commandes $\sigma_d \in E_k$

6: Déterminer les horizons N_z par le biais de l'algorithme 10

7: **Pour** chaque sous-intervalle $[\tau_{k+N_z}, \tau_{k+N_z+1}]$, $z = 0, \dots, Z$

8: Calculer la commande continue constante U_{cz} solution du problème d'optimisation (IV.28) sous les contraintes (IV.30)

9: **Fin pour**

10: **Fin pour**

11: Déterminer la commande hybride optimale qui minimise le critère $J_M(\tau_{k+N_c})$

12: Déterminer la sous-séquence σ_0 qui correspond à X_{d0}

13: Calculer $M_d(\tau_{k+1})$, $X_c(\tau_{k+1})$ et $M_c(\tau_{k+1})$

14: $\sigma_d^{tot} \leftarrow \sigma_d^{tot} \cup \sigma_0(\tau_k)$

15: $U_c^{tot} \leftarrow U_c^{tot} \cup U_{c0}$

16: $\tau_k \leftarrow \tau_{k+1}$

17: **Fin tant que**

18: Retourner σ_d^{tot} et U_c^{tot}

Afin d'illustrer l'efficacité de cette méthode et de la comparer à la commande simplifiée, nous avons repris l'exemple des trois réservoirs.

IV.2.4 Exemple illustratif 4

La séquence de commandes discrètes obtenue est $\sigma_d^{tot} = t_1(0.4) t_3(2.4) t_4(3) t_3(3.6) t_4(4.2) t_3(4.8) t_4(5.4) t_3(6) t_4(6.6) t_3(7.2) t_4(7.8) t_3(8.6) t_4(9.2) t_3(10.4) t_4(11) t_3(12.2) t_4(12.8) t_2(13.6) t_1(14.2) t_3(14.4) t_4(15) t_2(15.2) t_1(15.8) t_2(16.4) t_1(17)$. La commande continue associée est représentée sur la figure IV.16 :

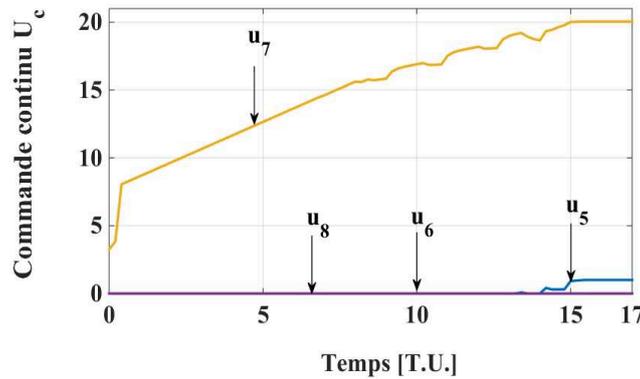


FIGURE IV.16 – Commande continue du RdPHDE lorsque $r_{dX_c} = 0.5$

À l'instant $\tau_k = 0$ U.T., les durées minimales de franchissement des transitions t_1 et t_3 sont mémorisées dans TCH et l'ensemble des séquences de commandes réalisables est donné par $E_0 = \{(\emptyset) (t_1) (t_3) (t_1 t_2) (t_1 t_3) (t_3 t_1) (t_3 t_4)\}$. Les dates de franchissement au plus tôt de chaque séquence sont calculées en utilisant l'échéancier ECH . Par exemple, dans le cas de la séquence $(t_1 t_2)$, la date de franchissement au plus tôt de t_1 et t_2 sont respectivement 0.3 U.T et 0.6 U.T. Ainsi, les horizons de prédiction intermédiaires sont $N_1 = 2$ et $N_2 = 3$. Les vecteurs de franchissements sont $X_{d1} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^t$ et $X_{d2} = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}^t$. Ils seront respectivement considérés, durant la prédiction continue, à partir des instants 2 U.T. et 3 U.T. De plus, l'intervalle de prédiction est décomposé en trois sous-intervalles définis comme suit : $[0, N_c] = [0, N_1] \cup [N_1, N_2] \cup [N_2, N_c] = [0, 2] \cup [2, 3] \cup [3, 10]$. La commande continue constante qui correspond à chaque intervalle (c-à-d U_{c0} , U_{c1} et U_{c2}) est déterminée par le biais de la proposition IV.2.1. La même procédure est effectuée pour les autres séquences. La commande hybride optimale correspond à la transition t_1 et à la commande continue associée. Cependant, t_1 ne peut pas être franchie à l'instant initial, donc seule la commande continue U_{c0} est appliquée au système.

À l'instant $\tau_k = 0.2$ U.T., les transitions t_1 et t_3 sont validées par le marquage discret,

les durées de franchissement restantes de ces deux transitions sont décrementées de θ . L'ensemble E_1 est identique à E_0 , mais les horizons de prédiction intermédiaires de chaque séquence ne sont plus les mêmes car les durées de franchissement restantes et donc les dates de franchissement au plus tôt sont modifiées. Considérons encore une fois la séquence précédente $(t_1 t_2)$, les horizons intermédiaires sont désormais $N_1 = 1$ et $N_2 = 2$. La commande hybride optimale à cet instant correspond à la transition t_1 et son action de commande continue. Comme à l'étape précédente, t_1 ne peut pas être franchie et U_{0c} est appliquée au système.

À $\tau_k = 0.4$ U.T., t_1 et l'action de commande continue associée correspondent toujours à la commande hybride qui minimise J_M . Puisque la durée minimale de franchissement de t_1 est écoulee, elle est franchie et elle sera supprimée de TCH . Cette procédure est répétée tant que le marquage désiré continu n'est pas atteint.

L'évolution du marquage continu en fonction du temps est présentée figure IV.17 :

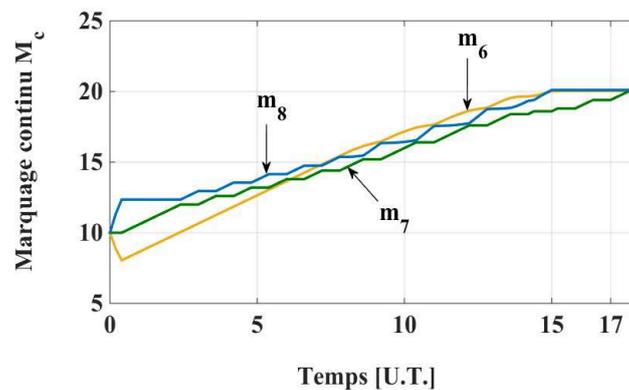


FIGURE IV.17 – Évolution temporelle du marquage continu du RdPHDE lorsque $r_{dX_c} = 0.5$

Comme pour les exemples précédents, une comparaison de la norme de l'accélération et de A_C entre $r_{dX_c} = 0$ et $r_{dX_c} = 0.5$ est proposée :

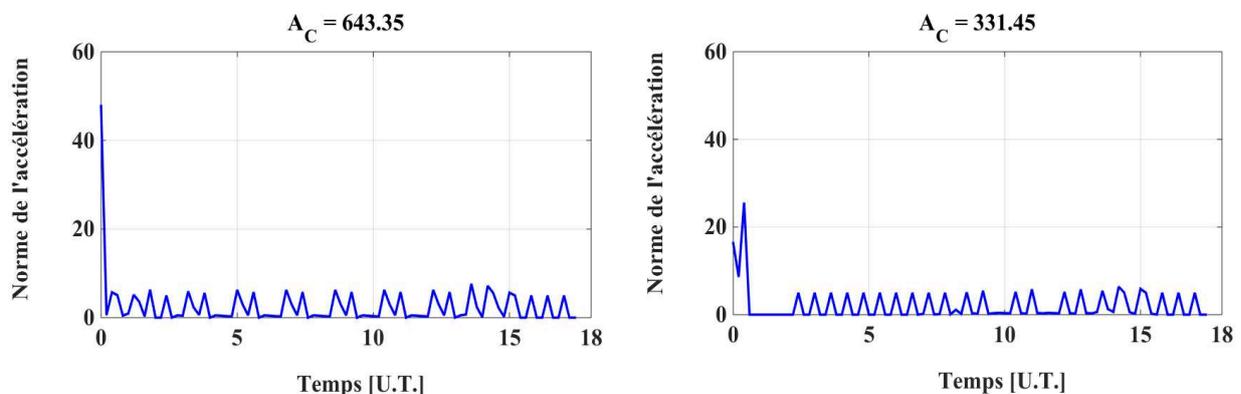


FIGURE IV.18 – Norme de l'accélération lorsque $r_{dX_c} = 0$ (gauche) et $r_{dX_c} = 0.5$ (droite)

La figure IV.18 montre que la valeur maximale de la norme de l'accélération a clairement diminué. Elle passe de 47.77 pour $r_{dX_c} = 0$ à 25.57 dans le cas de $r_{dX_c} = 0.5$. En outre, pour les mêmes valeurs de r_{dX_c} , A_C décroît de 643.35 à 331.45. Les valeurs ont diminué d'environ 47% lorsque la variation de débit est limitée ($r_{dX_c} = 0.5$). Finalement, nous pouvons affirmer que la commande prédictive proposée a permis au système étudié d'atteindre le niveau souhaité du liquide pour chaque réservoir avec une diminution de la sollicitation des actionneurs continus.

Conclusion

Ce chapitre concerne la conception d'une commande prédictive hybride qui permet au RdPHE d'atteindre un marquage désiré à partir d'un marquage initial. Principalement quatre contributions sont présentées.

La première propose une commande prédictive hybride en vue de commander le RdPHE pour lequel uniquement les actions de commande discrètes immédiates sont considérées. La deuxième contribution est l'élaboration d'une méthode qui permet de limiter la sollicitation excessive des actionneurs continus. La solution proposée consiste à ajouter, à la fonction de performance, un terme qui tient compte de la variation du flux. Les résultats de simulation montrent clairement l'efficacité de la méthode proposée.

La troisième contribution tient compte de la temporisation des transitions discrètes et suggère de garder le principe de la commande prédictive hybride avec limitation de la variation du flux mais en ne considérant dans l'ensemble des séquences faisables que celles dont les durées minimales de franchissement sont écoulées. Cette contribution présente un avantage important consistant à simplifier le problème d'optimisation en cas de temporisation. Cependant, elle n'est pas complète puisqu'elle se restreint aux séquences dont les durées minimales de franchissement sont écoulées. Pour cette raison, la quatrième contribution met l'accent sur le franchissement des transitions discrètes temporisées au cours de la prédiction continue. Plusieurs simulations numériques ont pu mettre en évidence les performances de ces méthodes pour commander le RdPHE temporisé.

CONCLUSIONS ET PERSPECTIVES

Notre étude, dans ce mémoire de thèse, porte sur l'utilisation de la commande prédictive pour commander les systèmes à événements discrets ainsi que les systèmes dynamiques continus et hybrides. Ces systèmes sont respectivement modélisés par des réseaux de Petri discrets (RdPD), des réseaux de Petri continus (RdPC) et des réseaux de Petri hybrides élémentaires (RdPHE). L'originalité de nos travaux réside dans quatre points :

- Le développement d'une commande prédictive continue de complexité réduite, par rapport à la commande prédictive classique. Cette commande permet, de plus, de contourner le problème de "hill-climbing", phase pendant laquelle la trajectoire du marquage doit s'éloigner du marquage désiré avant de l'atteindre.
- La limitation de la sollicitation excessive des actionneurs continus et l'étude de la stabilité du système. Ces contributions visent conjointement à préserver les équipements et à augmenter leur durée de vie.
- Dans le cas des systèmes à événements discrets, l'approche continue permet de déterminer la séquence de franchissements des RdPD pour atteindre un marquage désiré, sans exploration du graphe d'atteignabilité.
- Dans le cas des systèmes dynamiques hybride, cette même approche continue permet aussi de développer une commande prédictive hybride pour limiter la sollicitation des actionneurs continus des RdPHE.

Dans le but de limiter la complexité et donc le temps de calcul pour des horizons de prédiction larges ou des systèmes de grande taille, quelques modifications sont apportées à la commande prédictive usuelle. Il est supposé qu'au cours de la prédiction, la commande continue reste constante et que tous les marquages prédits restent dans la même région que celle du marquage courant. L'avantage de ces simplifications est d'éviter le problème de l'explosion combinatoire du nombre de configurations et de limiter l'espace

de recherche ainsi que le nombre des contraintes à considérer pendant la prédiction. Le critère de performance est également simplifié. Désormais, il ne tient compte que des écarts quadratiques du marquage et du flux obtenus au dernier pas de prédiction. Ce choix permet de considérer des horizons de prédiction importants, nécessaires lors des phases de “hill-climbing” qui se traduisent par une augmentation transitoire du critère de performance. Il est donc important de pouvoir prédire suffisamment loin pour détecter que le critère va diminuer et éviter que le système reste bloqué dans l'état courant qui serait un minimum local. L'horizon de prédiction doit être adaptatif et variable. Une expression analytique de la borne maximale est proposée. L'intérêt de cette borne est de garantir un temps de calcul inférieur à la période d'échantillonnage du calculateur. Les performances offertes par l'approche proposée, en termes de complexité, sont clairement améliorées par rapport à la commande prédictive classique.

En ce qui concerne la limitation de la sollicitation excessive des actionneurs, un nouveau terme qui prend en considération les variations de flux, est ajouté à la fonction de performance globale à minimiser. Plusieurs simulations numériques sont réalisées afin de mettre en relief l'importance de ce terme dans la limitation de la variation du flux. Finalement, une contrainte terminale est également ajoutée au problème d'optimisation. L'apport de cette contrainte est de garantir la stabilité asymptotique du système au voisinage du marquage désiré. Cela est démontré en s'appuyant sur le critère de Lyapunov.

Un algorithme de commande des RdPD temporisés, basé sur la commande prédictive des RdPC, est également proposé. Cette nouvelle stratégie de commande consiste à considérer simultanément le RdPD ainsi que le RdPC sous-jacent. Le RdPC est commandé à l'aide de la commande prédictive continue, pour atteindre le marquage discret désiré. La séquence de franchissements du RdPD est déterminée à partir du flux de franchissement des transitions continues. L'avantage principal de cet algorithme est de déterminer la séquence de commande discrète tout en évitant l'exploration du graphe d'atteignabilité du RdPD et donc de contourner le problème d'explosion combinatoire du nombre d'états. Une procédure qui calcule au fur et à mesure la date de franchissement au plus tôt des transitions discrètes, est aussi présentée.

Néanmoins, la commande discrète obtenue n'est pas nécessairement la commande optimale, en termes de durée globale ou de nombre de franchissements. De plus, l'algorithme proposé peut rencontrer un problème de convergence : la transition discrète sélectionnée pour être franchie peut ne pas être validée par le marquage discret courant.

Une commande prédictive hybride est finalement présentée. Les objectifs sont, d'une part, d'atteindre le marquage continu désiré à partir d'un marquage initial et, d'autre part de limiter la sollicitation excessive des actionneurs continus. La commande prédictive hybride proposée est composée de la commande prédictive discrète présentée dans [LL15] associée à la commande prédictive continue développée précédemment. La difficulté de la commande des RdPHE se situe dans la prise en compte des temporisations discrètes au cours de la prédiction continue. Une première approche consiste à supposer que toutes les transitions sont immédiates et qu'elles ne sont franchies qu'en début de chaque période échantillonnage. Cela permet de réduire la complexité de la commande hybride. Cette approche est étendue afin de prendre en compte les temporisations discrètes ; une première solution consiste à ne considérer que les séquences de franchissements discrets dont les durées minimales de franchissement sont écoulées. Cette proposition simplifie le problème d'optimisation et utilise une solution similaire à celle proposée dans le cas de transitions immédiates. La deuxième approche prend en compte le franchissement au plus tôt des transitions discrètes temporisées au cours de la prédiction continue. Elle suppose qu'entre deux franchissements discrets, la commande continue reste constante. Les simulations numériques effectuées ont mis en évidence les performances des différentes approches proposées.

En termes de perspectives pour ces travaux de recherche, nous envisageons de :

- Modifier la commande prédictive continue proposée afin de considérer des transitions continues non-commandables. Prendre aussi en considération les transitions discrètes non-commandables dans le cas hybride. Étudier la complexité ainsi que la convergence en nombre fini de pas de la commande obtenue. Étudier aussi le rejet de perturbations (c-à-d les franchissements inattendus de transitions non-commandables).
- Proposer des améliorations à l'algorithme de commande des RdPD pour que la séquence obtenue soit optimale et que la convergence soit assurée même si la transition discrète sélectionnée n'est pas validée. Puisque les RdPD considérés sont vivants, il existe au moins une transition validée par le marquage discret courant. Une des solutions, consisterait à franchir la transition discrète validée qui donne le marquage discret le plus proche du marquage désiré. Il faudrait également modifier l'algorithme afin d'éviter des marquages ou des régions interdites.
- Étendre la commande prédictive hybride développée afin de proposer une commande pour les réseaux de Petri hybrides non-élémentaires munis de transitions immédiates et temporisées. Limiter la sollicitation excessive des actionneurs conti-

- nus et discrets pour les systèmes modélisés par ces réseaux.
- Proposer une procédure qui adapte automatiquement les deux horizons de prédictions, continu et discret, utilisés dans la commande prédictive hybride.
 - Étudier la stabilité dans le cas des systèmes dynamiques hybrides et démontrer la convergence du système commandé via la commande prédictive hybride proposée.

- [AD88] H. Alla and R. David. Modeling of production systems by continuous Petri nets. In *3rd International Conference on CAD/CAM Robotics and Factories of the Future*, 344–348, 1988.
- [AOJMS11] H. Apaydin-Ozkan, J. Júlvez, C. Mahulea, and M. Silva. Approaching minimum time control of timed continuous Petri nets. *Nonlinear Analysis : Hybrid Systems*, 5(2) :136–148, 2011.
- [BCG06] F. Basile, P. Chiacchio, and A. Giua. Suboptimal supervisory control of Petri nets in presence of uncontrollable transitions via monitor places. *Automatica*, 42(6) :995–1004, 2006.
- [BCP13] F. Basile, R. Cordone, and L. Piroddi. Integrated design of optimal supervisors for the enforcement of static and behavioral specifications in Petri net models. *Automatica*, 49(11) :3432–3439, 2013.
- [Ben09] H. Benlaoukli. *Méthodes géométriques pour la construction des ensembles invariants. Application à la faisabilité des lois de commande prédictive*. Automatique/Robotique, Université Paris Sud - Paris XI, 2009.
- [BGM00] F. Balduzzi, A. Giua, and G. Menga. First-order hybrid Petri nets : a model for optimization and control. *IEEE Transactions on Robotics and Automation*, 16(4) :382–399, 2000.
- [BMDP02] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1) :3–20, 2002.
- [BPG04] O. T. Baruwa, M. A. Piera, and A. Guasch. Deadlock-free scheduling method for flexible manufacturing systems based on timed colored Petri nets and anytime heuristic search. *IEEE Systems, Man, and Cybernetics Society*, 34(1) :38–51, 2004.

- [DA92] R. David and H. Alla. *Petri Nets and Grafcet : tools for modeling discrete event systems*. London, 1992.
- [DA01] R. David and H. Alla. On hybrid Petri nets. *Discrete Event Dynamic Systems : Theory and Applications*, 11 :9–40, 2001.
- [DA08] A. Dideban and H. Alla. Reduction of constraints for controller synthesis based on safe Petri nets. *Automatica*, 44(7) :1697–1706, 2008.
- [DA10] R. David and H. Alla. *Discrete, Continuous, and Hybrid Petri Nets*. Springer-Verlag Berlin Heidelberg, second edition edition, 2010.
- [DBM⁺14] H. Dezani, R. D.S. Bassi, N. Marranghello, L. Gomes, F. Damiani, and I. N. da Silva. Optimizing urban traffic flow using genetic algorithm with Petri net analysis as fitness function. *Neurocomputing*, 124 :162–167, 2014.
- [DFGS08] M. Dotoli, M. Pia Fanti, A. Giua, and C. Seatzu. First-order hybrid Petri nets. An application to distributed manufacturing systems. *Nonlinear Analysis : Hybrid Systems*, 2(2) :408–430, 2008.
- [DFIM09] M. Dotoli, M. Pia Fanti, G. Iacobellis, and A. M. Mangini. A first-order hybrid Petri net model for supply chain management. *IEEE Transactions on Automation Science and Engineering*, 6(4) :744–758, 2009.
- [FIMU14] M. Pia Fanti, G. Iacobellis, A. M. Maangini, and W. Ukovich. Freeway traffic modeling and control in a first-order hybrid Petri net framework. *IEEE Transactions on Automation Science and Engineering*, 11(1) :90–102, 2014.
- [GA07] L. Ghomri and H. Alla. Modeling and analysis using hybrid Petri nets. *Nonlinear Analysis : Hybrid Systems*, 1(2) :141–153, 2007.
- [GDS92] A. Giua, F. DiCesare, and M. Silva. Generalized mutual exclusion constraints on nets with uncontrollable transitions. In *Proceeding IEEE International Conference on Systems, Man and Cybernetics*, 974–979, Chicago, IL, USA, 1992.
- [Gho12] L. Ghomri. *Synthèse de contrôleur de systèmes hybrides à flux continu par réseaux de Petri hybrides*. Ph.D. thesis, Université Abou-beker belaid-Tlemcen, Faculté de génie électrique et électronique, 2012.
- [GMLM13] B. Gudino-Mendoza and E. Lopez-Mellado. Modelling networked agents’ behaviour using timed hybrid Petri nets. In *3rd Iberoamerican Conference on Electronics Engineering and Computer Science*, 7, 289–296, San Luis Potosi, Mexico, 2013.
- [GMR⁺06] A. Giua, C. Mahulea, L. Recalde, C. Seatzu, and M. Silva. Optimal control of continuous Petri nets via model predictive control. In *8th International*

- Workshop on Discrete Event Systems*, 235–241, Ann Arbor, Michigan, USA, 2006.
- [JB10] J. Júlvez and R.K. Boel. A continuous Petri net approach for model predictive control of traffic systems. *IEEE Transactions on Systems, Man, and Cybernetics - Part A : Systems and Humans*, 40(4) :686–697, 2010.
- [JBR04] J. Júlvez, A. Bemporad, L. Recalde, and M. Silva. Event-driven optimal control of continuous Petri nets. In *43rd IEEE Conference on Decision and Control*, 69–74, Atlantis, Paradise Island, Bahamas, 2004.
- [JCBM13] J. Júlvez, S. Di Cairano, A. Bemporad, and C. Mahulea. Event-driven model predictive control of timed Petri nets. *International Journal of Robust and Nonlinear Control*, 24(12) :1724–1742, 2013.
- [JJRS05] E. Jimenez, J. Júlvez, L. Recalde, and M. Silva. On contrability of timed continuous Petri net systems : the join free case. In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, 7645–7650, Seville, Spain, 2005.
- [JMV11] J. Júlvez, C. Mahulea, and C. Vasquez. Analysis and simulation of manufacturing systems using SimHPN toolbox. In *IEEE Conference on Automation Science and Engineering*, 432–437, Trieste, Italy, 2011.
- [JVMS13] J. Júlvez, C. R. Vasquez, C. Mahulea, and M. Silva. *Control of Discrete-Event Systems Automata and Petri Net Perspectives*, 433. Springer-Verlag, London, 2013.
- [KALD09] R. Kara, M. Ahmane, J.J. Loiseau, and S. Djennoune. Constrained regulation of continuous Petri nets. *Nonlinear Analysis : Hybrid Systems*, 3(4) :738–748, 2009.
- [KHM06] F. Kaakai, S. Hayat, and A. El Moudni. Simulation of railway stations based on hybrid Petri nets. In *2nd IFAC Conference on Analysis and Design of Hybrid Systems*, 39, 50–55, Alghero, Italy, 2006.
- [KSN07] Y. W. Kim, T. Suzuki, and T. Narikiyo. FMS scheduling based on timed Petri net model and reactive graph search. *Applied Mathematical Modelling*, 31(6) :955–970, 2007.
- [LC04] G. F. List and M. Cetin. Modeling traffic signal control using Petri nets. *IEEE Transactions on Intelligent Transportation Systems*, 5(3) :177–187, 2004.
- [LD94] D. Y. Lee and F. DiCesare. Scheduling flexible manufacturing systems using Petri nets and heuristic search. *IEEE Transactions on Robotics and Automation*, 10(2) :123–132, 1994.

- [LHS05] L.Ghomri, H.Alla, and A. Sari. Structural and hierarchical translation of hybrid Petri nets in hybrid automata. In *IMACS'05*, Paris, France, 2005.
- [LL11] D. Lefebvre and E. Leclercq. Control design for continuous Petri net with piecewise-constant maximal firing speeds. In *18th IFAC World Congress*, 6036–6041, Milano, Italia, 2011.
- [LL13] E. Leclercq and D. Lefebvre. Feasibility of piecewise-constant control sequences for timed continuous Petri nets. *Automatica*, 49(12) :3654–3660, 2013.
- [LL15] D. Lefebvre and E. Leclercq. Control design for trajectory tracking with untimed Petri nets. *IEEE Transactions on Automatic Control*, 60(7) :1921–1926, 2015.
- [LLD13] D. Lefebvre, E. Leclercq, F. Druaux, and P. Thomas. Gradient-based controllers for timed continuous Petri nets. *International Journal of Systems Science, Taylor and Francis*, 46(9) :1661–1678, 2015.
- [MGR⁺08] C. Mahulea, A. Giua, L. Recalde, C. Seatzu, and M. Silva. Optimal model predictive control of timed continuous Petri nets. *IEEE Transactions on Automatic and Control*, 53(7) :1731–1735, 2008.
- [MRRS00] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control : Stability and optimality. *Automatica*, 36(6) :789–814, 2000.
- [MRS06] C. Mahulea, L. Recalde, and M. Silva. On performance monotonicity and basic servers semantics of continuous Petri nets. In *Proceedings of the 8th International Workshop on Discrete Event Systems*, 345–351, Ann Arbor, Michigan, USA, 2006.
- [MRSS06] C. Mahulea, L. Recalde, C. Seatzu, and M. Silva. On sampling continuous timed Petri nets : Reachability "equivalence" under infinite servers semantics. In *2nd IFAC Conference on Analysis and Design of Hybrid Systems*, 39, 37–43, Alghero, Italy, 2006.
- [MRTRS08] C. Mahulea, A. Ramirez-Trevino, L. Recalde, and M. Silva. Steady state control reference and token conservation laws in continuous Petri net systems. *IEEE Transactions on Automation Science and Engineering*, 5(2) :307–320, 2008.
- [MYK02] A. R. Moro, H. Yu, and G. Kelleher. Hybrid heuristic search for the scheduling of flexible manufacturing systems using Petri nets. *IEEE Transactions on Robotics and Automation*, 18(2) :240–245, 2002.

- [Pet66] Carl Adam Petri. *Communication with automata*. PhD thesis, 1966.
- [QB03] S. J. Qina and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7) :733–764, 2003.
- [RRTP78] J. Richalet, A. Rault, J.L. Testud, and J. Papon. Model predictive heuristic control : Applications to Industrial Processes. *Automatica*, 14(5) :413–428, 1978.
- [RW87] J.G. Ramadge and W.M. Wonham. Supervisory control of a class of discrete event processes. *SIAM Journal on Control and Optimization*, 25(1) :206–230, 1987.
- [RW89] J.G. Ramadge and W.M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1) :81–97, 1989.
- [San15] M. E. Fraca Santamaria. *Fluidization of Petri nets to improve the analysis of Discrete Event Systems*. Ph.D. thesis, Departamento de Informática e Ingeniería de Sistemas (DIIS) Universidad de Zaragoza, 2015.
- [SBB⁺11] N. Smata, D. Boudebous, J. Boukachour, S. Benmansour, and C. Tolba. Production, supply, and traffic systems : A unified modelling using Petri nets. In *4th International Conference on Logistics (Logistiqua)*, 405–411, Hammamet, Tunisia, 2011.
- [Sil93] M. Silva. Introducing Petri nets. *Practice of Petri Nets in Manufacturing*, 1993.
- [Sil13] M. Silva. Half a century after carl adam Petri’s Ph.D. thesis : A perspective on the field. *Annual Review in Controls*, 37(2) :191–219, 2013.
- [SJM^V11] M.l Silva, J. Júlvez, C. Mahulea, and C. R. Vasquez. On fluidization of discrete event models : observation and control of continuous Petri nets. *Discrete Event Dynamic Systems : Theory and Applications*, 21 :427–497, 2011.
- [Sma13] N. Smata. *Contribution à la modélisation et à l’analyse de la chaîne logistique en utilisant les réseaux de Petri*. Informatique/Automatique, Université du Havre, 2013.
- [SR02] M. Silva and L. Recalde. Petri nets and integrity relaxations : A view of continuous Petri net models. *IEEE Transactions Systems Man and Cybernetics-Part C*, 32(4) :314–327, 2002.
- [SR04] M. Silva and L. Recalde. On fluidification of Petri net models : from discrete to hybrid and continuous models. *Annual Reviews in Control*, 28(2) :253–206, 2004.

- [SR05] Manuel Silva and Laura Recalde. Continuization of timed Petri nets : from performance evaluation to observation and control. In *On the Equivalence Between Liveness and Deadlock-Freeness in Petri Nets*, 26–47, Miami, USA, 2005.
- [TLL14a] M. Taleb, E. Leclercq, and D. Lefebvre. Limitation of flow variation of continuous Petri nets via model predictive control. In *American Control Conference*, 4919–4924, Portland, Oregon, 2014.
- [TLL14b] M. Taleb, E. Leclercq, and D. Lefebvre. Limitation of flow variation of continuous Petri nets via model predictive control and lyapunov criterion. In *European Control Conference*, 1825–1830, Strasbourg, France, 2014.
- [TLTM05] C. Tolba, D. Lefebvre, P. Thomas, and A. El Moudni. Continuous and timed Petri nets for the macroscopic and microscopic traffic flow modeling. *Simulation Modeling Practice and Theory*, 13(5) :407–436, 2005.
- [VRRS08] C. R. Vazquez, A. Ramirez, L. Recalde, and M. Silva. *Hybrid Systems : Computation and Control*, 4981, chapter On Controllability of Timed Continuous Petri Nets, 528–541. Springer Berlin Heidelberg, 2008.
- [VS09] R.C. Vasquez and M. Silva. Piecewise-linear constrained control for timed continuous Petri net. In *Proceedings of the 48th IEEE Conference on Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference*, 5714–5720, Shanghai, P.R. China, 2009.
- [WCCZ10] N. Wu, C. Chu, F. Chu, and M. Zhou. Hybrid Petri net modeling and schedulability analysis of high fusion point oil transportation under tank grouping strategy for crude oil operations in refinery. *IEEE Transactions on Systems, Man, and Cybernetics-Part C : Applications and Reviews*, 40(2) :159–175, 2010.
- [WCCZ11] N. Wu, C. Chu, F. Chu, and M. Zhou. Schedulability analysis of short-term scheduling for crude oil operations in refinery with oil residency time and charging-tank-switch-overlap constraints. *IEEE Transactions on Automation Science and Engineering*, 8(1) :190–204, 2011.
- [WMJS10] L. Wang, C. Mahulea, J. Júlvez, and M. Silva. Minimum-time control for structurally persistent continuous Petri nets. In *49th IEEE Conference on Decision and Control*, 2771–2776, Atlanta, GA, USA, 2010.
- [WMJS12] L. Wang, C. Mahulea, J. Júlvez, and M. Silva. Control of continuous Petri nets using ON/OFF based method. In *11th IFAC Workshop on Discrete Event Systems*, 45 :47–53, Guadalajara, Mexico, 2012.

- [WMJS14] L. Wang, C. Mahulea, J. Júlvez, and M. Silva. ON/OFF strategy based minimum-time control of continuous Petri nets. *Nonlinear Analysis : Hybrid Systems*, 12 :50–65, 2014.
- [WMS13] L. Wang, C. Mahulea, and M. Silva. Distributed model predictive control of timed continuous Petri nets. In *IEEE 52nd Annual Conference on Decision and Control*, 6317–6322, Firenze, Italy, 2013.
- [XRS06] J. Xu, L. Recalde, and M. Silva. Tracking control of join-free timed continuous Petri net systems. In *2nd IFAC conference on Analysis and Design of Hybrid Systems*, 39 :30–36, Alghero, Italy, 2006.
- [XRS08a] J. Xu, L. Recalde, and M. Silva. Tracking control of join free timed continuous Petri net systems under infinite servers semantics. *Discrete Event Dynamic Systems*, 18(2) :263–283, 2008.
- [XRS08b] J. Xu, L. Recalde, and M. Silva. Tracking control of timed continuous Petri net systems under infinite servers semantics. In *Proceeding of the 17th World Congress. The International Federation of Automatic Control*, 3192–3197, Seoul, Korea, 2008.
- [YaLA96] K. Yamalidou, J. Moody and M. Lemmon, and P. Antsaklis. Feedback control of Petri nets based on place invariants. *Automatica*, 32(1) :15–28, 1996.
- [YRCL03a] H. Yu, A. Reyes, S. Cang, and S. Lloyd. Combined Petri net modelling and AI based heuristic hybrid search for flexible manufacturing systems-part I. Petri net modelling and heuristic search. *Computers and Industrial Engineering*, 44(4) :527–543, 2003.
- [YRCL03b] H. Yu, A. Reyes, S. Cang, and S. Lloyd. Combined Petri net modelling and AI-based heuristic hybrid search for flexible manufacturing systems-part II. heuristic hybrid search. *Computers and Industrial Engineering*, 44(4) :545–566, 2003.

Cette thèse concerne l'élaboration de stratégies de commande prédictive pour certaines classes de systèmes dynamiques continus, discrets et hybrides modélisés par des extensions de réseaux de Petri ad hoc.

Pour les systèmes continus et en vue de limiter la complexité de calcul inhérente à la forme standard de la commande prédictive, plusieurs améliorations sont proposées. Celles-ci permettent de surmonter le problème de "hill climbing" caractéristique des trajectoires obtenues avec certains réseaux de Petri. Elles assurent également la possibilité d'implémenter la commande en temps réel en adaptant l'horizon de prédiction pour réduire la complexité algorithmique. Enfin, elles permettent de limiter la sollicitation des actionneurs tout en garantissant la stabilité asymptotique du système commandé.

Pour les systèmes discrets temporisés et pour éviter l'exploration exhaustive du graphe d'atteignabilité, une méthode de commande est proposée, basée sur la commande prédictive appliquée à une approximation continue du système discret.

Enfin pour les systèmes hybrides, une commande prédictive hybride est développée, inspirée de la commande prédictive continue.

Les performances de ces différentes stratégies de commande sont évaluées et comparées avec différentes simulations numériques.

Mot-clés : *Systèmes dynamiques, Réseaux de Petri, Commande prédictive, Optimisation, Complexité algorithmique, Sollicitation des actionneurs, Stabilité.*

This thesis concerns the development of predictive control strategies for some classes of continuous, discrete and hybrid dynamic systems modeled by specific extensions of Petri nets.

For continuous systems and in order to limit the computational complexity inherent to the standard form of the predictive control, several improvements are proposed. These improvements allow overcoming the problem of hill climbing that characterizes trajectories obtained with some Petri nets. They also ensure the possibility to implement real-time control by adapting the prediction horizon in order to reduce the algorithmic complexity. Finally, they limit actuators solicitation while ensuring the asymptotic stability of the controlled system.

For timed discrete systems and in order to avoid the exhaustive exploration of the reachability graph, a control method is proposed, based on the predictive control applied to a continuous approximation of the discrete system.

Finally for hybrid systems, hybrid predictive control is developed, inspired by the continuous predictive control.

The performance of these different control strategies are evaluated and compared to different numerical simulations.

Keywords : *Dynamic systems, Petri nets, Predictive control, Optimization, Algorithmic complexity, Actuator stress, Stability.*