



HAL
open science

Approche par invariance positive et les techniques de l'intelligence artificielle pour la régulation des carrefours signalisés

Fadi Motawej

► To cite this version:

Fadi Motawej. Approche par invariance positive et les techniques de l'intelligence artificielle pour la régulation des carrefours signalisés. Ordinateur et société [cs.CY]. Université de Technologie de Belfort-Montbéliard, 2012. Français. NNT : 2012BELF0180 . tel-00720655

HAL Id: tel-00720655

<https://theses.hal.science/tel-00720655>

Submitted on 25 Jul 2012

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

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

Université de Technologie de Belfort-Montbéliard
École Doctorale Sciences Pour l'Ingénieur et
Microtechniques

THÈSE

Présentée pour obtenir le grade de

**Docteur de l'Université de Technologie de Belfort-
Montbéliard en Sciences pour l'Ingénieur**
Spécialité : Automatique et Informatique

Présentée le 30-05-2012 par
FADI MOTAJEJ

Approche par invariance positive et les techniques de l'intelligence artificielle pour la régulation des carrefours signalisés

Président : ABDELLAH EL MOUDNI (Professeur, UTBM Belfort)

Rapporteurs : DOMINIQUE SAUTER (Professeur, CRAN)
MOHAMED BENREJEB (Professeur, ENIT Tunisie)

Examineurs : MICHEL BASSET (Professeur, ENSISA)
ABDELAZIZ HAMZAOUI (Professeur, IUT Troyes)

Directeur : RACHID BOUYEKHF (Maître de conférences HDR, UTBM Belfort)

REMERCIEMENTS

Cette thèse n'aurait pas vu le jour sans la confiance du Professeur Abdellah El Moudni, que je veux vivement remercier de m'avoir accueillie dans son équipe de recherche. Son soutien constant et l'intérêt qu'il a porté à mon travail m'ont énormément aidé dans mon apprentissage de la recherche.

J'exprime ma profonde gratitude à Monsieur Rachid Bouyekhf qui m'a encadré pendant ces années. Son professionnalisme, son incroyable capacité de travail, son enthousiasme et son investissement humain resteront pour moi des références, que j'espère atteindre un jour. Je le remercie pour ses conseils scientifiques permanents qui m'ont guidé dans mes recherches, pour sa disponibilité et sa générosité, merci.

Je remercie Monsieur Mohamed Benrejeb, Professeur à l'ENIT Tunis et Monsieur Dominique Sauter, Professeur au Centre de Recherche en Automatique de Nancy (CRAN), de m'avoir fait l'honneur d'être rapporteurs de ce mémoire, et pour l'attention avec laquelle ils ont lu mon rapport.

Mes remerciements vont également aux membres du jury qui ont accepté de consacrer une partie de leur temps pour s'intéresser à mon travail de recherche, Messieurs Michel Basset, Professeur à l'Université de Haute Alsace, Abdelaziz Hamzaoui, Professeur à l'IUT de Troyes, et Abdellah El Moudni, Professeur à l'Université de Technologie de Belfort- Montbéliard.

Mes travaux de thèse ont été effectués au sein du Laboratoire des systèmes et transport SET, c'est pourquoi j'exprime ma sincère gratitude et mes plus profonds respects à l'ensemble du personnel du Laboratoire pour l'accueil et le soutien qu'ils m'ont témoigné, autant d'éléments indispensables à la réalisation de cette thèse.

Je tiens à remercier mes collègues de bureau qui ont animé mes journées avec leur présence dans une ambiance agréable et décontractée, en particulier, Florent, Jérémie et Adnen.

Je remercie de tout coeur mes amis pour leur soutien et encouragements permanents, pour ces moments de détente, de loisir et de plaisir que nous avons partagé ; merci donc à Mohammad Kabalo, Homam Issa, Abdulkader Tabanjat, Yacine Younes, Hussam Haroun et Rania Mkahl.

Je remercie sincèrement mon Université d'origine, Université Tichrine à Lattaquié en Syrie, pour avoir financé mes travaux de recherches durant cette thèse.

Enfin, je réserve mes remerciements les plus chaleureux à ma famille, en particulier Mes chers parents, mon frère et mes sœurs, pour m'avoir soutenu durant toutes ces années. Sans leur aide, ce projet n'aurait pas pu aboutir. Je leur serai toujours redevable de tous les efforts qu'ils ont fournis à mon égard.



A ma mère et mon père
A mon frère et mes sœurs

TABLE DES MATIÈRES

Liste des figures	x
Liste des tableaux	xi
Introduction générale	xv
Chapitre 1 Généralités sur la régulation du trafic dans un carrefour à feux	1
1.1 Introduction	1
1.2 Petit historique	2
1.3 Description d'un carrefour à feux	3
1.3.1 Caractéristiques d'un carrefour à feux	3
1.3.2 Caractéristiques des feux d'un carrefour	6
1.3.3 Découpage en phases	7
1.3.3.1 Approche du découpage en phases	9
1.3.4 L'écoulement du trafic dans un carrefour à feux	11
1.3.5 Boucle de régulation d'une intersection isolée à feux	12
1.3.6 Caractéristiques du mouvement du trafic	13
1.3.7 Infrastructure de mesure	14
1.4 Modélisation des carrefours à feux	15
1.4.1 Cas où le cycle des feux est constant	15
1.4.1.1 Cas d'une intersection à deux phases	15
1.4.1.2 Cas d'une intersection à trois phases	17
1.4.2 Cas où le cycle des feux est variable	18
1.4.3 Domaine de commande admissible	19
1.5 Modes de régulation des carrefours	19

1.6	Classification des différentes stratégies de régulation du trafic dans un carrefour	21
1.6.1	Stratégies de régulation algorithmiques	22
1.6.1.1	Stratégie fixe ou pré-déterminée	22
1.6.1.2	Stratégie semi-adaptative	24
1.6.1.3	Stratégie adaptative ou temps réel	25
1.6.1.4	Conclusion sur les méthodes algorithmiques	27
1.6.2	Stratégies de régulation analytiques	27
1.6.2.1	Travaux de Chang et Lin	28
1.6.2.2	Les travaux de De Schutter et De Moor	32
1.7	Conclusion et objectifs de la thèse	38
Chapitre 2 Approche des inégalités matricielles Linéaires (LMI)		41
2.1	Introduction	41
2.2	Rappels mathématiques et résultats préliminaires	42
2.2.1	Sur l'invariance positive des ensembles	43
2.2.2	Sur les Inégalités Matricielles Linéaires (LMI)	45
2.2.3	Sur l'optimisation sous contraintes inégalités	46
2.3	Position du problème de la commande	48
2.3.1	Cas où le cycle est constant	48
2.4	Solution du problème de la commande	50
2.4.0.1	Sur l'invariance positive de $\Omega(x)$	51
2.4.0.2	Sur l'inclusion $\Omega(x) \subseteq \mathbb{E}_x \cap \mathbb{U}_v$	52
2.4.0.3	Application aux carrefours à feux	59
2.4.0.4	Cas de deux phases	59
2.4.0.5	Cas de trois phases	61
2.4.1	Cas où le cycle est variable	62
2.4.1.1	Cas où $\text{rang}(B) < n$	63
2.4.1.2	Cas où $\text{rang}(B) = n$	64
2.4.1.3	Application aux carrefours à feux	69
2.5	Conclusion	72
Chapitre 3 Approche (A, B)-invariance		73
3.1	Introduction	73
3.2	Rappels mathématiques et résultats préliminaires	74
3.3	Position du problème de la commande	77
3.3.1	Cas où le cycle est constant	77

3.4	Solution du Problème de la commande	79
3.4.0.1	Sur l'invariance positive de \mathbb{E}_x	79
3.4.0.2	Sur l'inclusion $\mathbb{E}_x \subseteq \mathbb{U}_v$	84
3.4.0.3	Application aux carrefours à feux	91
3.4.0.4	Cas de deux phases	91
3.4.0.5	Cas de trois phases	92
3.4.1	Cas où le cycle des feux est variable	93
3.4.1.1	Cas où $\text{rang}(B) < n$	94
3.4.1.2	Cas où $\text{rang}(B) = n$	95
3.4.1.3	Application aux carrefours à feux	100
3.4.1.4	Cas de deux phases	100
3.4.1.5	Cas de trois phases	101
3.5	Conclusion	102
Chapitre 4 Réseaux de neurones pour la prévision des flux du trafic		105
4.1	Introduction	105
4.2	Séries temporelles	106
4.2.1	Modélisation linéaire des séries temporelles mono-variable	108
4.3	Rappels sur les réseaux de neurones artificiels (RNA)	109
4.3.1	Architectures d'un réseau de neurones	110
4.3.2	Apprentissage des réseaux de neurones	113
4.3.3	Les algorithmes d'apprentissage supervisé	116
4.4	Le problème de sur-ajustement (overfitting)	118
4.5	Modélisation non linéaire basée sur les réseaux de neurones	119
4.6	Application des réseaux de neurones à la prévision des flux du trafic	123
4.6.1	Description du site	123
4.6.2	Collecte des données des séries temporelles	124
4.6.3	Structure du modèle : cas du perceptron multi-couches (PMC)	125
4.7	Prévision des flux du trafic par les cartes auto-organisatrices	134
4.7.1	Introduction	134
4.7.2	La carte auto-organisatrice (SOM)	134
4.7.2.1	La prévision des séries temporelles par SOM	137
4.7.2.2	La performance du modèle SOM pour la prévision des flux de trafic	139
4.8	Application de la commande (A, B) -invariance	140
4.9	Conclusion	142

Conclusion générale	143
Annexes	146
Chapitre A Sur la solution positive de l'équation $Bu = 0$	147
Chapitre B La rétro-propagation du gradient	151
Références	161

TABLE DES FIGURES

1.1	Le premier feu de circulation électrique à Cleveland	2
1.2	Intersection à quatre branches avec mouvement tourne-à-gauche à priorité à droite (sans déplacements pédestres)	4
1.3	Courants de flux dans un carrefour élémentaire	4
1.4	Ensemble des éléments d'un tronçon	5
1.5	Exemple de séquence de phases	7
1.6	Conflits primaires	8
1.7	Conflits secondaires	9
1.8	Conflits tertiaires	9
1.9	Deux phases simples : mouvements de tourne-à-gauche peu importants . .	10
1.10	Trois phases : absence de voie spécialisé de tourne-à-gauche	10
1.11	Phasage simple d'un carrefour en T	11
1.12	Fonctionnement à trois phases	11
1.13	Boucle de régulation d'une intersection isolée	13
1.14	L'écoulement du trafic dans un carrefour à deux phases.	16
1.15	L'écoulement du trafic dans un carrefour à trois phases.	17
1.16	Intersection contrôlée par des feux de signalisation	33
2.1	Evolution des files d'attente.	60
2.2	Evolution des files d'attente.	61
2.3	Evolution des files d'attente.	70
2.4	Evolution des files d'attente.	71
3.1	Evolution des files d'attente.	91

3.2	Evolution des files d'attente.	93
3.3	Évolution des files d'attente.	100
3.4	Evolution des files d'attente.	102
4.1	Série temporelle de débit du trafic	107
4.2	Représentation du neurone formel	110
4.3	Réseau non bouclé mono-couche (perceptron)	112
4.4	Le perceptron multi-couches	112
4.5	Réseau récurrent avec neurones cachés	113
4.6	Schéma bloc de l'apprentissage supervisé	115
4.7	Schéma bloc de l'apprentissage non supervisé	115
4.8	Structure du modèle NNAR	120
4.9	Architecture de réseau de neurones pour la prédiction.	120
4.10	La procédure de modélisation	121
4.11	Intersection-Etude de cas.	124
4.12	La fonction d'auto-corrélation partielle (PACF)	126
4.13	Optional caption for list of figures	129
4.14	La visualisation de la prévision à un pas en avant de flux du trafic q_1	130
4.15	La visualisation de la prévision à un pas en avant de flux du trafic q_2	131
4.16	Optional caption for list of figures	131
4.17	Optional caption for list of figures	132
4.18	Exemples d'architecture possibles	135
4.19	La carte auto-organisatrice	135
4.20	Interprétation de l'algorithme SOM.	136
4.21	Différences entre (a) l'approche supervisée et (b) l'approche non-supervisée	139
4.22	Évolution des files d'attente.	141
B.1	Un perceptron multi-couches	151

LISTE DES TABLEAUX

1.1	Les commutations des feux de signalisation.	33
2.1	Variation du feu vert effectif g_{e_2} en fonction du cycle.	60
2.2	Variation des feux verts effectifs (g_{e_1}, g_{e_2}) en fonction du cycle.	62
2.3	Variation de la durée de cycle $c(k)$ et du feu vert effectif g_{e_2} en fonction du cycle.	70
2.4	Variation de la durée de cycle $c(k)$ et des feux verts effectifs g_{e_1}, g_{e_2} en fonction du cycle.	72
3.1	Variation du feu vert effectif g_{e_2} en fonction du cycle.	92
3.2	Variation des feux verts effectifs (g_{e_1}, g_{e_2}) en fonction du cycle.	93
3.3	Variation de $c(k)$ et de g_{e_2} en fonction de k	101
3.4	Variation de $c(k)$ et de g_{e_1}, g_{e_2} en fonction de k	102
4.1	Ordres des modèles des flux d'entrée du trafic	126
4.2	Nombre de neurones cachées dans les modèles neuronaux	127
4.3	Spécification d'apprentissage	128
4.4	Comparaison entre les réseaux initiaux et les réseaux finaux	129
4.5	Évaluation des performances des modèles	133
4.6	Evaluation des performances des modèles SOM	139
4.7	Variation du feu vert effectif en fonction du cycle pendant la période de 7h04 jusqu'à 9h04.	141

LISTE DES PUBLICATIONS

Journals

1. Fadi Motawej, Rachid Bouyekhf, Abdellah El Moudni. A dissipativity-based approach to traffic signal control for an over-saturated intersection. *Journal of the Franklin Institute*, Volume 348, Issue 4, pp. 703-717, 2011,
2. Fadi Motawej, Rachid Bouyekhf, Abdellah El Moudni. Contrôle de congestion des carrefours isolés. *e-STA (e-revue en Science et Technologies de l'Automatique)*, Volume 7, N°2, 2010.

Conférences

1. Fadi Motawej, Rachid Bouyekhf, Abdellah EL Moudni. Contrôle sous contraintes des carrefours isolés. *7ème Conférence Internationale Francophone d'Automatique (CIFA 2012)*, Grenoble, France, 4-6 juillet 2012.
2. F. Motawej, R. Bouyekhf, A. El Moudni. Contrôle sous contraintes d'une intersection sur-saturée par réseaux de neurones. *JD-MACS*, Juin 2011, Marseille, France.
3. Fadi Motawej, Rachid Bouyekhf, Abdellah EL Moudni. A Note on Artificial Intelligence Techniques and Dissipativity-Based Approach in Traffic Signal Control for an Over-Saturated Intersection. *18th IFAC World Congress*, 2011, Università Cattolica del Sacro Cuore, Milano, Italy.
4. Motawej, F. ; Bouyekhf, R. ; El Moudni, A. Design of a real-time controller for an over-saturated intersection using self-organizing maps. *IEEE Forum on Integrated*

- and Sustainable Transportation System (FISTS)*, June 29 2011-July 1 2011, Vienna, Austria.
5. Fadi Motawej, Rachid Bouyekhf, Abdellah EL Moudni. Traffic Control and Queues Management for a Single Intersection. *The 4th International Conference on Integrated Modeling and Analysis in Applied Control and Automation (IMAACA 2010)*, October 13-15, 2010 FES, Morocco.
 6. Motawej, Fadi ; Bouyekhf, Rachid ; El Moudni, Abdellah. Energy-based control for an over-saturated three-phase intersection. *12th IFAC Symposium on Large Scale Systems : Theory and Applications* (2010), Lille-France.

INTRODUCTION GÉNÉRALE

L'intérêt de contrôler les feux de circulation pour éviter ou réduire la congestion dans un carrefour à feux n'est plus à démontrer. En effet, les feux tricolores permettent, en premier lieu, de supprimer certains points conflictuels en partageant dans le temps l'utilisation d'un même espace, la zone de conflit précisément, entre des flux conflictuels. Par le choix des durées de chaque état des feux, et la synchronisation des feux entre eux, ils permettent aussi de gérer l'écoulement de la demande. Cependant, bien que plusieurs recherches dans le domaine soient réalisées, les efforts menés jusqu'à présent laissent encore le problème de la régulation des carrefours complètement ouvert.

Une des difficultés tient à la variabilité de la demande du trafic. La demande de trafic sur une entrée de carrefour varie de façon importante au cours d'une même journée, et une exploitation en temps fixe (phases et cycles fixés) ne permet pas d'offrir un bon niveau de service aux usagers. Il faut donc chercher à adapter l'offre de l'infrastructure à la demande du trafic, selon différents critères, comme le temps d'attente de tel ou tel type d'utilisateur, le nombre d'arrêts, la consommation de carburant, la pollution. D'où la nécessité de systèmes de régulation de trafic plus dynamiques.

De manière schématique, un système de contrôle dynamique du trafic est un système qui dispose de capteurs lui donnant des informations sur l'état du trafic, et notamment, le nombre de véhicules sur certaines rues ou intersections du réseau routier. Les informations obtenues sont alors transmises à un ordinateur qui a pour mission de donner aux commandes les valeurs (durée et synchronisation des feux) permettant de minimiser une certaine fonction exprimant la congestion du réseau routier. Il peut s'agir comme fonction, de la longueur des files d'attente aux intersections, du temps moyen d'attente aux feux, du temps moyen d'attente de la plus longue file d'attente, etc. Dans un système de

contrôle dynamique, la durée des feux serait donc recalculée à intervalle régulier dont le souci d'éviter ou minimiser la congestion.

Cependant, les systèmes de transport sont des systèmes dynamiques sujets à des limitations d'ordre physique et d'autres sont dictées par des impératifs de sécurité. Ces limitations se traduisent dans la majorité des cas par des contraintes sur les variables d'état et/ou la commande. La prise en compte de telles contraintes dans la conception du plan de feux est donc nécessaire. Notre travail s'inscrit dans ce cadre pour la régulation des carrefours isolés.

La régulation du trafic dans un carrefour à feux concerne en général deux objectifs distincts : la fluidification ou la résorption de congestion. Dans le premier cas, on évite de se retrouver dans une situation de trafic très dense en essayant d'ajuster les durées de commutations des feux en fonction de la demande d'affluence au carrefour : c'est une action **a priori**. Dans le second cas, on est confronté à un trafic saturé (état de congestion). Dans ce cas, il faudra agir **a posteriori**.

Dans ce mémoire, nous nous intéressons essentiellement à un travail en amont (action a priori) permettant d'éviter la congestion en forçant les files d'attente à ne pas dépasser le niveau du trafic correspondant à l'optimum opérationnel des lignes. Plus précisément, nous proposons des commandes par retour d'état basées sur le concept d'invariance positive des ensembles et permettant d'atteindre notre objectif. Le principe de cette approche implique que toutes les trajectoires issues d'un ensemble y restent.

Plan de la thèse

Ce mémoire est structuré en quatre chapitres.

Le premier chapitre présente tout d'abord les carrefours à feux ainsi que les différents éléments les caractérisant. Les notions essentielles pour la régulation du trafic aux abords des carrefours sont ensuite présentées. Une deuxième partie est consacrée à la modélisation des carrefours signalisés à deux phases et à trois phases dans le cas où le cycle des feux est constant ou variable. Une dernière partie est consacrée à la présentation, non exhaustive, des méthodes développées pour la gestion et la régulation des carrefours. Les résultats relativement comparables aux nôtres sont présentés de façon détaillée afin de dégager clairement notre contribution. Enfin, nous terminons le chapitre par une conclusion et la formulation du problème de la commande de ce travail.

Le deuxième chapitre est consacré à la solution de notre problème de la commande par l'approche des inégalités matricielles Linéaires (LMI). Après avoir exposé quelques définitions et résultats utiles donnés dans la littérature dont nous aurons besoin dans la

suite, nous donnons sous forme LMI des conditions nécessaires et suffisantes pour qu'un ensemble ellipsoïdal soit à la fois un ensemble positivement invariant de notre système et à la fois inclus dans l'ensemble des contraintes sur l'état et la commande. Ces résultats sont appliqués aux modèles de transport établis dans le chapitre 1. Deux cas sont étudiés : celui dont le cycle des feux est constant est d'abord analysé, ensuite, nous nous intéressons au cas où le cycle des feux est variable afin d'adapter la stratégie de commande de manière efficace aux variations permanentes du trafic.

Le caractère restreint des domaines ellipsoïdaux établis dans le chapitre 2 nous a amené à avoir recours à l'approche (A, B) -invariance pour la solution de notre problème de commande dans le chapitre 3. Après avoir exposé quelques résultats utiles, nous donnons sous forme de relations matricielles linéaires des conditions nécessaires et suffisantes pour que l'ensemble des contraintes sur l'état soit un domaine candidat à l'invariance en boucle fermée au lieu de construire un domaine ellipsoïdal candidat. Comme dans le chapitre 2, ces résultats sont ensuite appliqués aux modèles de transport établis dans le chapitre 1 pour le cas où le cycle des feux est constant ou variable.

Le chapitre 4 se focalise sur l'apport que pourrait avoir les techniques de l'intelligence artificielle pour la prévision des flux d'entrée au carrefour afin de garantir la faisabilité en temps réel des commandes proposées. Dans ce sens, les réseaux de neurones sont utilisés. Ainsi, deux modèles neuronaux sont proposés pour la prévision. D'abord le perceptron multi-couches et ensuite la carte auto-organisatrice. Dans une dernière partie, ces travaux sont appliqués sur un exemple en utilisant les données d'un carrefour réel du boulevard Anatole France de la ville de Belfort afin de montrer l'intérêt des résultats proposés.

Enfin, nous concluons ce mémoire en résumant les contributions de ce travail et en présentant quelques perspectives de nos travaux.

CHAPITRE 1

GÉNÉRALITÉS SUR LA RÉGULATION DU TRAFIC DANS UN CARREFOUR À FEUX

1.1 Introduction

Au cours des dernières années, le développement de la circulation routière a connu une augmentation majeure. L'amélioration du niveau de vie moyen et du taux d'équipement des ménages a permis au plus grand nombre d'accéder au déplacement en véhicule particulier. Dans le même temps, le trafic routier de marchandises a connu une progression aussi vive. Nous avons donc assisté à une course entre la croissance du trafic routier et les progrès quantitatifs et qualitatifs de la voirie. Cette quantité d'actions génère des problèmes au niveau de la fluidité du trafic, d'où l'apparition des phénomènes de congestion.

La congestion des réseaux routiers est un des problèmes majeurs de la circulation. Elle se produit de façon quasi-quotidienne. Elle est source de perte de temps, d'augmentation de la consommation d'énergie, de la nuisance et de la détérioration de l'environnement. La solution aux problèmes de congestion routière ne passe pas toujours par la construction de nouvelles routes, comme cela se faisait dans un passé récent. La tendance actuelle est plutôt à une meilleure utilisation des infrastructures existantes.

La régulation du trafic a vu le jour avec l'augmentation exponentielle du nombre de véhicules circulant dans les zones urbaines. Hormis la congestion issue de cette crois-

sance, l'insécurité routière a fait prendre conscience de la nécessité d'avoir recours à des méthodes de régulation de flux de véhicules.

Dans la suite de ce chapitre, nous décrivons d'une manière détaillée un carrefour à feux. Ceci passe par la définition des différents éléments le caractérisant. Ensuite, nous présentons deux modèles sous forme d'équation d'état d'un carrefour à deux phases et à trois phases. Deux cas sont étudiés : celui où le cycle des feux est constant et le cas où le cycle des feux est variable. Une fois que les modèles sont établis, nous introduisons les notions essentielles pour la régulation du trafic aux abords des carrefours. Une deuxième partie est consacrée à la présentation, non exhaustive, des méthodes développées pour la gestion et la régulation des carrefours. Les résultats relativement comparables aux nôtres sont présentés d'une façon détaillée afin de dégager clairement notre contribution.

1.2 Petit historique

Le premier feu de circulation utilisé dans le domaine remonte au sémaphore à commande manuelle utilisé à Londres dès 1868 [1]. Le premier feu de circulation électrique aux Etats-Unis a été développé par James Hoge et installé à Cleveland, Ohio, en 1914 [2] (Figure 1.1). Ceci a été suivi par l'introduction de feux inter-connectés en 1917 dans



FIGURE 1.1 – Le premier feu de circulation électrique à Cleveland

la ville de Salt Lake, Utah [3]. A ce moment, la propriété et l'utilisation de l'automobile étaient en pleine expansion et l'efficacité du contrôle des feux de circulation a été reconnue. La circulation automobile croissante dans les rues a conduit au développement et à l'implémentation des signaux actionnés en 1928 [4].

Depuis, les technologies de contrôle des feux de circulation ont connu une croissance rapide et les feux de circulation sont devenus un élément essentiel des systèmes de contrôle et de gestion du trafic moderne d'aujourd'hui. Du point de vue de l'application, les feux de circulation sont utilisés pour contrôler l'affectation du droit de passage des véhicules ou des piétons aux endroits où des conflits potentiellement dangereux existent, ou lorsque des dispositifs passifs (panneaux, marquages) ne fournissent pas un contrôle adéquat du trafic.

1.3 Description d'un carrefour à feux

Le carrefour à feux représente un espace de convergence de différents acteurs du transport. C'est un nœud important dans le réseau de transport urbain permettant d'écouler ou de stocker des flux de véhicules selon la variation du trafic. Son fonctionnement dépend de divers facteurs liés à la géométrie de l'aménagement et à son environnement. Ainsi, il constitue un terrain sensible pour les conflits de trafic et en même temps, il est considéré comme un levier important en matière de régulation du trafic urbain.

1.3.1 Caractéristiques d'un carrefour à feux

D'après la définition donnée par le livre de référence en matière d'ingénierie du trafic routier [5], un carrefour est situé à la rencontre de plusieurs rues déterminant des couloirs d'entrée et de sortie. La mise en place d'un système de feux à un carrefour réalise une séparation dans le temps de l'admission de différents flux de véhicules.

Pour étudier une intersection routière, nous nous limitons tout d'abord à l'analyse d'une intersection routière à deux phases et à quatre branches que nous pouvons qualifier comme étant un carrefour *élémentaire* (voir figure 1.2). L'étude des autres types d'intersection sera ramenée à ce type simple de carrefour. Ceci peut être réalisé, soit en décomposant ceux qui sont complexes en plusieurs carrefours *élémentaires*, soit en ajoutant des tronçons fictifs pour les intersections à moins de quatre branches.

Dans chaque intersection, nous pouvons identifier trois zones fonctionnelles :

- **Zone de conflit** : il s'agit de l'espace de croisement des routes. Autrement dit, c'est la ressource critique partagée par tous les véhicules qui traversent l'intersection ;
- **Zone de stockage** : en amont de la zone de conflit, elle est l'entrée empruntée par les véhicules ;
- **Zone de sortie** : en aval de la zone de conflit, elle permet le soulagement de cette zone.

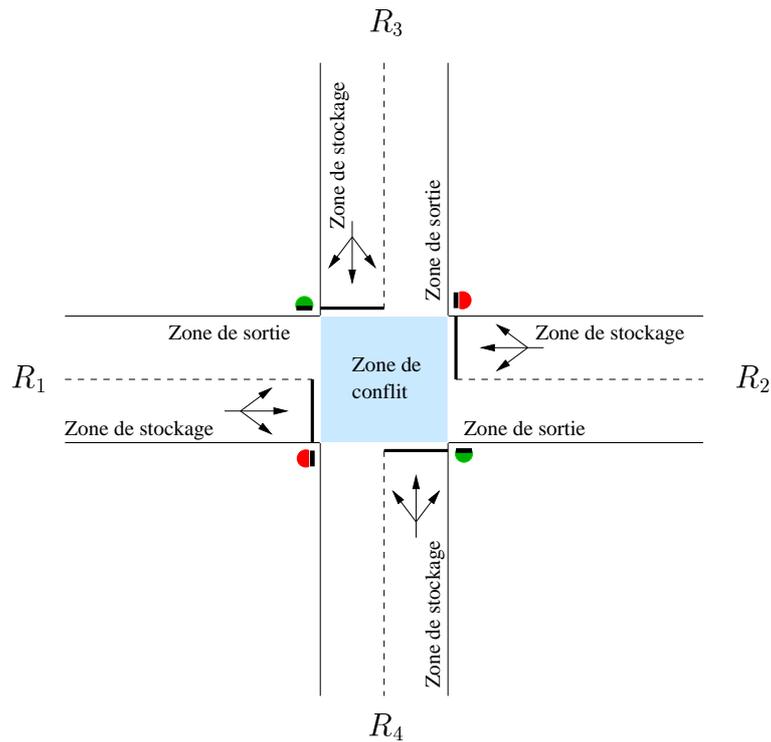


FIGURE 1.2 – Intersection à quatre branches avec mouvement tourne-à-gauche à priorité à droite (sans déplacements pédestres)

Pour bien comprendre le fonctionnement d'un carrefour, un certain nombre d'éléments doivent être définis :

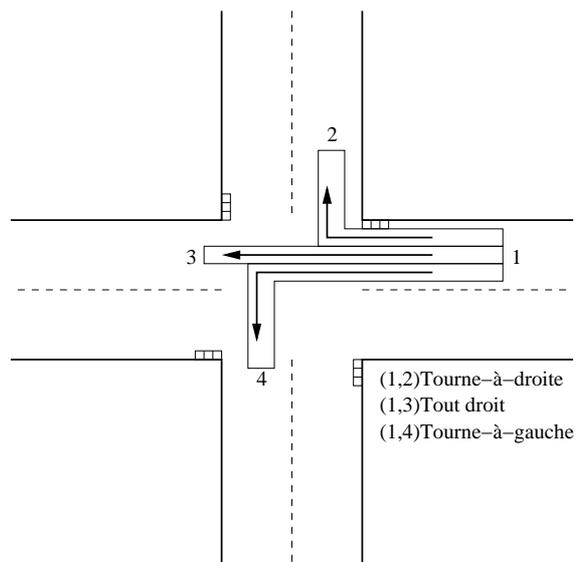


FIGURE 1.3 – Courants de flux dans un carrefour élémentaire

Couloir et tronçon : Un couloir est caractérisé par sa largeur, son nombre de voies ainsi que par les sens de circulation ; certaines de ces voies peuvent être affectées à un

courant particulier appelées *voies spéciales*. Les courants de véhicules sont soit *directs*, soit des courants de *tourne-à-gauche* ou *tourne-à-droite* (voir figure 1.3). Une voie spéciale pourrait être affectée à un courant *tourne-à-gauche* ou *tourne-à-droite*. Un tronçon d'un carrefour représente chaque couloir d'entrée et de sortie. Ainsi, il peut contenir plusieurs voies. Chaque voie écoule un mouvement de véhicules unidirectionnel. Un tronçon est caractérisé par plusieurs paramètres, entre autres, sa longueur qui représente la mesure en mètres de la distance séparant la ligne géométrique où sont implantés les feux associés au tronçon (cette ligne est considérée comme la sortie du tronçon) et la ligne géométrique en amont considérée comme l'entrée de ce tronçon (voir figure 1.4)

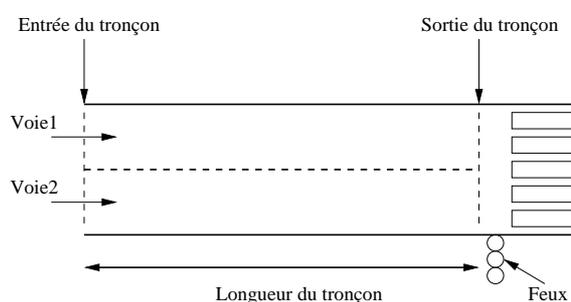


FIGURE 1.4 – Ensemble des éléments d'un tronçon

Concentration maximale et capacité de stockage d'un tronçon : La concentration maximale d'un tronçon est le nombre maximal de véhicules que l'on peut stocker par mètre linéaire dans le tronçon. Si un tronçon j est composé de deux voies et, sachant que la concentration d'une voie est de 0.2véh/m , alors la concentration de ce tronçon est de $2*0.2=0.4\text{véh/m}$. La capacité de stockage d'un tronçon représente le nombre maximal de véhicules que l'on peut stocker sur toute sa longueur. En d'autres termes, la capacité représente la somme des concentrations des voies composant le tronçon multipliée par la longueur du tronçon.

Débit de saturation d'une entrée : Le débit de saturation à l'entrée d'un carrefour à feux est le nombre maximum de véhicules pouvant utiliser le tronçon sans interruption par unité de temps, le feu étant continuellement au vert. C'est un paramètre de dimension temporelle. Pour chaque voie, il est exprimé en unités de véhicules particuliers par heure de feu vert. Ainsi, par exemple, on adopte souvent les valeurs de référence suivantes : $s=1820\text{ uvp/h/voie}$ dans une agglomération importante, $s=1650\text{ uvp/h/voie}$ dans une petite agglomération. Ces valeurs peuvent être ajustées en fonction de divers facteurs liés à la géométrie de l'aménagement du carrefour et à son environnement (taille de l'agglomération, visibilité,...).

1.3.2 Caractéristiques des feux d'un carrefour

En raison de l'insécurité qui existe à l'approche des carrefours, notamment ceux des grandes agglomérations, de nombreux carrefours sont équipés de feux tricolores. Notons que l'implantation de ces feux doit respecter la géométrie d'un carrefour.

Etats des feux : Un feu tricolore est un signal lumineux commandant le passage libre (*feu vert*), toléré (*feu orange*) ou interdit (*feu rouge*) du trafic des véhicules. Ces différentes couleurs sont nommées *les états de feux*. Le changement de feu d'un état à l'autre est appelé *commutation*. Pour la sécurité des usagers, le temps de passage par état doit au moins durer une période minimale, fixée par les exploitants. Contrairement aux durées du feu vert ou du feu rouge, la durée du feu orange est invariable et est fixée à trois secondes.

Rappelons quelques concepts essentiels de la signalisation à feux :

Cycle : Les indicateurs des feux - vert, orange, rouge - se succèdent à l'intérieur d'un cycle, défini comme étant la durée constante séparant deux passages successifs de l'ensemble des signaux par le même état.

Phase : Le cycle est partagé en phases, temps pendant lequel un ou plusieurs courants sont admis dans le carrefour. Une phase est dite spéciale quand elle a pour seul but de favoriser l'écoulement d'un mouvement tournant. Elle est dite **saturée** lorsqu'un véhicule au moins de cette phase est contraint d'attendre plus d'un cycle pour franchir le carrefour. Le carrefour est lui-même saturé quand au moins une des ses phases est saturée.

Vert effectif : On appelle temps de vert effectif, la somme du temps de vert réel et du temps de l'orange (qui est souvent de 3 ou 5 secondes selon la vitesse d'approche dans le cas d'un carrefour urbain ordinaire).

Vert utile : On définit le temps de vert utile, en retranchant du temps de vert effectif le temps perdu (somme des temps perdus au début du vert, c'est-à-dire au démarrage, et du temps perdu en fin de phase). Les temps perdus par phase se situent en moyenne autour de 8 secondes [6].

Rouge utile : On appelle rouge utile la différence entre la durée du cycle est celle du vert utile. Pour des raisons de sécurité, à la fin de chaque phase, l'ensemble des feux du carrefour est maintenu au rouge pendant une certaine durée, appelée **rouge intégral**.

Dans le cas d'un carrefour simple à 4 branches par exemple, on adopte une durée de rouge intégral de 2 secondes.

Le plan des feux d'un carrefour signalisé : Considérons l'intersection montrée dans la figure 1.2. On constate qu'il y a deux paires de courants en conflit, à savoir la paire de courants provenant des rues R_1 et R_2 et la paire de courants provenant des rues R_3 et R_4 . En supposant que les mouvements dominants sont les mouvements directs (aller tout droit) et que les mouvements de tourne-à-gauche sont soumis à la priorité à droite, nous pouvons découper le cycle en 2 phases, i.e. Phase I et Phase II.

Supposons que les durées des phases I et II soient identiques et que la séquence des phases soit fixe. Les feux installés aux rues R_1 et R_2 sont d'abord verts, puis rouges pour que les feux aux rues R_3 et R_4 deviennent à leur tour verts ; et ainsi de suite. Les deux phases se succèdent, constituant un cycle. Le plan des feux pour cette intersection est illustré dans la figure 1.5. Pour des raisons de sécurité, à la fin de chaque phase, l'ensemble des feux de l'intersection est maintenu au rouge pendant une certaine durée, appelée **rouge intégral**. La stratégie de régulation dépend de l'état du trafic estimé grâce aux mesures réalisées. Ainsi, la fluidité du trafic est généralement la mission du système de régulation qui décide de la durée de chaque phase et de la séquence des phases.

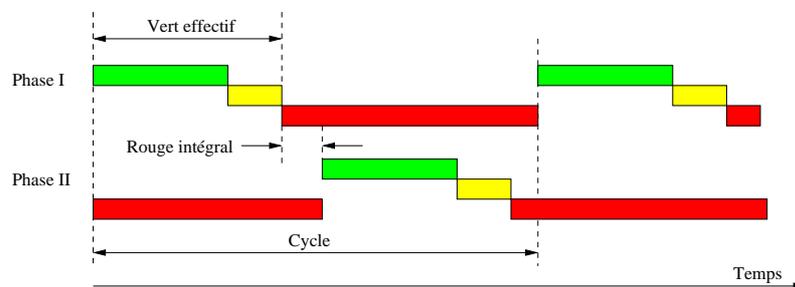


FIGURE 1.5 – Exemple de séquence de phases

Ainsi, l'équation qui décrit le plan des feux dans cet exemple est donnée par :

$$cycle = \text{Vert effectif}_{\text{Phase I}} + \text{Vert effectif}_{\text{Phase II}} + \text{Rouge intégral} \quad (1.1)$$

Notons que dans la suite de cette thèse, nous négligerons le rouge intégral puisqu'il est fixe et indépendant de la durée du cycle et du vert effectif. Ceci facilite d'une part les calculs et d'autre part n'a aucune influence sur les résultats obtenus par notre approche.

1.3.3 Découpage en phases

La dynamique du trafic dans un carrefour à feux peut être découpée en un nombre fini de flux (phases) interagissant entre eux. Ce point joue un rôle essentiel dans toute analyse

du fonctionnement d'un carrefour. En effet, l'objectif est de parvenir à un fonctionnement optimum, minimisant le nombre de phases ainsi que la charge de chacune d'entre elles.

Afin de définir proprement le découpage en phases, il est nécessaire de définir la notion de conflit.

Notions de conflits

L'activité aléatoire du trafic dans un carrefour simple ou à feux génère des situations de conflit. Un point conflictuel résulte de la rencontre, en un même lieu, d'au moins deux mouvements de mobiles avec un angle non nul. La zone de conflit d'une intersection quelconque est constituée de l'ensemble des points conflictuels. Certains trouvent une solution dans le respect du code de la route. Cependant, ils doivent être examinés avec soin en fonction de la visibilité et de l'importance respective des mouvements (Cf. la réglementation en vigueur dans [7]). D'autres doivent être systématiquement éliminés dès lors que l'on implante une signalisation lumineuse tricolore verticale. Nous pouvons classer généralement les situations de conflit en trois grandes catégories.

Conflits primaires : Ces conflits sont issus de la rencontre de deux mouvements sécants. Soit une rencontre véhicule-véhicule soit véhicule-piéton. La figure 1.6 illustre les deux situations conflictuelles de types primaires.

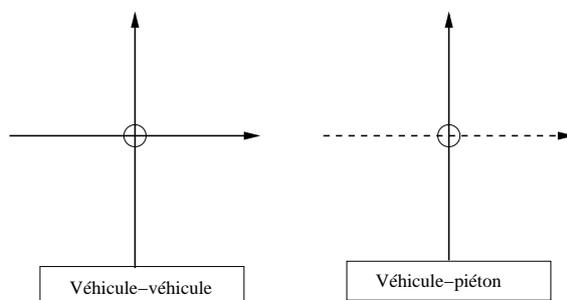


FIGURE 1.6 – Conflits primaires

Conflits secondaires : Ces types de conflits ne sont pas interdits, mais ils constituent un obstacle gênant. Pour remédier à ces situations conflictuelles, nous adoptons les règles de code de la route qui définissent les priorités respectives. La figure 1.7 donne une explication de ces types de conflits.

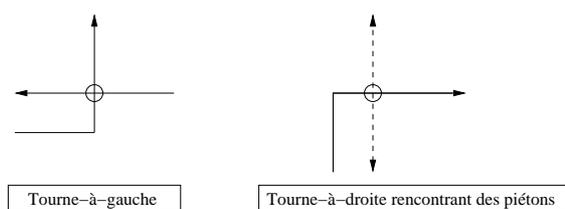


FIGURE 1.7 – Conflits secondaires

Conflits tertiaires : Nous pouvons dire que cette catégorie n'est pas vraiment une situation de conflit. Cependant, dans certains cas précis, nous pouvons être amenés à séparer les courants. En effet, c'est une sorte de séparation entre les différents types de véhicules (voir figure 1.8). Par exemple, on donne la priorité à des voies de haute fréquentation.

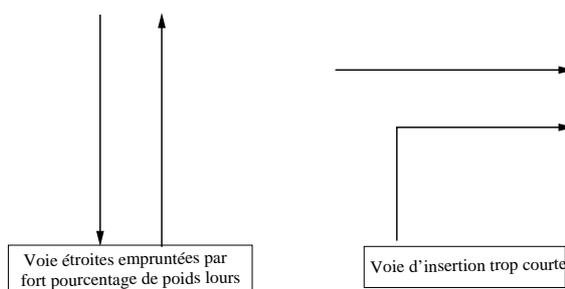


FIGURE 1.8 – Conflits tertiaires

Toutes ces différentes situations conflictuelles peuvent être regroupées dans une matrice, appelée *matrice des conflits*. C'est une matrice regroupant les différentes situations conflictuelles dans un carrefour. Elle est obtenue à partir du repérage préalable des différents mouvements traversant le carrefour. Les mouvements des piétons sont aussi pris en compte au même titre que ceux des véhicules. Pour plus de détails sur la matrice de conflits, nous invitons le lecteur à consulter la référence [8].

1.3.3.1 Approche du découpage en phases

Pour obtenir un découpage en phases des différents mouvements traversant un carrefour, plusieurs approches, théoriques ou empiriques, permettent de le faire. Les méthodes théoriques, utilisant les données fournies par la matrice des conflits, font appel à la théorie des graphes. Quant à la démarche empirique, elle est basée sur l'observation du comportement moyen de l'automobiliste.

D'une manière générale, on s'efforce d'élaborer un phasage qui :

- supprime ou minimise les conflits ;

- prend en compte les piétons : cheminement, temps de traversée, sécurité ;
- tient compte de la géométrie existante.

Cas des carrefours à quatre branches : a priori, on cherche à se rapprocher d'un fonctionnement à 2 phases (figure 1.9), en utilisant au mieux les possibilités d'adaptabilité des matériels.

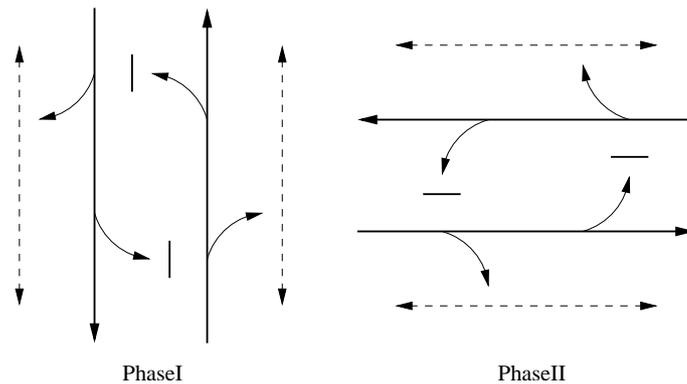


FIGURE 1.9 – Deux phases simples : mouvements de tourne-à-gauche peu importants

Si le découpage précédent ne permet pas d'assurer une sécurité et une fluidité suffisante, on examine dans ce cas un fonctionnement à 3-phases avec un mouvement tourne-à-gauche (voir figure 1.10).

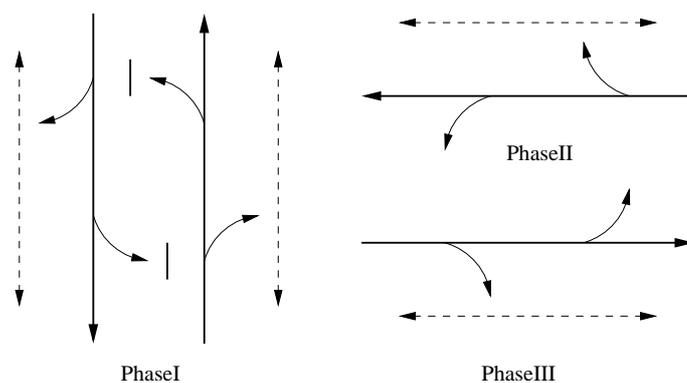


FIGURE 1.10 – Trois phases : absence de voie spécialisée de tourne-à-gauche

En dernier ressort, si ce découpage ne satisfait pas nos besoins, on examine donc le cas d'un découpage à 4-phases. Toutefois, une telle pratique est peu souhaitable en raison des temps d'attente imposés.

Cas des carrefours à trois branches : Tout d'abord, on cherche un fonctionnement à 2-phases selon le schéma suivant (figure 1.11) :

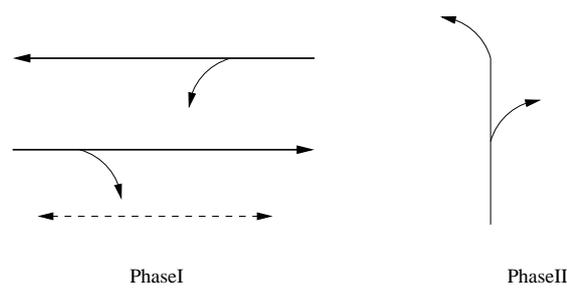


FIGURE 1.11 – Phasage simple d'un carrefour en T

Ce type de carrefour correspond généralement à l'intersection d'une voie principale supportant un important trafic et d'une voie secondaire à plus faible circulation.

Si les mouvements tourne-à-gauche sont importants et s'il existe une voie spécialisée, on peut envisager un fonctionnement à 3-phases comme le montre la figure 1.12.

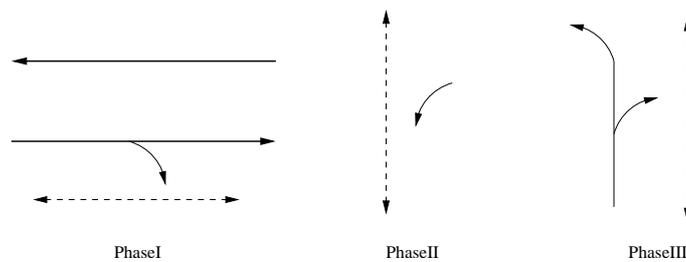


FIGURE 1.12 – Fonctionnement à trois phases

1.3.4 L'écoulement du trafic dans un carrefour à feux

Généralement, on associe à chaque carrefour une liste des entrées et une liste des sorties. Ces listes sont décrites par des mouvements. Chaque mouvement est défini comme étant un parcours de véhicules allant d'une entrée vers une sortie. Ainsi, par exemple, pour un carrefour à feux simples à quatre branches (*élémentaire*, voir figure 1.3), nous distinguons trois mouvements de flux pour le tronçon 1 :

- (1,2) mouvement tourne-à-droite
- (1,3) mouvement direct
- (1,4) mouvement tourne-à-gauche

Il faut noter que le mouvement tourne-à-gauche pose une complexité majeure pour analyser le comportement des différents flux qui entrent dans un carrefour. Par exemple, les véhicules provenant d'une entrée du carrefour qui suivent le mouvement tourne-à-gauche (1,4) perçoivent devant eux les véhicules du mouvement direct (3,1), le mouvement tourne-à-droite (3,4) et le mouvement tourne-à-gauche (3,2). Donc, le mouvement tourne-à-gauche (1,4) doit attendre jusqu'à ce que la voie soit libre (la priorité à droite). En même temps,

le mouvement tourne-à-gauche (3,2) doit aussi céder la priorité à droite des mouvements direct (1,3), tourne-à-droite (1,2) et tourne-à-gauche (1,4).

Dans un carrefour élémentaire (voir figure 1.3), il existe environ 12 mouvements. Si nous considérons que l'écoulement du trafic dans un carrefour suit une modélisation macroscopique ; cela signifie que le mouvement est représenté par un segment dont les caractéristiques géométriques sont celles de l'aire qu'occupent les véhicules. Pour chaque mouvement, on définit les grandeurs suivantes :

- la longueur du mouvement représentant le parcours moyen des véhicules.
- la vitesse des véhicules.
- la capacité de stockage des véhicules.

La gestion de l'écoulement du trafic s'effectue à travers la **boucle de régulation de l'intersection** (voir figure 1.13). Cette boucle est constituée de deux parties qui sont l'intersection et le **système de feux de signalisation** associé. L'intersection est caractérisée par sa géométrie, les véhicules qui s'y déplacent, les mouvements autorisés et les volumes correspondants. Le système de feux de signalisation associé est choisi en fonction de la politique de régulation de la ville, tout en assurant la sécurité et la fluidité du trafic.

1.3.5 Boucle de régulation d'une intersection isolée à feux

Le système de feux de signalisation associé à une intersection urbaine est généralement un système complexe qui ne se limite pas aux signalisations verticales et horizontales visibles par les conducteurs. En effet, des instruments de mesure, des postes de contrôle, un central, des liaisons filaires ou GPRS, etc. sont souvent requis pour assurer le fonctionnement de la régulation. Si nous limitons notre étude à une intersection isolée, nous pouvons au moins identifier trois éléments qui composent son système de feux de signalisation (voir figure 1.13). L'intégration de ces trois éléments à une intersection constitue la boucle de régulation du trafic à son niveau. Ces éléments sont :

- **Les dispositifs de signalisation** : Il s'agit de l'ensemble des signaux verticaux et horizontaux transmis aux conducteurs pour assurer la sécurité de passage des véhicules. Ils visent à éviter la rencontre des mouvements conflictuels. Dans le cadre d'une intersection isolée à feux, le système de signalisation verticale se distingue par la présence de feux de signalisation. Par la suite, nous limiterons notre étude aux feux tricolores ;
- **Les infrastructures de mesure** : Elles reflètent l'état du trafic à travers ses différentes grandeurs. Ces grandeurs sont mesurées par des capteurs installés au niveau de l'infrastructure ;

- **Le régulateur** : Il est le cerveau de la boucle de régulation. Il précise en temps réel les stratégies de régulation, en fonction des données issues de l'infrastructure de mesure, afin d'atteindre des objectifs bien précis, comme minimiser la longueur des files d'attente, le temps d'attente des véhicules, etc.

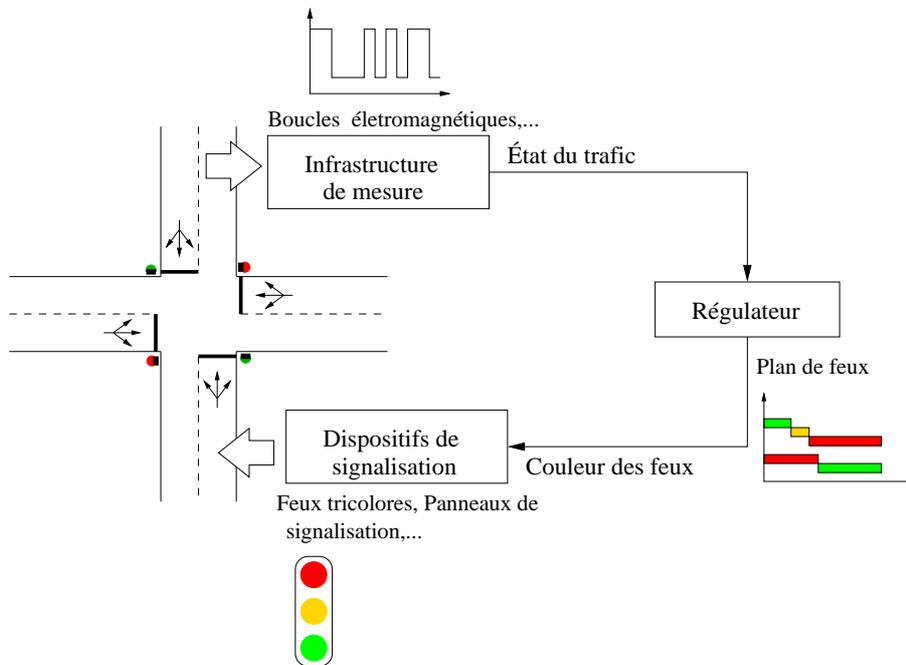


FIGURE 1.13 – Boucle de régulation d'une intersection isolée

1.3.6 Caractéristiques du mouvement du trafic

On distingue trois grandeurs principales caractérisant les mouvements du trafic dans un carrefour à feux.

Débit des flux : C'est le nombre de véhicules par unité de temps franchissant une ligne d'entrée d'un tronçon. Nous partageons ce débit en deux parties ; le débit d'entrée et le débit de sortie. Notons que, contrairement aux arrivées, les départs sont dépendants des durées de commutation de l'ensemble des feux d'un carrefour.

Charge d'un carrefour : La charge Y d'une entrée est définie comme le rapport entre le débit d'entrée mesuré q et le débit de saturation s . C'est donc le rapport de la demande sur l'offre :

$$Y = \frac{q}{s} \quad (1.2)$$

Cette définition s'étend à l'ensemble du carrefour. La charge globale du carrefour est la somme des charges de chacune des phases :

$$Y = \sum y_p \quad (1.3)$$

La somme étant effectuée sur le nombre n de phases. y_p représente la charge du courant prépondérant de chaque phase définie par $y_p = \text{Max}(y_c)$, où y_c définit la charge des courants admis simultanément dans le carrefour au cours de la phase. Chaque phase se caractérise donc par une charge correspondant à celle de son entrée prédominante.

Remarque 1.1 Si le courant directionnel ayant la plus forte charge est commun à 2 phases, on prend la charge immédiatement inférieure. Toutefois, il convient de s'assurer que la somme de la charge immédiatement inférieure et de la charge prédominante de l'autre phase est supérieure à la charge du courant utilisant les 2 phases. Dans le cas contraire, c'est cette dernière que l'on utilise. ►

Longueur des files d'attente : cette variable correspond au nombre de véhicules arrêtés derrière la ligne d'arrêt à un feu de circulation. La longueur des files d'attente constitue l'une des grandeurs majeures de l'évaluation d'une stratégie de régulation.

1.3.7 Infrastructure de mesure

Il existe de nombreux types de capteurs permettant les mesures directes ou indirectes des variables de circulation. Les deux principaux types de capteurs que l'on peut rencontrer [9], [10] sont :

- Les capteurs enfouis dans la chaussée :
 - Les boucles électromagnétiques : c'est aujourd'hui le dispositif de mesure des paramètres du trafic le plus répandu dans le monde, tant en ville que sur les voies rapides et les autoroutes urbaines. Il permet de mesurer les débits, le taux d'occupation, la vitesse et l'écart de temps inter-véhiculaire. Le capteur est constitué d'une boucle inductive, noyée dans le revêtement de la chaussée. Le passage de la masse métallique d'un véhicule au-dessus de la boucle provoque une variation du champ électromagnétique. Cette variation se traduit par un créneau de tension dont la longueur est liée à celle du véhicule et du capteur et au temps de passage.
- Les détecteurs latéraux :
 - Capteurs à ultrasons et à effet Doppler : ce capteur est constitué d'une antenne fixée sur un portique, lequel est positionné dans l'axe de la voie de circulation. Cette antenne émet constamment des ondes qui se propagent à une vitesse

- connue. Lors du passage d'un véhicule, l'onde émise rencontre un obstacle et est réfléchiée. A partir de la différence entre les fréquences de l'onde émise et de celle reçue, la vitesse du véhicule peut être estimée.
- Capteurs optiques (vidéo) : le principe de ces capteurs consiste à utiliser une caméra vidéo et à traiter de manière automatique les images fournies par cette caméra afin de déduire les paramètres du trafic [11], [12].

1.4 Modélisation des carrefours à feux

La mise en place d'un système de feux à un carrefour réalise une séparation dans le temps de l'admission de différents courants de véhicules. Généralement, pour étudier une intersection routière, on se limite tout d'abord à l'analyse d'une intersection routière à deux ou à trois phases et à quatre branches. Ce n'est pas parce que l'étude des intersections de natures très diverses ne s'impose pas aussi bien dans les applications, c'est surtout parce que, comme dans toutes les sciences, on s'est attaché d'abord à ce qui était plus simple et plus facile. L'étude des autres types d'intersections sera ramenée à ce type simple de carrefour, soit en décomposant ceux qui sont complexes en plusieurs carrefours simples, soit en ajoutant des tronçons fictifs pour les intersections à moins de quatre branches.

Généralement, on associe à chaque carrefour des listes d'entrées et de sorties. Ces listes sont décrites par des mouvements. Chaque mouvement est défini comme étant un parcours des véhicules allant d'une entrée vers une sortie. Les courants de véhicules sont soit directs, soit des courants de tourne-à-gauche ou tourne-à-droite (voir plus haut).

1.4.1 Cas où le cycle des feux est constant

1.4.1.1 Cas d'une intersection à deux phases

Considérons maintenant l'intersection de la figure 1.14. Notons x_i la file d'attente de chaque tronçon i . Comme les tronçons 1 et 2 de la phase 1 ont les mêmes caractéristiques, nous nous intéressons uniquement au tronçon 1. De même pour la phase 2, nous nous intéressons uniquement au tronçon 3.

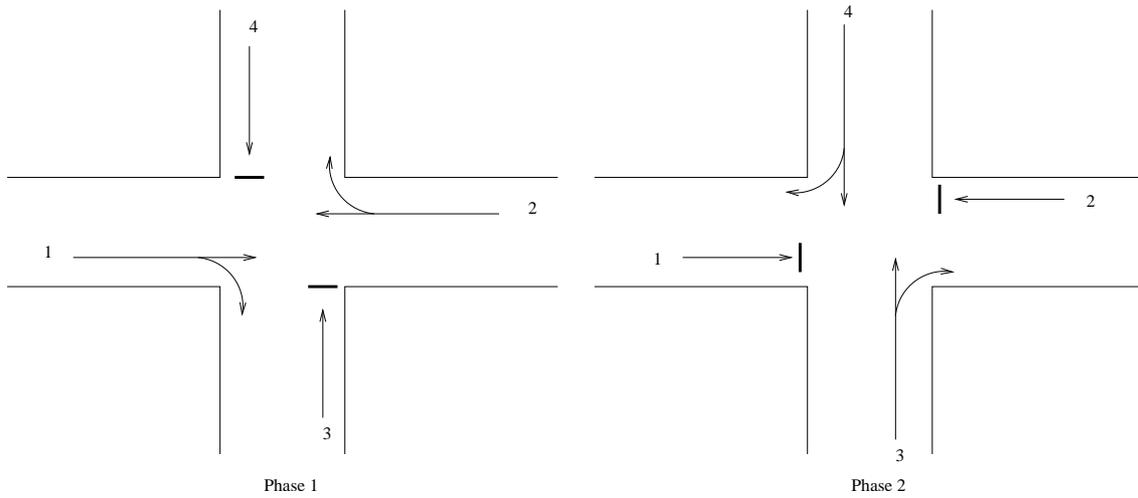


FIGURE 1.14 – L'écoulement du trafic dans un carrefour à deux phases.

L'idée fondamentale de la modélisation du système est basée sur le principe de conservation des flux du trafic. Cela signifie que la file d'attente du cycle $k + 1$ est égale à la somme de la file d'attente du cycle k et du nombre des arrivées $E_i(k) = q_i c$, moins le nombre des départs $D_i(k) = s_i g_{e_i}(k)$ pour la phase i . Ainsi, soit $x_i(k)$ la longueur de la file d'attente dans chaque phase au début du cycle k , l'équation de récurrence modélisant les files d'attente pour une succession de cycles pendant une période est donnée par :

$$x_i(k + 1) = x_i(k) + E_i(k) - D_i(k) \quad (1.4)$$

Ceci conduit alors pour chaque phase à l'équation suivante :

$$x_1(k + 1) = x_1(k) + q_1 c - s_1 g_{e_1}(k)$$

$$x_2(k + 1) = x_2(k) + q_2 c - s_2 g_{e_2}(k)$$

De là et du fait que $g_{e_1} + g_{e_2} = c$, on déduit aussitôt :

$$x_1(k + 1) = x_1(k) + (q_1 - s_1)c + s_1 g_{e_2}(k) \quad (1.5)$$

$$x_2(k + 1) = x_2(k) + q_2 c - s_2 g_{e_2}(k) \quad (1.6)$$

Cette équation peut s'écrire sous la forme matricielle suivante :

$$x(k + 1) = x(k) + Bu(k) + L \quad (1.7)$$

où $x(k)$ est le vecteur d'état du système, $u(k) = g_{e_2}(k)$ est la variable de commande et :

$$B = \begin{pmatrix} s_1 \\ -s_2 \end{pmatrix}; \quad L = \begin{pmatrix} (q_1 - s_1)c \\ q_2 c \end{pmatrix}$$

Remarque 1.1 Dans les carrefours à feux, il existe une situation où le nombre d'entrées est égal au nombre de sorties. Cette situation est appelée **état nominal** ou **état d'équilibre** et est notée x_N . La commande correspondante quant à elle est notée u_N . Dans cette situation, nous avons $Bu_N + L = 0$. ▶

1.4.1.2 Cas d'une intersection à trois phases

La complexité d'une intersection à 3-phases peut être ramenée à l'évolution des mouvements tourne-à-gauche. Ces mouvements peuvent créer des conflits avec d'autres mouvements considérés comme "élémentaires". Comme nous pouvons le constater d'après la figure 1.15, il existe trois types de courants de véhicules : directs, tourne-à-droite (phases 1 et 3) et tourne-à-gauche (phase 2).

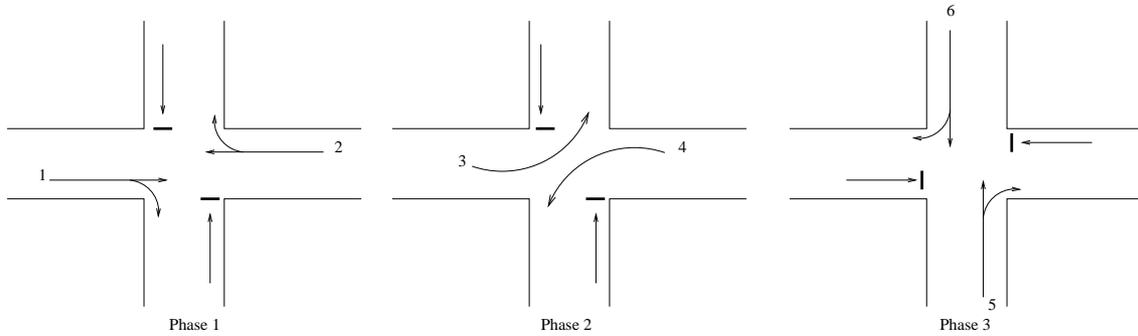


FIGURE 1.15 – L'écoulement du trafic dans un carrefour à trois phases.

Comme dans la section précédente, soit $x_i(k)$ la longueur de la file d'attente dans chaque phase à la fin du cycle k . L'application du principe de conservation des flux du trafic nous donne immédiatement le modèle suivant :

$$x_1(k+1) = x_1(k) + q_1c - s_1g_{e_1}(k)$$

$$x_2(k+1) = x_2(k) + q_2c - s_2g_{e_2}(k)$$

$$x_3(k+1) = x_3(k) + q_3c - s_3g_{e_3}(k)$$

comme $c = g_{e_1}(k) + g_{e_2}(k) + g_{e_3}(k)$, il en résulte :

$$x_1(k+1) = x_1(k) + q_1c - s_1g_{e_1}(k)$$

$$x_2(k+1) = x_2(k) + q_2c - s_2g_{e_2}(k)$$

$$x_3(k+1) = x_3(k) + (q_3 - s_3)c + s_3g_{e_1}(k) + s_3g_{e_2}(k)$$

soit sous forme matricielle :

$$x(k+1) = x(k) + Bu(k) + L \tag{1.8}$$

où $x(k)$ est le vecteur d'état, $u(k) = (g_{e_1}(k), g_{e_2}(k))^T$ est le vecteur de commande, et

$$B = \begin{pmatrix} -s_1 & 0 \\ 0 & -s_2 \\ s_3 & s_3 \end{pmatrix}; L = \begin{pmatrix} q_1 c \\ q_2 c \\ (q_3 - s_3)c \end{pmatrix}$$

D'une manière générale, nous nous intéressons au modèle d'état suivant :

$$x(k+1) = x(k) + Bu(k) + L \quad (1.9)$$

où $x \in \mathbb{R}_+^n$, $u \in \mathbb{R}_+^m$, $B \in \mathbb{R}^{n \times m}$ et $m < n$.

1.4.2 Cas où le cycle des feux est variable

Pour le cas d'un carrefour à deux phases, le système considéré est celui établi dans la section 1.4.1.1 (voir les équations (1.5) et (1.6)) et dans lequel les équations de l'évolution des files d'attente sont données par :

$$\begin{aligned} x_1(k+1) &= x_1(k) + (q_1 - s_1)c(k) + s_1 g_{e_2}(k) \\ x_2(k+1) &= x_2(k) + q_2 c(k) - s_2 g_{e_2}(k) \end{aligned}$$

ou d'une manière équivalente :

$$x(k+1) = x(k) + Bu(k) \quad (1.10)$$

avec cette fois-ci, $u(k) = (c(k), g_{e_2}(k))^T$ est le vecteur de commande et :

$$B = \begin{pmatrix} q_1 - s_1 & s_1 \\ q_2 & -s_2 \end{pmatrix}$$

De la même manière, nous obtenons le modèle d'état du carrefour à trois phases.

$$x(k+1) = x(k) + Bu(k) \quad (1.11)$$

où $u(k) = (c(k), g_{e_1}(k), g_{e_2}(k))^T$ est le vecteur de commande, et

$$B = \begin{pmatrix} q_1 & -s_1 & 0 \\ q_2 & 0 & -s_2 \\ q_3 - s_3 & s_3 & s_3 \end{pmatrix}$$

D'une manière générale, dans le cas où le cycle est variable, nous nous intéressons au modèle d'état suivant :

$$x(k+1) = x(k) + Bu(k) \quad (1.12)$$

où $x \in \mathbb{R}_+^n$, $u \in \mathbb{R}_+^n$ et $B \in \mathbb{R}^{n \times n}$.

1.4.3 Domaine de commande admissible

Comme pour toute stratégie de synthèse, il est nécessaire de définir clairement le domaine de commande admissible. En effet, la commande du système est à rechercher dans un ensemble défini par deux limitations pratiques :

- La première est la nécessité de prendre en compte des contraintes sur l'état. En effet, pour que le modèle de transport ait un sens physique, il faut que l'état appartienne au domaine $\mathbb{E} = \{x \in \mathbb{R}^n / x \geq 0\}$. Ainsi la commande est à rechercher parmi les suites de commandes telles que $x \in \mathbb{E}$ pour tout $k \in \mathbb{N}$.
- La deuxième limitation est la nécessité de prendre en compte des contraintes sur la commande. En effet, les feux verts effectifs doivent respecter les conditions aux limites suivantes $0 \leq g_{e_i} \leq c$. Cependant, certaines conditions aux limites ne correspondent pas à la réalité physique du système. En particulier, $g_{e_1} = 0$ ou $g_{e_1} = c$ se traduisent par l'absence des feux verts effectifs pour la phase 1 ou la phase 2 respectivement. Par conséquent, il est important de sélectionner deux bornes g_{min} et g_{max} sur la valeur de la commande telles que $g_{min} \leq g_{e_i} \leq g_{max}$. Ces bornes peuvent être fournies par les actionneurs et doivent être convenablement choisies. En effet, une durée courte ou longue du feu vert effectif n'est pas acceptable sur le plan pratique. De ce fait, les feux verts effectifs doivent respecter les conditions aux limites suivantes :

$$u_{min} = g_{min} \leq g_{e_i} \leq g_{max} = u_{max} \quad (1.13)$$

Notons que ces deux contraintes traduisent une cohérence avec le fonctionnement réel d'un carrefour. En effet, puisque le cycle de signalisation est borné, le temps effectif du feu vert doit être borné. En outre, puisque la file d'attente traduit dans notre cas le nombre de véhicules présents dans un couloir, nous comprenons aisément qu'elle soit positive.

1.5 Modes de régulation des carrefours

La régulation des carrefours par les feux de signalisation peut fonctionner sous divers modes. Outre le contrôle manuel, on distingue généralement le contrôle automatique, le contrôle adaptatif et le contrôle coordonné. Ces modes ne sont pas toujours distincts et peuvent se combiner à plusieurs titres, voir se substituer l'un à l'autre.

Mode manuel : dans ce cas, c'est un individu qui actionne le changement d'état du carrefour. C'était le cas des premiers feux de circulation, mais aujourd'hui encore, les

feux peuvent fonctionner en mode manuel si la situation l'exige, ou également en mode semi-manuel (ou semi-automatique).

Mode automatique : ce mode n'exige aucune intervention extérieure ; les états successifs d'un cycle se déroulent séquentiellement suivant les plans de feux en place dans le contrôleur. On parle aussi de fonctionnement en cycles fixes.

Mode adaptatif : contrairement au mode automatique, ce mode permet l'adaptation de la durée des intervalles des feux verts, et par conséquent la modulation de l'ensemble des phases d'un cycle en fonction des variations du trafic. Ainsi, un feu vert peut être allongé pour écouler le flux d'une voie où s'écoule un trafic plus important que dans les voies adjacentes. Un autre exemple d'adaptation est la priorité donnée aux véhicules de transport public : leur arrivée dans un carrefour déclenche soit l'allongement de l'état du feu vert, soit la réduction de l'intervalle du feu rouge sur la voie qu'ils empruntent.

Mode coordonné : ce mode signifie que plusieurs contrôleurs, donc plusieurs carrefours, sont soumis à une même stratégie de régulation. Il existe deux grandes applications du mode coordonné. La première est la régulation d'axes de circulation. Elle consiste à coordonner les phases des différents carrefours d'un même axe routier afin de rendre plus confortable et plus fluide la progression des véhicules, c'est-à-dire, d'éliminer leurs arrêts aux carrefours (du moins s'ils respectent une vitesse moyenne, définie par le gestionnaire des feux). La seconde application de la coordination est la régulation des zones. Dans ce cas, ce sont les feux de l'ensemble d'un périmètre géographique déterminé (un quartier urbain par exemple) qui sont coordonnés entre eux, afin d'optimiser les déplacements des usagers, c'est-à-dire, de minimiser le temps qu'ils passent dans cette zone.

Les deux derniers modes qu'on vient d'évoquer soulèvent chacun des complexités techniques que ne posent pas les deux premiers. Ces modes (adaptatif, coordonné) exigent de disposer de moyens de détection des variations du trafic, c'est-à-dire de recueillir des informations caractérisant la circulation en temps réel. D'où la nécessité d'utiliser des capteurs automatiques. Il en existe différentes sortes : pneumatiques, hydrauliques, acoustiques, à boucles électromagnétiques, à ultrasons, à infrarouges, les radars, les caméras, etc... Ces capteurs sont placés à une certaine distance du carrefour concerné reliés par câble ou par ondes au contrôleur qui reçoit et interprète l'information codée par le détecteur (un élément électronique qui transforme l'information issue du capteur en signal électrique).

Notons également que les modes adaptatifs et coordonnés, ainsi que leur combinaison, se trouvent aujourd'hui gérés à distance à partir d'un poste de contrôle. Cette télé-gestion repose évidemment sur des transmissions d'informations entre les feux et le poste de contrôle. Les modes manuel et automatique entrent dans le cadre des systèmes de commande en **boucle ouverte**. Ce sont des systèmes qui se basent sur des bases de données prédéfinies. Dans ces conditions, le système marche sans prendre en considération les problèmes brusques : la forte congestion, les incidents, etc. L'utilisation de ces systèmes est désormais impossible pour répondre à la demande actuelle. Donc il est nécessaire d'adopter des approches qui nous permettent d'en savoir plus sur l'état en temps réel et qui agissent directement pour fluidifier le trafic. D'où l'utilisation actuelle dans les recherches récentes des systèmes en **boucle fermée** dont nous pouvons inclure les modes adaptatifs et coordonnés.

1.6 Classification des différentes stratégies de régulation du trafic dans un carrefour

La régulation du trafic dans un carrefour à feux concerne en général deux objectifs distincts : la fluidification ou la résorption de congestion. Dans le premier cas, on évite de se retrouver dans une situation de trafic très dense en essayant d'ajuster les durées de commutation des feux en fonction de la demande d'affluence au carrefour : c'est une action *a priori*. Dans le second cas, on est confronté à un trafic saturé (état de congestion). Dans ce cas, il faudra agir *a posteriori*. Notons qu'une stratégie de régulation au niveau d'un seul carrefour est appelée *stratégie locale*. Elle permet la régulation des feux d'un carrefour indépendamment de la stratégie adoptée pour les feux des carrefours avoisinants. Dans le cas d'un carrefour congestionné sur une zone étendue englobant ainsi plusieurs carrefours, une stratégie locale ne saurait résorber cette forte densité de trafic. En effet, pour répondre au phénomène d'accumulation de véhicules, une stratégie globale permettant la régulation simultanée de plusieurs carrefours interdépendants pourrait atténuer ce phénomène.

La littérature concernant la régulation du trafic dans un carrefour à feux est abondante. Bien qu'une quelconque classification soit toujours partiellement arbitraire, il nous semble qu'une classification conduisant à séparer les travaux existants en deux catégories principale et raisonnable.

Une première catégorie de travaux englobe une grande partie de la littérature sur la régulation des carrefours. Elle est celle communément désignée sous le nom de **régulation**

algorithmique. Le souci qui a accompagné son développement était d'intégrer l'évolution technique dans les infrastructures existantes. Ceci a conduit, comme il est présenté dans la suite du chapitre, à l'adoption de stratégies plus ou moins performantes mais ne permettant pas d'obtenir des résultats théoriques. Cependant, l'absence de résultats théoriques n'a pas empêché une utilisation intensive de ces stratégies pour résoudre les problèmes du trafic.

Une deuxième partie de la littérature sur la régulation des carrefours tente de fournir des stratégies de régulation en passant par des représentations mathématiques. Ces représentations font appel, à des stades plus ou moins avancés de leur élaboration, à la notion de modèle d'état. Cette catégorie est celle communément désignée sous le nom de **régulation analytique.** Elle se conforme le plus aux objectifs fixés pour notre investigation, à savoir, commande en temps réel pour la régulation des carrefours, portant directement sur le modèle du système. Ces commandes mettent en jeu des grandeurs et des caractéristiques concrètes qui se prêtent, autant que possible, à une interprétation physique pour le modèle du carrefour.

Dans la suite de ce chapitre, les deux catégories de travaux mentionnées plus haut sont présentées. La première catégorie, ne s'inscrivant pas dans le contexte de notre étude, est présentée d'une manière non exhaustive. La deuxième catégorie est présentée d'une façon plus ou moins détaillée afin de dégager aussi clairement que possible notre propre contribution.

1.6.1 Stratégies de régulation algorithmiques

Dans la catégorie de stratégie de régulation algorithmique, nous distinguons trois grandes stratégies de régulation.

1.6.1.1 Stratégie fixe ou pré-déterminée

Dans ce mode de régulation, tous les paramètres suivants sont déterminés et réglés à partir de données recueillies avant l'installation : la durée de cycle, les phases, les durées de vert/rouge. Les commutations des feux de signalisation sont constantes d'un cycle à un autre. Ainsi, on retrouve à chaque cycle la même longueur et la même durée des phases. Nous trouvons dans ce mode de régulation les systèmes à base de plans de feux fixes.

Le rôle principal des plans de feux fixes est de mettre en marche cycliquement une série de phases pendant des durées données. Cette série de phases doit permettre de mieux répondre à la demande moyenne de la capacité estimée du carrefour. Pour calculer le plan de feux (la charge d'un carrefour, la durée de cycle, la répartition des temps de vert entre

phases...), il est nécessaire de définir la notion de phasage. Cette notion est définie comme étant l'ensemble des phases utilisées pour la commande de l'intersection. Une phase est définie comme un ensemble de feux qui ne sont pas antagonistes entre eux, c'est-à-dire les feux d'une phase qui autorisent des mouvements compatibles entre eux.

D'un point de vue technique, ces plans de feux fixes sont des machines à états qui spécifient les temps des feux vert et de rouge sur chaque ligne d'un carrefour, ainsi que leur séquençement. Ces plans fixes sont préalablement définis en fonction des conditions du trafic typiques de certaines heures ou certains jours. Il existe déjà un certain nombre de logiciels qui permettent de préparer ces plans. L'un des plus connus et utilisés est le logiciel **TRANSYT** [13] dont il existe de nombreuses versions. Ce logiciel fournit un plan optimal du point de vue des durées de feu vert et de leurs décalages. TRANSYT minimise un indicateur s'appuyant entre autres sur le temps du trajet global et sur le nombre d'arrêts. Il est à noter que les plans de feux fixes ne sont pas fixés pour toutes les périodes d'une journée ou pour tous les jours. Un choix doit être effectué entre les différents plans disponibles au Poste de Commande selon la situation du trafic. Un opérateur peut réaliser ce choix manuellement ; c'est généralement le cas lorsqu'il détecte un événement exceptionnel nécessitant une gestion adaptée du réseau.

Il va sans dire que ce choix peut également être conduit automatiquement à partir de la planification horaire. Par exemple, un plan pour les heures de pointe, un pour les heures fluides ou pour les week-ends, etc. Cette méthode montre rapidement ses limites quand les conditions du trafic changent rapidement. Il devient alors nécessaire de déployer des méthodes comme la méthode à seuils ou la méthode des vecteurs.

La méthode à seuils consiste en une structure arborescente où la position de la valeur du débit détecté par rapport à un ou plusieurs seuils entraîne un aiguillage vers un type de décision ou un autre. Leur mise en œuvre, et particulièrement la détermination des seuils, est délicate.

La méthode des vecteurs a pour principe la comparaison, à intervalles réguliers, d'un vecteur représentatif de la situation courante à des vecteurs types. Ces vecteurs sont associés à des situations de trafic de référence pour lesquelles on dispose d'un plan de feux. Ces plans de feux sont classiquement calculés à la main ou à l'aide d'outils spécialisés comme par exemple **TRANSYT-7F** [14], [15]. Le plan de feux choisi est celui dont le vecteur de référence minimise sa distance avec le vecteur identifiant la situation courante. En France, la méthode des vecteurs est celle la plus utilisée dans les gros réseaux urbains.

1.6.1.2 Stratégie semi-adaptative

Dans ce mode de régulation, la durée de cycle ainsi que les durées du feu vert peuvent varier d'un cycle à un autre en fonction de la demande. Cette seconde méthode entre dans le cadre de la micro-régulation en se basant essentiellement sur les plans de feux fixes. Elle ajuste les durées de commutation d'un cycle à un autre en fonction de la demande enregistrée en amont.

L'action menée par ce type de méthode de régulation est obtenue à partir de l'analyse instantanée de la demande des véhicules pour prendre en compte des variations exceptionnelles ou aléatoires du trafic. Cette analyse est rendue possible grâce aux différents capteurs mis en place : des boucles noyées dans la chaussée, radar etc. La mise en service d'une micro-régulation à un carrefour à feux répond le plus souvent aux objectifs suivants : rendre crédible la signalisation tricolore, assurer le confort de l'utilisateur en minimisant son temps d'attente, diminuer le nombre d'arrêts de véhicules et surtout minimiser la longueur des files d'attente. Quelques systèmes ont été développés dans ce sens. Nous présentons dans ce qui suit les systèmes de type semi-adaptatifs les plus utilisés dans de grandes villes mondiales :

SURF2000 : Système Urbain de Régulation des Feux : Pour la régulation du trafic du réseau parisien, la ville de Paris utilise le système centralisé SURF2000. Les actions de SURF2000 se partagent en actions globales pour optimiser le réseau, appelées *macro-régulation*, et en actions locales au niveau des carrefours, appelées *micro-régulation*. Les stratégies de *micro-régulation* employées sur le réseau parisien consistent à limiter le temps d'attente des transports en commun détectés par des boucles magnétiques placées au niveau des arrêts de bus, et à activer les feux sur appel. Par exemple, dans les rues où il y a peu de circulation, le feu ne passe au vert que si un véhicule est détecté.

SCOOT : Split Cycle and Offset Optimisation Technique : Ce système a été développé en 1981 par le *Traffic and Road Research Laboratory TRRL* en Grande Bretagne. Ce système se base initialement sur un plan de feux fixes issu du système TRANSYT-7F avec un algorithme d'optimisation fourni pour l'application en ligne ([16], [17], [18], [19]). Il opère sur des petites variations des durées du feu vert et de la durée de cycle et ceci en fonction de l'état du trafic. Le réseau est représenté sous la forme d'un graphe où les sommets sont des intersections. Les arcs reliant deux sommets représentent un courant de trafic unidirectionnel entre deux intersections.

SCATS : Sydney Coordinated Adaptive Traffic system : Ce système a été développé dans les années quatre vingts [20]. Il est mis en place dans plusieurs villes en Australie. Le rôle essentiel de ce système est la reconstitution en temps réel des plans de feux fixes disponibles dans une base de données. Il se base sur l'optimisation de trois paramètres de trafic [21] : les durées de feu vert, les décalages et la longueur des cycles. Il compose, à partir d'une bibliothèque de durées de feu vert et les décalages des cycles, un plan de feux fixes adéquat à la situation du trafic observé à un instant donné.

1.6.1.3 Stratégie adaptative ou temps réel

La recherche actuelle s'est orientée vers la réalisation de systèmes de régulation qui tiennent compte de l'évolution du trafic au cours du temps et qui réagissent en temps réel. Selon [22], un système de contrôle en temps réel du trafic doit répondre aux exigences suivantes :

- la stratégie devrait répondre réellement à la demande actuelle et non pas à une demande prédite ou historique.
- la stratégie ne devrait pas être réduite à intervenir à des périodes de contrôle fixées arbitrairement.

C'est dans cet esprit que plusieurs types de système ont été élaborés pour ce mode de régulation. Les plus connus sont :

UTOPIA : Urbain Traffic Optimization by Integrated automation : Ce système a été développé et testé en Italie [23]. Dans ce système, la régulation s'effectue sur deux niveaux :

- le niveau local : chaque contrôleur agit selon son propre modèle d'écoulement du trafic, en utilisant au maximum l'information provenant du niveau supérieur pour améliorer la stratégie au niveau d'une intersection.
- le niveau global : le contrôleur régional a un rôle de supervision et assure l'optimisation globale du réseau. Il transmet aux différents contrôleurs locaux des règles à appliquer sous forme de plans de feux fixes de référence.

UTOPIA est un système hiérarchique décentralisé. Ses principales caractéristiques sont, d'une part la priorité absolue donnée à certains véhicules publics et d'autre part l'optimisation du trafic des véhicules en toute situation.

PRODYN : PROgrammation DYNamique : C'est un système développé en France [24]. Il est essentiellement fondé sur la programmation dynamique. Pour réguler un réseau de carrefours, l'algorithme PRODYN décompose le problème en plusieurs sous-

problèmes. Il utilise des techniques de décomposition pour convertir le problème d'optimisation initial de grand taille, en plusieurs sous-problèmes de taille plus petite [25]. Chacun de ces sous-problèmes est résolu par la méthode de programmation dynamique. Par la suite, le problème global est résolu en utilisant une procédure itérative à deux niveaux (local concernant une intersection isolée et globale pour un réseau).

OPAC : Optimization Policies for Adaptive Control : OPAC a été développé par [26]. Il s'agit d'une stratégie de commande des feux en temps réel adaptée à la demande, avec le caractère d'optimisation par la programmation dynamique (DP). OPAC utilise un niveau de contrôle à l'intersection locale pour la minimisation du délai total de l'intersection et le niveau du réseau pour la synchronisation. Le type de contrôle et les niveaux d'influence locale et globale sont flexibles [27] ; [28] ; [29].

OPAC base les signalisations de signal local sur des données détectées de toutes les directions pendant une période principale (typiquement 15 secondes) et des données prévues pendant une nouvelle période plus longue (typiquement 60 secondes). Bien qu'OPAC fonctionne acycliquement dans le niveau local, il utilise le cycle virtuel pour maintenir la synchronisation de réseau au niveau de réseau. La longueur du cycle virtuel varie selon les besoins de l'intersection critique ou de la majorité des intersections. Cela permet typiquement au cycle virtuel de varier d'une seconde par cycle. Dans cette limitation, OPAC fournit la coordination locale en considérant des flux dans et hors d'une intersection en sélectionnant son "offset" et ses longueurs de phase.

Le processus de contrôle de sur-saturation dans OPAC tente de maximiser la sortie en choisissant l'intersection avec la demande de trafic maximale. Il est fait en considérant des flux de saturation et l'espace disponible pour stocker des véhicules sur chaque liaison. Le contrôle de congestion implique la détermination de la phase suivante donnée quand il n'y a aucune liaison critique qui est sur le point ou entrain de se dissiper.

Il faut noter que OPAC souffre d'une exigence informatique énorme. La mise en oeuvre en temps réel de la stratégie OPAC est toujours limitée aux intersections isolées en raison de la complexité exponentiellement accrue du calcul de réseau.

CRONOS : ConROI of Networks by Optimization of Switchovers : Ce système adopte la même stratégie que PRODYN. Il a été développé par l'INRETS en France dans les années 1990 [30]. Il peut être exploité selon deux modes distincts : mode centralisé dans lequel un ensemble de carrefours est régulé simultanément ou le mode décentralisé dans lequel chaque carrefour est régulé séparément. La méthode d'optimisation est basée sur un algorithme fondé sur le principe des méthodes de voisinage [31].

1.6.1.4 Conclusion sur les méthodes algorithmiques

Dans cette section, nous avons présenté trois types de stratégies utilisant les techniques algorithmiques pour la régulation des carrefours à feux. Pour ces stratégies, nous avons relevé les inconvénients suivants :

- les contrôleurs de régulation basés sur les plans de feux fixes atteignent très vite leurs limites. En effet, il existe un grand écart entre la demande estimée (qui sert dans le calcul des plans) et la demande réelle (mesurée sur le terrain).
- ces stratégies ne s’adaptent pas de manière efficace aux variations permanentes du trafic. Le contrôle d’un tel type de trafic nécessite un degré important d’adaptabilité pour répondre, dans de très courts délais, à une demande très fluctuante et aléatoire. Des temps de réponse plus courts conduiront à une régulation de trafic plus efficace et plus adaptée à la demande. Il n’en demeure pas moins qu’il existe des degrés différents d’adaptabilité en fonction de la taille de l’intersection ou du réseau.
- en période de congestion, toutes ces stratégies sortent de leur domaine de validité. Des expériences multiples visant à comparer, en situation saturée, l’efficacité de ces méthodes révèlent de faibles écarts entre ces approches [32]. Ce constat milite donc pour une résolution des problèmes de congestion, faisant d’avantage appel à des méthodes nouvelles.

En conclusion, nous pouvons dire que, l’inadéquation des plans à moyen terme, suite à l’évolution du trafic et l’adaptation insuffisamment souple en cas de trafic perturbé, rendent l’application de ce type de régulation très restreint sur des cas ordinaires.

1.6.2 Stratégies de régulation analytiques

Alors que la première catégorie des systèmes de régulation se base sur des méthodes algorithmiques, la deuxième catégorie se caractérise par un fond théorique. En effet, les stratégies de régulation algorithmique sont fondées sur des bases de données que l’on peut considérer comme des informations qui alimentent le système central. La deuxième catégorie ne se contente pas de cette base de donnée. Au contraire, elle commence tout d’abord par une modélisation de l’environnement qui, d’une part traduit l’évolution du système et d’autre part met en valeur toutes les contraintes qui l’entoure. Ensuite, elle utilise tout l’arsenal mathématique pour identifier, modéliser, optimiser et enfin réguler les flux de véhicules au sein de l’intersection. Notre travail s’inspire des résultats obtenus dans le cadre de cette catégorie et propose une nouvelle approche pour la régulation des intersections signalisées.

Soulignons tout d'abord que l'objectif de cette présentation n'est pas de décrire d'une façon détaillée les travaux en question. Il est de répondre à deux soucis : le premier étant de rendre compte aussi fidèlement que possible des travaux portant étroitement sur le même sujet qui ont précédé notre réflexion, et qui ont été, par conséquent d'une inspiration certaine. Le deuxième objectif de cette présentation est qu'elle soit suffisamment détaillée pour que notre propre contribution puisse être clairement dégagée.

Les travaux adoptant une approche du type analytique pour réguler les carrefours à feux congestionnés ne sont pas nombreux. Le travail de Chang et Lin [33] est celui qui nous intéresse le plus. En effet, la régulation d'une intersection congestionnée est étudiée dans le cadre d'une modélisation discrète. C'est la raison pour laquelle ce travail est relativement bien détaillé dans la présentation qui suit afin de pouvoir situer clairement notre apport relatif.

Le travail de Schutter [34] concerne la modélisation continue. Ceci conduit à une formulation légèrement différente dans laquelle l'élaboration de la commande ainsi que son implémentation pour la régulation des carrefours à feux nécessitent non seulement des hypothèses relativement sévères, mais aussi des méthodes difficiles à appliquer. Les particularités associées au caractère continu du système considéré dans [34] rendent la comparaison avec notre travail peu significative. Ceci justifie que dans ce qui suit, ce travail soit présenté d'une manière relativement succincte.

1.6.2.1 Travaux de Chang et Lin

Chang et Lin [33] ont développé un modèle qui rentre dans le contexte des modèles déterministes en se basant sur une modélisation de flux macroscopique. L'objectif est de minimiser le retard des véhicules pendant la période de congestion dans une intersection élémentaire à deux phases. Les résultats de l'étude ont abouti à une stratégie de commande synchronisée pour commander les feux de signalisation. Pour définir proprement ce travail, nous avons besoin des notations suivantes :

q_j : Le taux des arrivées de chaque ligne $j \in \{1, \dots, 4\}$

g_i : Le temps effectif du feu vert de la phase $i \in \{1, 2\}$

s_j : Le taux de saturation de chaque ligne $j \in \{1, \dots, 4\}$

c : Le cycle de signalisation

D_j : Le retard des véhicules de chaque ligne $j \in \{1, \dots, 4\}$

Formulation du problème

Le modèle proposé dans [33] gère l'évolution du retard causé par la congestion dans une intersection élémentaire à deux phases (voir figure 1.3). Plus précisément, l'équation

d'état du système global est donnée par :

$$D(k+1) = D(k) + Bu(k) + W(k) \quad (1.14)$$

où

$$D(k) = D_1(k) + D_2(k) + D_3(k) + D_4(k)$$

$$u(k) = g_2^2(k)$$

$$B = \frac{1}{2}(s_1 + s_2 + s_3 + s_4)$$

$$W(k) = W_1(k) + W_2(k) + W_3(k) + W_4(k)$$

$D(k)$ représente la somme des retards des quatre lignes, $u(k)$ définit la variable de commande et $W(k)$ est considérée comme la somme des perturbations $W_i(k)$ de chaque ligne qui sont complètement connues et dont les évolutions sont données par :

$$W_i(k+1) = \frac{1}{2}[q_i c^2 - s_i g_2^2(k) - 2s_i c^2 + 2s_i g_2(k)c + q_i c^2] \quad (1.15)$$

Remarque 1.2 Signalons tout de suite que la perturbation $W(k)$ est une fonction non-linéaire par rapport à la commande et les paramètres du système. Cette perturbation est par conséquent susceptible de changer sensiblement de valeur, pour des variations infinitésimales des paramètres définissant le modèle du système. Or, les perturbations conditionnent aussi bien le système que le calcul de la commande optimale. Ce conditionnement peut alors se manifester par le fait que, pour un système, les commandes à appliquer sont acceptables tandis que pour un autre système, pourtant infiniment voisin, elles ne le sont pas. Ainsi, considérer $W(k)$ comme une perturbation présente un inconvénient certain. ▶

Le critère d'optimisation

Le critère d'optimisation précise les objectifs par rapport auxquels l'optimalité est définie. Or, comme l'objectif principal de cette étude est de minimiser le retard des véhicules pendant la durée de la congestion, on conçoit facilement que le critère soit défini par la relation suivante :

$$J = \frac{1}{2}D^2(N) + \frac{1}{2} \sum_{k=2}^{N-1} D^2(k). \quad (1.16)$$

Il va sans dire que le critère (1.16) permet de prédire le retard total qu'aurait les véhicules par l'application de la suite de commandes $u(k)$ sur toute la période de congestion. C'est la raison pour laquelle, dans la suite de l'exposé, N est appelé **horizon de commande**.

domaine d'optimisation

L'objectif de la commande est de minimiser le critère d'optimisation (1.16). Cette optimisation est à rechercher dans un ensemble défini par deux limitations :

- La première est la nécessité de prendre en compte des contraintes sur l'état. En particulier, pour que le modèle du retard ait un sens physique, il faut que l'état appartienne au domaine $\Omega = \{D \in \mathbb{R}/D > 0\}$.
- La deuxième limitation est la nécessité de prendre en compte des contraintes sur la commande. En effet, les feux verts doivent respecter les conditions aux limites suivantes $g_{min} \leq g(k) \leq g_{max}$.

Ainsi la commande est à rechercher parmi les suites de commandes telle que :

$$D \in \Omega, \quad g_{min} \leq g(k) \leq g_{max}; \forall k \in \mathbb{N}. \quad (1.17)$$

D'autre part, une contrainte finale sur l'état est imposée, à savoir :

$$D(N) = \zeta. \quad (1.18)$$

Cette contrainte implique que la séquence de commande $u(k)$ appliquée à partir de l'instant initial où le système est à l'état D_0 doit conduire le système à la valeur ζ après N cycles.

La stratégie de commande

Une fois le critère et le domaine d'optimisation donnés, le problème de la commande optimale consiste à choisir, connaissant l'état initial D_0 , une suite de commandes admissibles $u(0), \dots, u(N-1)$ pour les contraintes (1.14)-(1.17)-(1.18) telles que le critère (1.16) prenne la valeur minimale. Il s'agit d'un problème d'optimisation linéaire quadratique à extrémité gauche libre et à extrémité droite fixe. Autrement dit, l'état $D(N)$ est supposé défini, tandis que l'état $D(0)$ n'est aucunement réglementé.

Le problème ainsi posé est tout à fait classique et peut être résolu par les méthodes indirectes qui consistent à résoudre les conditions d'optimalité en utilisant les algorithmes sur le calcul variationnel et le principe de maximum, Ces méthodes permettent d'écrire les équations que doit satisfaire la commande pour optimiser le critère J . En effet, définissons l'Hamiltonien :

$$H = \frac{1}{2}D^2(N) + \frac{1}{2}D^2(k) + \lambda(k+1)[D(k) + Bu(k) + W(k)] \quad (1.19)$$

où $\lambda(k)$ est le multiplicateur de Lagrange. Les équations d'optimisation avec conditions initiales libres et finales fixes s'écrivent alors :

$$\frac{\partial 1/2D^2(N)}{\partial D(N)} = \lambda(N) \Rightarrow D(N) = \lambda(N), \quad (1.20)$$

$$\frac{\partial H}{\partial D(k)} = \lambda(k) \Rightarrow D(k) + \lambda(k+1) = \lambda(k), \quad (1.21)$$

$$\frac{\partial H}{\partial \lambda(k+1)} = D(k) + Bu(k) + W(k), \quad (1.22)$$

$$(1.23)$$

Comme il a été supposé par Chang [33], si on considère que l'Hamiltonien H , donné par l'équation (1.19), est une fonction linéaire par rapport à u , ceci conduit à la commande optimale suivante :

$$u(k) = g_2^2(k) = \begin{cases} g_{2min}^2 & \text{si } \lambda(k+1)B > 0; \\ g_{2max}^2 & \text{si } \lambda(k+1)B < 0. \end{cases}$$

La commande optimale ainsi définie est communément désignée sous le nom de **commande bang-bang**. Ainsi, l'idée de la régulation de l'intersection devient alors plus claire. g_{min} et g_{max} agissent alternativement sur la phase dont les taux d'arrivées sont les plus grands. Par exemple, au début de la procédure $g_2 = g_{2max}$ est affilié aux lignes 3 et 4 de la phase II dont on sait que leurs taux d'arrivées sont les plus grands. Le temps du feu vert de la phase I est alors calculé une fois que le temps du cycle est déterminé, i.e., $g_1 = c - g_{2max}$. Une fois g_1 terminé, le deuxième cycle commence et g_{2min} remplace g_{2max} et ainsi de suite. La procédure se termine une fois $x_1 = x_2 = 0$ ou $x_3 = x_4 = 0$

Commentaires critiques

Dans ce travail, nous avons relevé les inconvénients suivants :

- le commentaire critique de la remarque 1.2 est toujours valable ;
- l'hypothèse de considérer le terme $W(k)$ donné par l'équation (1.15) comme une perturbation peut limiter l'intérêt pratique de la commande bang-bang. Cette limitation se manifeste par trois points :
 - la commande bang-bang serait une commande sous-optimale (encore faut-il le démontrer). En effet, l'absence explicite de cette hypothèse rendrait le problème d'optimisation non-linéaire. Ainsi, une commande bang-bang ne doit pas être espérée dans ce cas ;

- la non prise en compte des perturbations instantanées dans chaque cycle (le changement brusque de flux causé par $W(k)$), ce qui implique l'attente d'une durée considérable pour basculer les feux.
- dans le cas où $\lambda(k+1)B = 0$, l'Hamiltonien H est indépendant de la commande et par conséquent, il ne peut être minimisé par rapport à u . Ce problème de singularité peut être évité si la stratégie de la commande assure qu'à chaque instant $\lambda(k+1) \neq 0$. Or, si la perturbation ramène l'état du système proche de la singularité, la stratégie doit commuter vers une loi de commande qui éloigne l'état de la région de singularité. Cet objectif ne peut être assuré par la commande bang-bang, ce qui accentue davantage l'inconvénient souligné au point précédent.

1.6.2.2 Les travaux de De Schutter et De Moor

Le travail réalisé par De Schutter et De Moor [34], nous emmène vers un autre cadre d'étude. Cette étude est basée essentiellement sur la recherche d'une séquence de commutations optimale des feux de signalisation dans une intersection élémentaire. Une fois que le taux des arrivées et le taux maximal des départs des véhicules dans cette intersection sont connus, nous pouvons calculer les commutations des feux de signalisation qui minimisent plusieurs critères.

Anticipant sur la présentation qui va suivre, on peut dire que la méthode de De Schutter et De Moor, appliquée aux carrefours à feux, tire profit des propriétés bien établies d'une méthode bien connue dans la littérature anglophone par ELCP (**Extended Linear Complementarity Problem**). Pour ce faire, le comportement du système est dynamiquement identifié par une équation différentielle linéaire hybride. La modélisation se fait d'une manière itérative suivant que le feu est rouge ou vert. La commande optimale des feux de signalisation est calculée alors en résolvant le problème ELCP et en choisissant parmi les solutions, celle qui optimise un critère donné.

Modélisation du système

De Schutter et De Moor ont modélisé l'évolution des files d'attente dans une intersection à feux élémentaire (voir figure 1.3). Cette intersection est considérée comme étant un système à quatre lignes L_1, L_2, L_3 et L_4 . Chaque ligne $L_{i=1,2,3,4}$ représente un couloir de flux de véhicules. A la fin de chaque couloir, nous attribuons des périodes de feux de signalisation $T_{i=1,2,3,4}$.

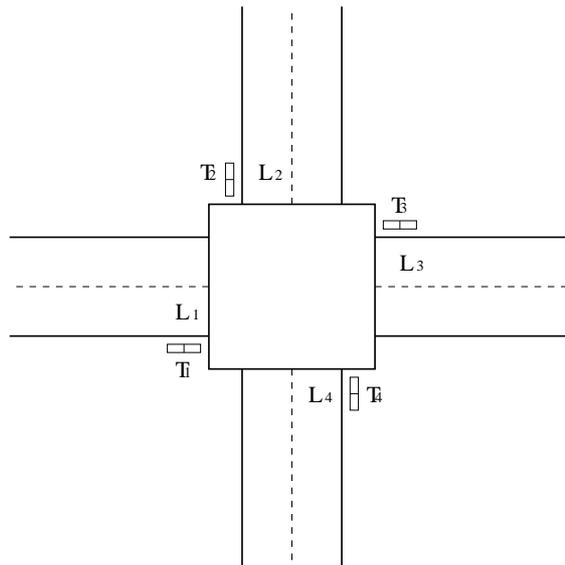


FIGURE 1.16 – Intersection contrôlée par des feux de signalisation

Pour décrire les équations décrivant la relation entre les instants de commutation et les files d’attente, un certain ensemble de notation est nécessaire :

- $l_i(t)$: la longueur de la file d’attente dans la ligne L_i à l’instant t , $i = 1, \dots, 4$,
- λ_i : le taux moyen des arrivées des véhicules dans la ligne L_i , $i = 1, \dots, 4$,
- μ_i : le taux moyen des départs des véhicules dans la ligne L_i , $i = 1, \dots, 4$,
- t_0, t_1, t_2, \dots : les instants de commutations des feux de signalisation. Ces instants de commutations sont définis par le tableau 1.1,
- $\delta_k = t_{k+1} - t_k$: la durée entre deux instants de commutation consécutifs.

Durée	T_1	T_2	T_3	T_4
$t_1 - t_0$	rouge	vert	rouge	vert
$t_2 - t_1$	vert	rouge	vert	rouge
$t_3 - t_2$	rouge	vert	rouge	vert
\vdots	\vdots	\vdots	\vdots	\vdots

TABLE 1.1 – Les commutations des feux de signalisation.

Considérons maintenant la ligne L_1 . La description de la variation de l_1 part de l’idée que, pour chaque cycle :

- **si T_1 est rouge**, le nombre de véhicules augmente avec un taux d’arrivée λ_1 , et cela sans qu’il y ait de sortie de véhicule. Ceci se traduit par l’équation suivante :

$$\frac{dl_1(t)}{dt} = \lambda_1, \quad \forall t \in]t_{2k}, t_{2k+1}[, k \in \mathbb{N}. \quad (1.24)$$

Ainsi, l'intégration de l'équation (1.24) donne :

$$l_1(t_{2k+1}) = l_1(t_{2k}) + \lambda_1 \delta_{2k}, \quad k \in \mathbb{N}. \quad (1.25)$$

Puisque $l_1(t) \geq 0$ pour tout t , l'équation (1.25) peut être réécrite comme :

$$l_1(t_{2k+1}) = \max(l_1(t_{2k}) + \lambda_1 \delta_{2k}, 0), \quad k \in \mathbb{N}. \quad (1.26)$$

- **si T_1 est vert**, le nombre de véhicules qui reste dans la file d'attente est donné par la différence des taux d'arrivée et de départ $\lambda_1 - \mu_1$. Ainsi, puisque $l_1(t) \geq 0$ pour tout t , cette situation se traduit par l'équation suivante :

$$\frac{dl_1(t)}{dt} = \begin{cases} \lambda_1 - \mu_1 & \text{si } l_1(t) > 0, \\ 0 & \text{si } l_1(t) = 0, \end{cases} \quad (1.27)$$

pour tout $t \in]t_{2k+1}, t_{2k+2}[$, $k \in \mathbb{N}$. Il vient aussitôt, après intégration de l'équation (1.27), que :

$$l_1(t_{2k+2}) = \max(l_1(t_{2k+1}) + (\lambda_1 - \mu_1)\delta_{2k+1}, 0). \quad (1.28)$$

En résumé, le comportement dynamique de la ligne L_1 est modélisé, pour tout $k \in \mathbb{N}$, par :

$$\begin{cases} l_1(t_{2k+1}) = \max(l_1(t_{2k}) + \lambda_1 \delta_{2k}, 0), & \text{si } T_1 \text{ est rouge,} \\ l_1(t_{2k+2}) = \max(l_1(t_{2k+1}) + (\lambda_1 - \mu_1)\delta_{2k+1}, 0), & \text{si } T_1 \text{ est vert.} \end{cases} \quad (1.29)$$

De la même manière, nous pouvons modéliser le comportement dynamique des lignes $L_{2,3,4}$. Ainsi, si nous posons :

$$\begin{aligned} x_k &= [l_1(t_k), \quad l_2(t_k), \quad l_3(t_k), \quad l_4(t_k)]^T, \\ b_1 &= [\lambda_1, \quad \lambda_2 - \mu_2, \quad \lambda_3, \quad \lambda_4 - \mu_4]^T, \\ b_2 &= [\lambda_1 - \mu_1, \quad \lambda_2, \quad \lambda_3 - \mu_3, \quad \lambda_4]^T, \end{aligned}$$

nous obtenons pour tout $k \in \mathbb{N}$ le système suivant :

$$\begin{cases} x_{2k+1} = \max(x_{2k} + b_1 \delta_{2k}, 0), & \text{si } T_i \text{ est rouge,} \\ x_{2k+2} = \max(x_{2k+1} + b_2 \delta_{2k+1}, 0), & \text{si } T_i \text{ est vert.} \end{cases} \quad (1.30)$$

Ainsi, le système opère alternativement suivant deux régimes caractérisés par deux équations différentielles de la forme (1.24) ou (1.27) et qui dépendent d'une commande discrète pouvant prendre soit la valeur "**rouge**" soit la valeur "**vert**". Les événements discrets dans ce système hybride sont les commutations du feu vert vers le feu rouge ou vice-versa.

Le critère d'optimisation

Les critères utilisés dans [34] pour définir le problème d'optimisation sont donnés par :

$$J_1 = \sum_{i=1}^4 \frac{\omega_i}{t_N - t_0} \int_{t_0}^{t_N} l_i(t) dt, \quad (1.31)$$

$$J_2 = \max_i \left(\frac{\omega_i}{t_N - t_0} \int_{t_0}^{t_N} l_i(t) dt \right), \quad (1.32)$$

$$J_3 = \sum_{i=1}^4 \frac{\omega_i}{t_N - t_0} \left(\sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \frac{l_i(t)}{\lambda_i^{(k)}} dt \right), \quad (1.33)$$

$$J_4 = \max_i \frac{\omega_i}{t_N - t_0} \left(\sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \frac{l_i(t)}{\lambda_i^{(k)}} dt \right), \quad (1.34)$$

où J_1 , J_2 , J_3 et J_4 représentent, respectivement, la moyenne des files d'attente, la moyenne de la plus grande file d'attente, le temps d'attente moyen de toutes les files d'attente et la plus longue durée de temps d'attente. $\omega_i > 0$ pour tout i est un facteur de pondération affectant chaque ligne. Son choix dépend de l'importance de chaque ligne au sein de l'intersection.

Le domaine d'optimisation

Dans ce travail, le domaine d'optimisation est contraint par :

$$\begin{cases} \delta_{min,r} \leq \delta_{2k} \leq \delta_{max,r} & \text{si } k \in \alpha(\mathbb{N}), \\ \delta_{min,v} \leq \delta_{2k+1} \leq \delta_{max,v} & \text{si } k \in \beta(\mathbb{N}), \\ 0 \leq x_k \leq x_{max} & \text{si } k = 1, \dots, \mathbb{N}, \end{cases} \quad (1.35)$$

où

$$\alpha(\mathbb{N}) = \{0, 1, \dots, \lfloor \frac{N-1}{2} \rfloor\},$$

$$\beta(\mathbb{N}) = \{0, 1, \dots, \lfloor \frac{N}{2} \rfloor - 1\}.$$

Les contraintes sur la commande impliquent que les périodes du feu rouge δ_{2k} et celles du feu vert δ_{2k+1} ne doivent pas dépasser certaines bornes fixées préalablement par l'ingénieur. Ceci est cohérent avec les objectifs de la commande. La contrainte sur l'état, quant à elle, traduit, d'une part le caractère positif des files d'attente, et d'autre part exclut l'existence des trajectoires de longueur infinie.

La stratégie de commande

La stratégie d'optimisation consiste à calculer pour un horizon N donné, une séquence de commutations optimale t_0, t_1, \dots, t_N telle que l'un des critères $J_i, i = 1, \dots, 4$ prenne une valeur minimale sous les contraintes (1.30)-(1.35).

Notons que le caractère hybride des contraintes (1.30) rend le problème d'optimisation difficile. Pour contourner cette difficulté, les auteurs ont transformé le problème de l'optimisation en un problème à complémentarité linéaire étendu. Ce problème est défini comme suit :

Étant donnés $A \in \mathbb{R}^{p \times n}$, $B \in \mathbb{R}^{q \times n}$, $c \in \mathbb{R}^p$, $d \in \mathbb{R}^q$ et m sous-ensembles $\phi_1, \phi_2, \dots, \phi_m$ de $\{1, 2, \dots, p\}$. Trouver un vecteur $x \in \mathbb{R}^n$ tel que :

$$\sum_{j=1}^m \prod_{i \in \phi_j} (Ax - c)_i = 0$$

sous les contraintes $Ax \geq c$ et $Bx = d$, ou montrer qu'aucun vecteur x n'existe.

Ainsi, l'interaction entre le problème d'optimisation et le problème ELCP devient alors plus claire. En effet, considérons l'équation (1.30) quand T_i est rouge. Pour un k quelconque, cette équation peut être réécrite sous la forme :

$$\begin{aligned} x_{2k+1} &\geq x_{2k} + b_1 \delta_{2k}, \\ x_{2k+1} &\geq 0, \\ (x_{2k+1})_i &= (x_{2k} + b_1 \delta_{2k})_i, \quad \text{ou} \quad (x_{2k+1})_i = 0, \quad i = 1, \dots, 4, \end{aligned}$$

ce qui est équivalent à :

$$\begin{aligned} x_{2k+1} - x_{2k} + b_1 \delta_{2k} &\geq 0, \\ x_{2k+1} &\geq 0, \\ \sum_{i=1}^4 (x_{2k+1} - x_{2k} + b_1 \delta_{2k})_i (x_{2k+1})_i &= 0. \end{aligned}$$

De la même manière, nous arrivons à la même équivalence pour l'équation (1.30) quand T_i est vert. Ainsi, pour N commutation des feux, si nous définissons les deux nouvelles variables $x^* = [x_1, \dots, x_N]^T$ et $\delta^* = [\delta_1, \dots, \delta_{N-1}]^T$, le système (1.30) peut être décrit par :

$$\begin{aligned} Ax^* + B\delta^* + c &\geq 0 \\ x^* &\geq 0 \\ (Ax^* + B\delta^* + c)^T x^* &= 0 \end{aligned}$$

où A, B sont des matrices de dimensions appropriées. De même, les contraintes (1.35) peuvent aussi être décrites par :

$$Ex^* + D\delta^* + f \geq 0.$$

En résumé, le problème de commande peut être formulé comme suit :

Minimiser J_i sous les contraintes suivantes :

$$Ax^* + B\delta^* + c \geq 0 \quad (1.36)$$

$$x^* \geq 0 \quad (1.37)$$

$$Ex^* + D\delta^* + f \geq 0 \quad (1.38)$$

$$(Ax^* + B\delta^* + c)^T x^* = 0 \quad (1.39)$$

Il est clair que ce problème est un cas particulier du problème ELCP.

De Schutter et De Moor ont développé un algorithme qui à la fois, évoque une description paramétrique de l'ensemble de solutions du problème ELCP et calcule les valeurs des paramètres qui emmènent à une minimisation du critère J_i .

Commentaires critiques : Dans ce travail, nous avons relevé les inconvénients suivants :

- La transformation des contraintes en un problème à complémentarité linéaire étendu pourrait être trompeuse. En effet, ceci suggérerait que le problème d'optimisation lié au caractère hybride du système est résolu. Or, il n'en est rien, car la complexité est simplement déplacée vers la recherche de solutions au problème ELCP qui peut s'avérer NP-difficile pour un grand horizon de commande N (comme c'est le cas pour la régulation des carrefours). La mise en œuvre de cette méthode devient alors problématique.
- L'algorithme développé par les auteurs ne fait pas appel à des propriétés dont la véracité pourrait être jugée sans vérification explicite. De plus, les solutions du problème ELCP sont paramétrées par des inconnus, ce qui rend la vérification encore plus critique. Il va sans dire que ce manque de simplicité d'interprétation vient accentuer davantage l'inconvénient souligné au point précédent.

Soulignons en plus un point qui nous paraît de nature à justifier notre contribution. En effet, le modèle développé par De Schutter et De Moor concerne une intersection saturée. Or, c'est précisément la situation de non-saturation que nous étudions.

1.7 Conclusion et objectifs de la thèse

Compte tenu de la complexité de l'implémentation des systèmes de régulation du trafic, il n'existe pas de méthode parfaite pour contrôler le flux dans une intersection routière. L'adéquation d'une méthode est très liée au condition du trafic (fluide, dense et saturé etc...). Les recherches qui ont été menées sur la régulation du trafic ont permis de développer plusieurs stratégies de régulations. Toutefois, la plupart d'entre elles sont destinées, de manière exclusive, aux méthodes d'optimisation en ligne pour minimiser un critère formulé à partir d'un comportement idéal souhaité. La minimisation du critère permet de déterminer la commande à appliquer pour mieux réaliser les objectifs de la commande. Cependant, et indépendamment d'une quelconque comparaison, ces travaux souffrent de trois inconvénients majeurs suivants :

- La faisabilité en temps réel de la commande.
- La robustesse vis-à-vis des perturbations qui sont inhérentes aux systèmes de transport.
- La non prise en compte des changements brusques des flux.

Ce constat milite donc pour une résolution des problèmes de régulation des carrefours, faisant d'avantage appel à des méthodes nouvelles.

Objectifs de la thèse : Dans le cadre de notre travail, nous adoptons une stratégie différente pour résoudre le problème de la régulation du trafic urbain par les feux de signalisation. Nous nous intéressons essentiellement à un travail en amont permettant d'éviter la congestion en forçant les files d'attente à ne pas dépasser le niveau de trafic correspondant à l'optimum opérationnel des lignes. Plus précisément, considérons les systèmes (1.9) et (1.12) avec une capacité maximale de chaque ligne regroupée dans le vecteur x^* . La congestion peut se produire dans le système si la longueur de la file d'attente de chaque ligne dépasse sa capacité. Une conséquence de ce type d'instabilité est l'accumulation illimitée de véhicules dans le système.

L'objectif de la commande est alors de trouver un contrôleur par retour d'état qui respecte la contrainte (1.13)

$$g_{min} \leq g_{e_i} \leq g_{max}$$

et est en mesure de satisfaire :

$$0 \leq x(k) \leq x^*, \quad \forall k \geq 0, \quad \forall 0 \leq x_0 \leq x^* \quad (1.40)$$

De cette manière, la congestion peut être évitée en forçant les files d'attente à ne jamais dépasser la capacité des lignes. Ainsi, le contrôle a priori de congestion des carrefours

isolés est un problème de régulation sous contraintes de type inégalité sur les variables de commande et de l'état. Pour résoudre ce problème :

1. nous proposons une commande par retour d'état basée sur le concept d'invariance positive. Deux approches sont utilisées :
 - La première fait appel aux inégalités matricielles linéaires (LMI). Leur succès vient du développement des méthodes dites du point intérieur (interior point methods) qui permettent d'apporter des solutions intéressantes par la manipulation de contraintes de façon systématique.
 - La deuxième approche utilise le concept de la (A, B) -invariance issu de la généralisation du théorème de Farkas. La mise en évidence de cette approche a ouvert des possibilités telles qu'il est devenu possible d'aborder des problèmes d'analyse et/ou de commande jusque là non envisageables.
2. Pour mieux contrôler les feux de signalisation, des informations précises en temps réel sur les conditions de circulation sont nécessaires. Dans ce sens, nous proposons l'approche neuronale pour estimer les flux d'entrée afin de garantir la faisabilité en temps réel de la commande proposée.

L'idée qui a guidé notre choix a été d'appliquer des techniques et des formulations qui ont prouvé une efficacité certaine dans des applications industrielles. Elles mettent en jeu des grandeurs et des caractéristiques concrètes qui s'interprètent clairement comme une interconnexion entre le système et son environnement. En outre, elles se conforment le plus aux objectifs fixés pour notre investigation, à savoir, l'obtention de commandes pour la régulation des carrefours permettant :

- La faisabilité en temps réel.
- Une flexibilité dans son utilisation.
- Une prise en compte des perturbations instantanées affectant le système.

CHAPITRE 2

APPROCHE DES INÉGALITÉS MATRICIELLES LINÉAIRES (LMI)

2.1 Introduction

Synthétiser une loi de commande consiste à trouver des entrées u qui satisfont au mieux le cahier des charges. Ce dernier peut imposer aussi bien des objectifs à atteindre que des contraintes à vérifier pendant le processus de régulation. Dès lors, l'intégration des contraintes dans la formulation du problème de commande semble assez naturelle.

En pratique, les limitations rencontrées dans le fonctionnement des processus se traduisent souvent par des contraintes de type inégalité sur les variables de commande et de l'état du modèle considéré. L'approche utilisée en général pour traiter le problème de la commande des systèmes dynamiques soumis à des contraintes est basée sur le concept d'invariance positive. Ce concept implique que toutes les trajectoires du système issues d'un domaine y restent dans ce domaine. Étant donné que les contraintes définissent en général un domaine dont les trajectoires du système ne doivent pas s'échapper, on cherche à garantir le respect des contraintes par la restriction des états initiaux admissibles en boucle fermée à un ensemble positivement invariant contenu dans le domaine des contraintes.

Notre contribution dans ce chapitre consiste à fournir des solutions à ce type de problème pour les systèmes de transport, qui soient aussi simples que possible à mettre en œuvre en temps réel. Notre première approche dans ce mémoire vise à appliquer les techniques des inégalités matricielles Linéaires (LMI). Leur succès vient du développement des méthodes dites du point intérieur (interior point methods) qui permettent d'apporter

des solutions intéressantes par la manipulation de contraintes de façon systématique. La mise en évidence de l'approche LMI a ouvert des possibilités telles qu'il est devenu possible d'aborder des problèmes d'analyse et/ou de commande jusque là non envisageables. En outre, de nombreux logiciels (solveurs) conviviaux existent dont certains offrent une interface avec MATLAB.

Dans un premier temps, nous rappelons quelques définitions et résultats utiles donnés dans la littérature et nous donnons certains résultats préliminaires relatifs au concept d'invariance positive. Ensuite, nous précisons la définition du problème de commande qui nous concerne. Une fois que le problème est bien défini, nous donnons sous forme LMI des conditions nécessaires et suffisantes pour qu'un ensemble ellipsoïdal soit à la fois un ensemble positivement invariant de notre système et inclus dans l'ensemble des contraintes sur l'état et la commande. Ces résultats sont appliqués aux modèles de transport établis dans le chapitre 1. Deux cas sont étudiés : celui où le cycle des feux est constant est d'abord analysé. Ensuite, nous nous intéressons au cas où le cycle des feux est variable afin d'adapter la stratégie de commande de manière efficace aux variations permanentes du trafic.

2.2 Rappels mathématiques et résultats préliminaires

Nous définissons ici différentes notations concernant les vecteurs et les opérateurs positifs. Ces notations sont conformes à celles classiquement introduites dans la littérature. Nous introduisons tout d'abord deux relations d'ordre partiel dans \mathbb{R}^n : $>$ et \geq . Considérons deux vecteurs x et y de \mathbb{R}^n ; alors :

$$- x > y \Leftrightarrow \forall i \in \{1, \dots, n\}, x_i > y_i$$

$$- x \geq y \Leftrightarrow \forall i \in \{1, \dots, n\}, x_i \geq y_i$$

Il est clair que : $x > y \Rightarrow x \geq y$. Ces relations d'ordre partiel nous permettent de définir les vecteurs positifs de \mathbb{R}^n :

$$- x \text{ est non-négatif, i.e., } x \geq 0$$

$$- x \text{ est positif, i.e., } x > 0$$

Définissons les orthants non-négatifs et positifs, que nous noterons respectivement \mathbb{R}_+^n et \mathbb{R}_{+*}^n :

$$\mathbb{R}_+^n = \{x \in \mathbb{R}^n, x \geq 0\}$$

$$\mathbb{R}_{+*}^n = \{x \in \mathbb{R}^n, x > 0\}$$

De la même façon, nous définissons les relations $>$ et \geq pour les matrices ($m \times n$) :

$$- A > B \Leftrightarrow \forall (i, j), a_{ij} > b_{ij}$$

$$- A \geq B \Leftrightarrow \forall (i, j), a_{ij} \geq b_{ij}$$

Nous définissons donc les matrices :

$$- A \text{ est non-négative, i.e., } A \geq 0$$

$$- A \text{ est positif, i.e., } A > 0$$

Clairement : $A > 0 \Rightarrow A \geq 0$.

2.2.1 Sur l'invariance positive des ensembles

Considérons le système discret suivant :

$$x(k+1) = f(x(k)), \quad x(k_0) = x_0, \quad (2.1)$$

où $x(k) \in \mathbb{R}^n$ est le vecteur d'état du système à l'instant $k \in \mathbb{N}$, k_0 est l'instant initial et x_0 est l'état initial. Le vecteur $f(\cdot)$ est une application de \mathbb{R}^n dans \mathbb{R}^n . L'équation (2.1) définit une transformation ponctuelle qui fait correspondre, pour un k donné, à un point $M(x_k)$ de l'espace à n dimension un point $M(x_{k+1})$ du même espace.

Soit $\mathcal{S} \subset \mathbb{R}^n$. Dans toute la suite, on désigne par :

- $\bar{\mathcal{S}}$ la fermeture de \mathcal{S} ; c'est l'intersection de tous les fermés contenant \mathcal{S} , c'est donc le plus petit fermé contenant \mathcal{S} .
- $Int(\mathcal{S})$ l'intérieur de \mathcal{S} ; c'est la réunion de tous les ouverts inclus dans \mathcal{S} , c'est donc le plus grand ouvert contenu dans \mathcal{S} .
- $\partial\mathcal{S}$ la frontière de \mathcal{S} ; c'est sa fermeture privée de son intérieur.

Une propriété intéressante à connaître est la façon avec laquelle la solution de (2.1) se comporte vis à vis d'un ensemble \mathcal{S} .

Définition 2.1 Soit $\mathcal{S} \subset \mathbb{R}^n$. \mathcal{S} est dit positivement invariant par rapport à la solution du système (2.1) si et seulement si :

$$x_0 \in \mathcal{S} \Rightarrow f(x(k)) \in \mathcal{S}$$

pour tout $(k_0, k) \in \mathbb{N} \times \mathbb{N}$. ▶

Un ensemble invariant piège donc les trajectoires : si une trajectoire commence dans \mathcal{S} , elle reste dans \mathcal{S} pour tout $k \in \mathbb{N}$. Pour les systèmes dynamiques, une trajectoire est un ensemble invariant, c'est l'exemple le plus simple d'un tel ensemble.

Il se pose le problème de savoir dans quelles conditions l'ensemble \mathcal{S} est positivement invariant par rapport à la solution du système (2.1). Pour expliquer comment on peut répondre à cette question, il nous faut commencer par la définition suivante.

Définition 2.2 Soit \mathcal{S} un voisinage ouvert de $x = 0$. Une fonction $V(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$ est dite *fonction frontière* de l'ensemble \mathcal{S} si et seulement si \mathcal{S} possède une frontière et il existe $\xi \in \mathbb{R}$ tel que pour tout $x \in \mathbb{R}^n$:

$$V(x) < \xi \Leftrightarrow x \in \mathcal{S} \quad \text{et} \quad V(x) = \xi \Leftrightarrow x \in \partial\mathcal{S}.$$

Exemple 2.1 Soit

$$\mathcal{S} = \{x \in \mathbb{R}^n : \|x\|^2 < 1 + \xi\}$$

où $\xi > -1$. Il est clair que \mathcal{S} est un voisinage ouvert et connexe de $x = 0$. Sa fonction frontière est donnée par :

$$V(x) = \|x\|^2 - 1$$

Ainsi, nous avons $V(x) < \xi, \forall x \in \mathcal{S}$ et $V(x) = \xi, \forall x \in \partial\mathcal{S}$.

Le résultat suivant est appelé à jouer, comme nous allons le voir, un rôle fondamental dans la construction des théorèmes de l'invariance positive des ensembles que nous allons étudier.

Lemme 2.1 Soit $\mathcal{S} \subseteq \mathbb{R}^n$ un ensemble ouvert et $V(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$ sa fonction frontière. Pour que la fermeture $\overline{\mathcal{S}}$ de \mathcal{S} soit positivement invariant par rapport à la solution du système (2.1), il est nécessaire et suffisant que sa fonction frontière $V(\cdot)$ vérifie :

$$\Delta V(x) \leq 0, \quad \forall x \in \overline{\mathcal{S}}. \quad (2.2)$$

Preuve: (Suffisance :) Soit $x_0 \in \overline{\mathcal{S}}$ un point arbitrairement choisi. Si nous supposons qu'il existe un instant $k_i \in \mathbb{N}$ tel que $x_i = x(k_i; x_0) \in \overline{\mathcal{S}}$ et $x_{i+1} = x(k_{i+1}; x_i) = x(k_{i+1}; x_0) \notin \overline{\mathcal{S}}$. Alors, en vertu de la définition 2.2, nous avons $V(x_i) \leq \zeta$ et $V(x_{i+1}) > \zeta$. Il vient :

$$\Delta V(x_i) = V(x_{i+1}) - V(x_i) > 0.$$

Ceci contredit la condition (2.2).

(Nécessité :) soit $\overline{\mathcal{S}}$ positivement invariant par rapport à la solution du système et soit $x_i = x(k_i; x_0) \in \partial\mathcal{S}$. Ainsi, nous avons $V(x_i) = \xi$ et $V(x_{i+1}) \leq \xi$. Supposons que :

$$\Delta V(x_i) = V(x_{i+1}) - V(x_i) > 0$$

Il vient :

$$V(x_{i+1}) - V(x_i) > 0 \Rightarrow V(x_{i+1}) > \xi$$

ce qui implique que $x_{i+1} \notin \overline{\mathcal{S}}$ et prouve le Lemme. ■

2.2.2 Sur les Inégalités Matricielles Linéaires (LMI)

Une inégalité matricielle linéaire (LMI) est une relation de type :

$$F(x) = F_0 + x_1 F_1 + \dots + x_n F_n = F_0 + \sum_{i=1}^n x_i F_i \succeq 0 \quad (2.3)$$

où $x = (x_1, \dots, x_n)^T$ est un vecteur réel à n composantes et les matrices $F_i \in \mathbb{R}^{n \times n}$ sont symétriques. F_i sont des données, x est l'inconnue. Le signe d'inégalité \succeq signifie que la matrice symétrique $F(x)$ doit être semi-définie positive. Il est possible de traiter des inégalités strictes du type $F(x) \succ 0$ qui signifie que la matrice $F(x)$ est définie positive. On consultera avec profit l'ouvrage de [35] qui constitue un très bon répertoire de nombreux problèmes faisant appel aux LMIs.

Considérons à titre d'exemple le problème de la recherche d'une matrice réelle $P = P^T$ d'ordre 2 définie positive. La matrice P dépend alors de trois paramètres $x_i, i = 1, 2, 3$ et peut s'écrire :

$$P = P^T = \begin{pmatrix} x_1 & x_2 \\ x_2 & x_3 \end{pmatrix} = x_1 \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}}_{F_1} + x_2 \underbrace{\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}}_{F_2} + x_3 \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}}_{F_3} \succ 0$$

Une propriété remarquable des LMIs est la possibilité de regrouper plusieurs LMIs $F^1(x) \succeq 0, \dots, F^q(x) \succeq 0$ en une seule LMI bloc-diagonale :

$$F(x) = \begin{bmatrix} F^1(x) & & \\ & \ddots & \\ & & F^q(x) \end{bmatrix} \succeq 0$$

Cette propriété découle du fait que les valeurs propres d'une matrice diagonale par bloc sont constituées par les valeurs propres des matrices sur la diagonale.

Quelques outils pour LMI

Certaines inégalités non-linéaires peuvent être également transformées en LMIs à l'aide du complément de Schur. En effet, soit

$$F(x) = \begin{bmatrix} R(x) & S(x) \\ S^T(x) & Q(x) \end{bmatrix}$$

où $R(x) = R^T(x), Q(x) = Q^T(x)$ inversible, et $S(x)$ dépend linéairement de x .

Définition 2.3 Le complément de Schur du bloc $Q(x)$ de la matrice $F(x)$ est la matrice :

$$R(x) - S(x)Q^{-1}(x)S^T(x)$$

Le complément de Schur apparaît naturellement comme le résultat d'un échantillonnage partiel de $F(x)$ à l'aide du pivot de Gauss. Nous avons alors le résultat suivant :

Lemme 2.2 ([35]) *On suppose que $Q(x)$ est définie positive (respectivement semi-définie positive).*

- $F(x) \succ 0$ si et seulement si $R(x) - S(x)Q^{-1}(x)S^T(x) \succ 0$.
- $F(x) \succeq 0$ si et seulement si $R(x) - S(x)Q^{-1}(x)S^T(x) \succeq 0$.

Remarquons à partir de ce lemme que $R(x) \succeq 0$ est une condition nécessaire pour que $F(x) \succeq 0$. Pour le montrer, il suffit de remarquer que :

$$(y^T, z^T) \begin{bmatrix} R & S \\ S^T & Q \end{bmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = y^T R y + 2y^T S z + z^T Q z \succeq 0$$

implique en posant $z = 0$, $y^T R y \succeq 0$.

Soient maintenant deux matrices $M; N$. Ces deux matrices sont dites congruentes s'il existe une matrice inversible K telle que $M = K^T N K$. On a alors la proposition suivante.

Lemme 2.3 ([35]) *Soient M et N deux matrices symétriques congruentes alors $M \succeq 0$ si et seulement si $N \succeq 0$.*

2.2.3 Sur l'optimisation sous contraintes inégalités

considérons le problème de maximisation suivant :

$$\begin{aligned} & \max f(x) \\ & \text{sous les contraintes } h_j(x) \leq c_j, \quad j = 1, \dots, m \end{aligned}$$

On définit le Lagrangien L par la fonction suivante :

$$L(x, \lambda) = f(x) - \sum_{j=1}^m \lambda_j (h_j(x) - c_j)$$

où les variables λ_j sont les multiplicateurs de Lagrange associés à chaque contrainte j . Notons qu'il est très facile de passer d'un problème de minimisation sous contraintes prenant la forme d'inégalité à un problème de maximisation. En effet, il suffit de remarquer que le problème de minimisation $\min f(x)$: est équivalent au problème maximisation $\max -f(x)$.

Soit \bar{x} la solution du problème de maximisation de la fonction $f(x)$. Deux situations sont envisageables pour chaque contrainte j .

1. soit $h_j(\bar{x}) = c_j$: dans ce cas, on dit que la contrainte j est saturée à l'optimum ;
2. soit $h_j(\bar{x}) < c_j$: dans ce cas, on dit que la contrainte j est non saturée à l'optimum.

Conditions de qualification des contraintes : Pour pouvoir utiliser le Lagrangien dans la résolution d'un programme d'optimisation sous contraintes prenant la forme d'inégalité, il suffit que l'une des conditions suivantes soit vérifiée :

- a) Soit $s \leq m$ le nombre de contraintes saturées à l'optimum \bar{x} . Si la matrice jacobienne de ces s fonctions contraintes, de taille (s, n) , est de rang s lorsqu'elle est évaluée à l'optimum \bar{x} , alors la condition de qualification des contraintes est vérifiée.
- b) Les fonctions contraintes h_j , $j = 1, \dots, s$ sont toutes linéaires.

Conditions du premier ordre (Kuhn et Tucker) : On suppose que les contraintes de qualification sont vérifiées. Si le vecteur $\bar{x} \in \mathbb{R}^n$ est une solution du problème de maximisation de la fonction f , alors il existe un unique vecteur $\bar{\lambda} \geq 0$ tel que \bar{x} vérifie les conditions suivantes :

$$\begin{aligned} \frac{\partial L(\bar{x}, \bar{\lambda})}{\partial x} &= 0 \\ \bar{\lambda}_j (h_j(\bar{x}) - c_j) &= 0 \Leftrightarrow \bar{\lambda}_j = 0, \text{ ou } h_j(\bar{x}) - c_j = 0, \quad j = 1, \dots, m \\ \frac{\partial L(\bar{x}, \bar{\lambda})}{\partial \lambda} &\geq 0 \Leftrightarrow h_j(\bar{x}) \leq c_j, \quad j = 1, \dots, m \end{aligned}$$

Notons que ces conditions n'excluent pas la possibilité que $\bar{\lambda}_j = 0$ et $h_j(\bar{x}) - c_j = 0$ simultanément.

Conditions suffisantes du second ordre pour un optimum global : Supposons qu'il existe un $\bar{x} \in \mathbb{R}^n$ qui vérifie les conditions du premier ordre.

1. Si f est concave et les h_j sont convexes, \bar{x} est un maximum global.
2. Si f est convexe et les h_j sont concaves, \bar{x} est un minimum global.

En pratique, la résolution des conditions de Kuhn et Tucker est compliquée par le fait qu'il faut envisager successivement toutes les configurations possibles : toutes les contraintes sont saturées à l'équilibre, toutes sauf une, deux, ..., aucune (tous les $\bar{\lambda}_j$ sont nuls à l'équilibre). Pour trouver la bonne solution, il faut procéder par élimination en montrant que parmi l'ensemble de ces possibilités, certaines aboutissent à des contradictions.

Sur ce, nous terminons notre aperçu général des rappels mathématiques pour passer aux sections suivantes à la définition du problème de la commande et sa solution en terme de LMI.

2.3 Position du problème de la commande

2.3.1 Cas où le cycle est constant

Dans le cas où le cycle des phases est constant, le modèle du carrefour à feux établi au chapitre 1 est de la forme (voir (1.7) chapitre 1) :

$$y(k+1) = y(k) + Bu(k) + L \quad (2.4)$$

$y \in \mathbb{R}_+^n$ est le vecteur représentant les longueurs des files d'attente dans chaque phase au début du cycle k , $u \in \mathbb{R}_+^m$ est le vecteur de commande représentant les feux de signalisation, $B \in \mathbb{R}^{m \times n}$ et $L \in \mathbb{R}^n$. Les capacités maximales de chaque ligne sont regroupées dans le vecteur y^* . Ainsi, les contraintes sur l'état sont définies par :

$$\mathbb{E}_y = \{y \in \mathbb{R}_+^n / 0 \leq y(k) \leq y^*, \quad \forall k \geq 0, \quad \forall 0 \leq y_0 \leq y^*\} \quad (2.5)$$

L'ensemble des commandes admissibles quant à lui est défini par :

$$\mathbb{U}_u = \{u \in \mathbb{R}_+^m / u_{min} \leq u \leq u_{max}; \quad u_{min} > 0\} \quad (2.6)$$

Les domaines \mathbb{E}_y et \mathbb{U}_u sont des ensembles polyédraux dissymétriques bornés de \mathbb{R}^n et \mathbb{R}^m respectivement.

Soulignons tout de suite que les contraintes (2.5) et (2.6) sont de nature fondamentalement différente. En effet, la contrainte (2.5) exprime une exigence du cahier des charges du contrôleur, alors que la contrainte (2.6) est une contrainte technique imposée par l'ingénieur pour garantir la faisabilité de son contrôleur. Ainsi, violer les contraintes (2.5), c'est ne pas respecter l'objectif de la commande, alors que trouver un autre schéma de commande dans lequel (2.5) n'est pas nécessaire, c'est réaliser les objectifs autrement.

Il va sans dire que, contrairement à la théorie des systèmes positifs, la contrainte de positivité de l'état n'impose aucune restriction sur la matrice B et le vecteur L . Le problème de synthèse peut alors être interprété comme contraindre l'état d'être seulement positif. Ce qui confère aux résultats proposés dans ce chapitre un caractère général.

Dans la suite, nous supposons qu'il existe un état nominal y_N pour lequel il existe une commande $u_N \in \mathbb{U}_u$ telle que (voir remarque 1.1, chapitre 1) :

$$Bu_N + L = 0 \quad (2.7)$$

Généralement, un minimum de connaissances sur le processus à commander permet de dire quels sont les états d'équilibre intéressants. Nous supposons donc que l'ensemble de ces états d'équilibre est connu. Dans ces conditions, posons comme nouveau vecteur de commande :

$$v(k) = u(k) - u_N \quad (2.8)$$

et comme nouveau vecteur d'état :

$$x(k) = y(k) - y_N \quad (2.9)$$

Les équations (2.4), (2.7), (2.8) et (2.9) donnent alors le système suivant :

$$x(k+1) = x(k) + Bv(k); \quad x \in \mathbb{R}^n; v \in \mathbb{R}^m \quad (2.10)$$

Les contraintes sur $u(k)$ se transportent sur $v(k)$ en :

$$\mathbb{U}_v = \{v \in \mathbb{R}^m / -v_2 \leq v \leq v_1; \quad v_1, v_2 > 0\} \quad (2.11)$$

avec

$$v_2 = u_N - u_{min}; \quad v_1 = u_{max} - u_N$$

L'ensemble \mathbb{E}_y quant à lui se transforme naturellement en :

$$\mathbb{E}_x = \{x \in \mathbb{R}^n / -x_2 \leq x(k) \leq x_1; \quad x_1, x_2 > 0\} \quad (2.12)$$

avec

$$x_2 = y_N; \quad x_1 = y^* - y_N$$

Notons tout de suite que la positivité de l'état ainsi que celle de la commande ne sont plus demandées. Il est clair que $x \in \mathbb{E}_x; v \in \mathbb{U}_v$ si et seulement si $y \in \mathbb{E}_y; u \in \mathbb{U}_u$.

La commande du système dynamique (2.10) aura pour expression :

$$v(k) = Gx(k) \quad (2.13)$$

où $G \in \mathbb{R}^{m \times n}$. Dans ce cas, le système (2.10) en boucle fermée est donné par :

$$x(k+1) = (I + BG)x(k) \quad (2.14)$$

En outre, l'existence des contraintes (2.11) implique que le retour d'état (2.13) n'est admissible que dans une région déterminée de l'espace d'état, c'est-à-dire :

$$\mathbb{U}_v = \{x \in \mathbb{R}^n / -v_2 \leq Gx \leq v_1; \quad v_1 > 0; v_2 > 0\} \quad (2.15)$$

Ce domaine représente l'ensemble des états admissibles pour la commande pour G choisie telle que $\text{rang}(G) = m$. Il est polyédral dissymétrique contenant l'origine.

Toutefois, toute condition initiale émanant de \mathbb{E}_x peut parfaitement donner une trajectoire sortant de l'ensemble \mathbb{U}_v . De ce fait, il est nécessaire d'assurer l'invariance positive des ensembles \mathbb{E}_x et \mathbb{U}_v .

Rappelons toutefois que quelques auteurs (ainsi, par exemple gutman [36], bitsoris [37], cheganças [38], Benzaouia [39], Rami [40]) ont étudié particulièrement ce problème

dans le cas où les contraintes sont symétriques et concernent uniquement la commande. Ceci suggérerait que les résultats obtenus par ses auteurs restent valables pour notre problème. Or, il n'en est rien, car dans notre formulation, deux éléments supplémentaires sont à prendre en considération. D'abord les contraintes supplémentaires sur l'état et la dissymétrie de nos contraintes. Toutefois, le travail de ces auteurs fut pour nous une source d'inspiration certaine.

L'idée de base de ces auteurs (sauf Rami) était de trouver un domaine de conditions initiales \mathbb{E}_x et un ensemble $\Omega(x)$ qui soit positivement invariant tels que $\mathbb{E}_x \subseteq \Omega(x) \subseteq \mathbb{U}_v$. Or cette démarche n'est pas transposable à notre problème puisque l'ensemble de conditions initiales \mathbb{E}_x est imposé. L'idée alors est de chercher à garantir le respect des contraintes sur l'état et la commande par la restriction des conditions initiales de l'état en boucle fermée à un ensemble positivement invariant $\Omega(x)$ contenu dans $\mathbb{E}_x \cap \mathbb{U}_v$. De cette manière, si $x_0 \in \Omega(x)$ alors on est sûr que $x(k) \in \mathbb{E}_x$ et $v(k) \in \mathbb{U}_v$. l'objectif de la commande est donc le suivant :

Objectif : *Trouver un contrôleur par retour d'état $v = Gx$ et un ensemble $\Omega(x)$ tels que :*

1. $\Omega(x)$ soit positivement invariant par rapport au système (2.14) et,
2. $\Omega(x) \subseteq \mathbb{E}_x \cap \mathbb{U}_v$

Pour trouver l'ensemble $\Omega(x)$, nous allons associer au système (2.14) une fonction de Lyapunov $\mathcal{V}(x)$ qui donnera un ensemble de la forme :

$$\Omega(x) = \{x \in \mathbb{R}^n \mid \mathcal{V}(x) \leq \lambda\} \quad (2.16)$$

Ainsi, le problème posé est le suivant : Trouver un gain G tel que l'on puisse associer au système 2.14 une fonction de Lyapunov $\mathcal{V}(x)$ et un réel $\lambda > 0$ pour lesquels $\Omega(x)$ soit positivement invariant par rapport au système (2.14) et :

$$\Omega(x) \subseteq \mathbb{E}_x \cap \mathbb{U}_v \quad (2.17)$$

2.4 Solution du problème de la commande

Pour la solution du problème de la commande, nous allons prendre $\mathcal{V}(x) = x^T P x$, où $P = P^T$ est une matrice symétrique définie positive. Nous pouvons aussi supposer sans perte de généralité que $\lambda = 1$ (quitte à prendre $P^* = \frac{1}{\lambda} P$). Ainsi, le domaine $\Omega(x)$ peut s'écrire comme :

$$\Omega(x) = \{x \in \mathbb{R}^n \mid x^T P x \leq 1\} \quad (2.18)$$

C'est un ellipsoïde centré à l'origine de fonction frontière $\mathcal{V}(x) = x^T P x$.

Dans la suite, nous adoptons une démarche qui consiste à séparer l'étude de l'invariance positive de $\Omega(x)$ et l'étude de l'inclusion $\Omega(x) \subseteq \mathbb{E}_x \cap \mathbb{U}_v$. Il s'agit de simplifier la présentation tout en traitant l'ensemble des aspects du problème. Une fois que ces deux études sont présentées séparément, leur intégration dans une formulation complète devient évidente.

2.4.0.1 Sur l'invariance positive de $\Omega(x)$

Dans cette partie, nous allons étudier les conditions sur G et P pour que $\Omega(x)$ soit positivement invariant par rapport au système (2.14). En effet, d'après le lemme 2.1, l'invariance positive de $\Omega(x)$ est équivalente à :

$$\Delta \mathcal{V}(x) = \mathcal{V}(x(k+1)) - \mathcal{V}(x) \leq 0$$

Ceci donne :

$$\Delta \mathcal{V}(x) = x^T \left[(I + BG)^T P (I + BG) - P \right] x \leq 0$$

ce qui implique que la matrice $(I + BG)^T P (I + BG) - P$ doit être semi-définie négative, c'est-à-dire :

$$(I + BG)^T P (I + BG) - P \preceq 0$$

ou d'une manière équivalente :

$$P - (I + BG)^T P (I + BG) \succeq 0 \quad (2.19)$$

Or, on sait que $A \succeq 0$ est équivalente à $P^{-1} A P^{-1} \succeq 0$, car $x^T A x = y^T P^{-1} A P^{-1} y \geq 0$, où $y = P x$. Ainsi, (2.19) est équivalente à :

$$P^{-1} - P^{-1} (I + BG)^T P (I + BG) P^{-1} \succeq 0$$

Il s'ensuit que :

$$P^{-1} - (P^{-1} + B G P^{-1})^T P (P^{-1} + B G P^{-1}) \succeq 0$$

Posons $Q = P^{-1}$. Il vient alors :

$$Q - (Q + B G Q)^T Q^{-1} (Q + B G Q) \succeq 0$$

Si nous appliquons maintenant le changement de variable $Y = G Q$, la dernière inégalité devient :

$$Q - (Q + B Y)^T Q^{-1} (Q + B Y) \succeq 0$$

L'application du lemme 2.2 de Schur avec $R = Q$, $S = (Q + BY)^T$ donne :

$$\begin{pmatrix} Q & Q + Y^T B^T \\ Q + BY & Q \end{pmatrix} \succeq 0, \quad Q = P^{-1}, Y = GQ \quad (2.20)$$

Ainsi, l'invariance positive de $\Omega(x)$ par rapport au système (2.14) est équivalente à la faisabilité de la LMI (2.20). D'où le lemme suivant.

Lemme 2.4 *Pour que $\Omega(x)$ soit positivement invariant par rapport au système (2.14) il est nécessaire et suffisant que la LMI suivante soit faisable :*

$$\begin{pmatrix} Q & Q + Y^T B^T \\ Q + BY & Q \end{pmatrix} \succeq 0, \quad Q = P^{-1}, Y = GQ \quad (2.21)$$

Remarquons à partir de ce lemme que pour calculer le gain G , Il suffit donc de résoudre la LMI ci-dessus en $Y = GQ$ et Q puis de calculer $G = YQ^{-1}$.

2.4.0.2 Sur l'inclusion $\Omega(x) \subseteq \mathbb{E}_x \cap \mathbb{U}_v$

Regardons d'abord les nouvelles conditions sur P pour que $\Omega(x) \subseteq \mathbb{E}_x$. En effet, Désignons par \mathbb{X}^1 et \mathbb{X}^2 les matrices diagonales de termes diagonaux x_1^i et x_2^i respectivement. Ainsi nous avons :

$$(\mathbb{X}^1)^{-1} = \begin{pmatrix} \frac{1}{x_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{x_1^2} & 0 & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{1}{x_1^n} \end{pmatrix}, \quad (\mathbb{X}^2)^{-1} = \begin{pmatrix} \frac{1}{x_2} & 0 & \cdots & 0 \\ 0 & \frac{1}{x_2^2} & 0 & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{1}{x_2^n} \end{pmatrix}$$

Il vient alors :

$$x \leq x_1 \Leftrightarrow (\mathbb{X}^1)^{-1}x \leq e \quad \text{et} \quad -x \leq x_2 \Leftrightarrow -(\mathbb{X}^2)^{-1}x \leq e$$

où $e = (1, 1, \dots, 1)^T$ est le vecteur dont toutes ses composantes sont égales à 1. Il s'ensuit que l'ensemble \mathbb{E}_x peut s'écrire sous la forme suivante :

$$\mathbb{E}_x = \left\{ x \in \mathbb{R}^n \mid \begin{pmatrix} (\mathbb{X}^1)^{-1} \\ -(\mathbb{X}^2)^{-1} \end{pmatrix} x \leq e \right\} = \left\{ x \in \mathbb{R}^n \mid Fx \leq e \right\}$$

où

$$F = \begin{pmatrix} F^1 = (\mathbb{X}^1)^{-1} \\ F^2 = -(\mathbb{X}^2)^{-1} \end{pmatrix}$$

L'ensemble \mathbb{E}_x peut aussi s'écrire sous la forme :

$$\mathbb{E}_x = \left\{ x \in \mathbb{R}^n \mid F_i x \leq 1, \quad i = 1, \dots, 2n \right\} \quad (2.22)$$

où F_i est la $i^{\text{ième}}$ ligne de la matrice F . Ceci fait, il est clair que la condition $\Omega(x) \subseteq \mathbb{E}_x$ est équivalente à :

$$\max_{x \in \Omega(x)} \{F_i x\} \leq 1 \quad (2.23)$$

Il s'agit maintenant de connaître la valeur maximale de la fonction linéaire $F_i x$ sur l'ensemble $\Omega(x)$, c'est-à-dire :

$$\begin{aligned} & \max F_i x \\ & \text{sous la contrainte } x^T P x \leq 1 \end{aligned}$$

Pour résoudre ce problème, nous allons appliquer les conditions de Kuhn et Tucker (voir section 2.2.3). En effet, soit le Lagrangien L :

$$L(x, \lambda) = F_i x - \lambda(x^T P x - 1)$$

Alors, si \bar{x} est la solution recherchée, il existe un scalaire $\bar{\lambda} \geq 0$ tel que :

$$\frac{\partial L}{\partial x} = F_i^T - 2\bar{\lambda} P \bar{x} = 0 \quad (2.24)$$

$$\bar{\lambda}(\bar{x}^T P \bar{x} - 1) = 0 \Leftrightarrow \bar{\lambda} = 0, \text{ ou } \bar{x}^T P \bar{x} - 1 = 0 \quad (2.25)$$

$$\frac{\partial L}{\partial \lambda} \geq 0 \Leftrightarrow \bar{x}^T P \bar{x} \leq 1 \quad (2.26)$$

Pour déterminer les solutions de ce système, il faut envisager successivement tous les cas de figure possibles portant sur la saturation des contraintes et procéder par élimination.

Cas 1 : $\bar{x}^T P \bar{x} = 1$ (la contrainte est saturée à l'optimum). Dans ce cas, on a :

$$\text{rang}\left(\frac{\partial \bar{x}^T P \bar{x}}{\partial x}\right) = \text{rang}(2P\bar{x}) = 1, \quad \forall \bar{x} \in \mathbb{R}^n$$

Ainsi, la contrainte de qualification est vérifiée. Il s'ensuit alors que dans ce cas nous avons $\bar{\lambda} > 0$. Il vient à partir de (2.24) :

$$\bar{x} = \frac{1}{2\bar{\lambda}} P^{-1} F_i^T$$

En multipliant (2.24) par x^T il vient :

$$\bar{\lambda} = \frac{1}{2} (F_i P^{-1} F_i^T)^{\frac{1}{2}}$$

En remplaçant $\bar{\lambda}$ dans \bar{x} on obtient :

$$\bar{x} = (F_i P^{-1} F_i^T)^{-\frac{1}{2}} P^{-1} F_i^T$$

qui est une solution possible du problème de maximisation de la fonction $F_i x$ sur l'ensemble $\Omega(x)$.

Cas 2 : $\bar{x}^T P \bar{x} < 1$ (la contrainte n'est pas saturée à l'optimum). Dans ce cas, on a $\bar{\lambda} = 0$. Or, l'équation (2.24) donne $F_i^T \bar{x} = 0$, ce qui viole le fait que $F_i \neq 0$.

Finalement, on en déduit que les conditions de Kuhn et Tucker admettent une seule solution :

$$\bar{x} = (F_i P^{-1} F_i^T)^{-\frac{1}{2}} P^{-1} F_i^T$$

Or, $F_i x$ est concave et les contraintes sont convexes, donc \bar{x} est un maximum global. Par conséquent, la valeur maximale de $F_i x$ sur l'ensemble $\Omega(x)$ est donnée par :

$$\max_{x \in \Omega(x)} \{F_i x\} = (F_i P^{-1} F_i^T)^{\frac{1}{2}}$$

Ainsi, d'après (2.23), la condition $\Omega(x) \subseteq \mathbb{E}_x$ est équivalente à :

$$(F_i P^{-1} F_i^T)^{\frac{1}{2}} \leq 1$$

qui à son tour équivalente à :

$$F_i P^{-1} F_i^T \leq 1$$

d'où :

$$1 - F_i P^{-1} F_i^T \geq 0, \quad i = 1, \dots, 2n$$

Puisque $1 - F_i P^{-1} F_i^T \in \mathbb{R}$, l'application du lemme 2.2 de Schur avec $R = 1$ et $S = F_i$ donne la LMI suivante :

$$\begin{pmatrix} 1 & F_i \\ F_i^T & P \end{pmatrix} \succeq 0, \quad i = 1, \dots, 2n$$

Pour harmoniser cette LMI avec la LMI (2.21) en terme de Q , nous allons appliquer le lemme 2.3 de congruence. En effet, Soit

$$K = \begin{pmatrix} I & \mathbb{O} \\ \mathbb{O} & Q \end{pmatrix}$$

Cette matrice est bien évidemment symétrique et inversible puisque $Q = P^{-1}$. Il s'ensuit d'après le lemme 2.3 de congruence que :

$$\begin{pmatrix} 1 & F_i \\ F_i^T & P \end{pmatrix} \succeq 0, \quad \iff \begin{pmatrix} I & \mathbb{O} \\ \mathbb{O} & Q \end{pmatrix} \begin{pmatrix} 1 & F_i \\ F_i^T & Q^{-1} \end{pmatrix} \begin{pmatrix} I & \mathbb{O} \\ \mathbb{O} & Q \end{pmatrix} \succeq 0$$

Or :

$$\begin{pmatrix} I & \mathbb{O} \\ \mathbb{O} & Q \end{pmatrix} \begin{pmatrix} 1 & F_i \\ F_i^T & Q^{-1} \end{pmatrix} \begin{pmatrix} I & \mathbb{O} \\ \mathbb{O} & Q \end{pmatrix} = \begin{pmatrix} 1 & F_i Q \\ Q F_i^T & Q \end{pmatrix}$$

Il vient aussitôt que :

$$\begin{pmatrix} 1 & F_i \\ F_i^T & P \end{pmatrix} \succeq 0, \quad \iff \begin{pmatrix} 1 & F_i Q \\ Q F_i^T & Q \end{pmatrix} \succeq 0$$

d'où le lemme suivant.

Lemme 2.5 Pour que $\Omega(x) \subseteq \mathbb{E}_x$, il est nécessaire et suffisant que la LMI suivante soit faisable :

$$\begin{pmatrix} 1 & F_i Q \\ Q F_i^T & Q \end{pmatrix} \succeq 0, \quad i = 1, \dots, 2n \quad (2.27)$$

Remarquons que si P est une matrice diagonale, alors nous avons :

$$1 - F_i P^{-1} F_i^T = 1 - \frac{(F_i)^2}{p_i}$$

où p_i est le $i^{\text{ième}}$ élément diagonal de P . Ainsi :

$$1 - F_i P^{-1} F_i^T \geq 0, \quad i = 1, \dots, 2n \Leftrightarrow I - F P^{-1} F \geq 0$$

et la condition (2.27) est équivalente à :

$$\begin{pmatrix} \begin{pmatrix} I & F^1 \\ F^1 & P \end{pmatrix} & \mathbb{O} \\ \mathbb{O} & \begin{pmatrix} I & F^2 \\ F^2 & P \end{pmatrix} \end{pmatrix} \succeq 0$$

d'où le corollaire suivant.

Corollaire 2.1 Pour $P = \text{diag}(p_i)$, $\Omega(x) \subseteq \mathbb{E}_x$ si et seulement si la LMI suivante est faisable :

$$\begin{pmatrix} \begin{pmatrix} I & F^1 Q \\ Q F^1 & Q \end{pmatrix} & \mathbb{O} \\ \mathbb{O} & \begin{pmatrix} I & F^2 Q \\ Q F^2 & Q \end{pmatrix} \end{pmatrix} \succeq 0 \quad (2.28)$$

Voyons à présent les nouvelles conditions sur le gain G pour que $\Omega(x) \subseteq \mathbb{U}_v$. En effet, rappelons d'abord qu'avec le retour d'état $v = Gx$, l'ensemble \mathbb{U}_v s'écrit :

$$\mathbb{U}_v = \{x \in \mathbb{R}^n / -v_2 \leq Gx \leq v_1; \quad v_1 > 0; v_2 > 0\} \quad (2.29)$$

ce qui se traduit par :

$$\mathbb{U}_v = \{x \in \mathbb{R}^n / Gx \leq v_1 \text{ et } -Gx \leq v_2; \quad v_1 > 0; v_2 > 0\} \quad (2.30)$$

Soient $\mathbb{V}_1; \mathbb{V}_2$ les matrices diagonales de termes diagonaux v_1^i et v_2^i respectivement. Nous avons alors :

$$\mathbb{V}_1^{-1} = \begin{pmatrix} \frac{1}{v_1^1} & 0 & \dots & 0 \\ 0 & \frac{1}{v_1^2} & 0 & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \frac{1}{v_1^m} \end{pmatrix}, \quad \mathbb{V}_2^{-1} = \begin{pmatrix} \frac{1}{v_2^1} & 0 & \dots & 0 \\ 0 & \frac{1}{v_2^2} & 0 & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \frac{1}{v_2^m} \end{pmatrix}$$

Il s'ensuit que l'ensemble \mathbb{U}_v peut s'écrire sous la forme suivante :

$$\mathbb{U}_v = \{x \in \mathbb{R}^n \mid \begin{pmatrix} \mathbb{V}_1^{-1}G \\ -\mathbb{V}_2^{-1}G \end{pmatrix} x \leq e\} = \{x \in \mathbb{R}^n \mid Hx \leq e\}$$

avec

$$H = \begin{pmatrix} \mathbb{V}_1^{-1}G \\ -\mathbb{V}_2^{-1}G \end{pmatrix}$$

ou d'une manière équivalente :

$$\mathbb{U}_v = \{x \in \mathbb{R}^n \mid H_i x \leq 1, \quad i = 1, \dots, 2m\}$$

où H_i est la $i^{\text{ième}}$ ligne de la matrice H . Donc, en suivant les mêmes arguments que précédemment, l'inclusion $\Omega(x) \subseteq \mathbb{U}_v$ est équivalente à la faisabilité de la LMI suivante :

$$\begin{pmatrix} 1 & H_i \\ H_i^T & P \end{pmatrix} \quad i = 1, \dots, 2m$$

En appliquant le lemme 2.3 de congruence avec :

$$K = \begin{pmatrix} I & \mathbb{O} \\ \mathbb{O} & Q \end{pmatrix}$$

cette LMI est équivalente à la LMI suivante :

$$\begin{pmatrix} 1 & H_i Q \\ Q H_i^T & Q \end{pmatrix} \succeq 0, \quad i = 1, \dots, 2m; \quad Q = P^{-1}$$

D'autre part, puisque $H_i = \frac{1}{v_1^i} G_i$ ou $H_i = -\frac{1}{v_2^i} G_i$, avec G_i est la $i^{\text{ième}}$ ligne de la matrice G , il s'ensuit que $H_i Q = \frac{1}{v_1^i} G_i Q$ et $H_i Q = -\frac{1}{v_2^i} G_i Q$. Or, puisque $Y = GQ$, il vient $Y_i = G_i Q$, où Y_i est la $i^{\text{ième}}$ ligne de la matrice Y . Ainsi, $H_i Q = \frac{1}{v_1^i} Y_i$ ou $H_i Q = -\frac{1}{v_2^i} Y_i$.

Par conséquent, la dernière LMI devient :

$$\begin{pmatrix} 1 & \frac{1}{v_1^i} Y_i \\ \frac{1}{v_1^i} Y_i^T & Q \end{pmatrix} \succeq 0, \quad \begin{pmatrix} 1 & -\frac{1}{v_2^i} Y_i \\ -\frac{1}{v_2^i} Y_i^T & Q \end{pmatrix} \succeq 0, \quad i = 1, \dots, m; \quad Q = P^{-1}; Y = GQ$$

En outre, puisque plusieurs LMI peuvent s'exprimer sous la forme de la LMI simple $\text{diag}(LMI_j) \succeq 0$, nous avons alors le résultat suivant.

Lemme 2.6 *Pour que $\Omega(x) \subseteq \mathbb{U}_v$, il est nécessaire et suffisant que la LMI suivante soit faisable :*

$$\begin{pmatrix} \begin{pmatrix} 1 & \frac{1}{v_1^i} Y_i \\ \frac{1}{v_1^i} Y_i^T & Q \end{pmatrix} & & \mathbb{O} \\ & & \\ & \mathbb{O} & \begin{pmatrix} 1 & -\frac{1}{v_2^i} Y_i \\ -\frac{1}{v_2^i} Y_i^T & Q \end{pmatrix} \end{pmatrix} \succeq 0, \quad i = 1, \dots, m; \quad Q = P^{-1}; Y = GQ \quad (2.31)$$

Finalement, en résumant toutes les conditions citées plus haut, nous avons la solution de notre problème de commande en termes des LMIs.

Théorème 2.1 *Pour que $\Omega(x)$ soit positivement invariant par rapport au système (2.14) et*

$$\Omega(x) \subseteq \mathbb{E}_x \cap \mathbb{U}_v$$

Il est nécessaire et suffisant qu'il existe un gain G et une matrice $P = P^T$ définie positive tels que les LMIs suivantes soient faisables.

$$\begin{aligned} & \begin{pmatrix} Q & Q + Y^T B^T \\ Q + BY & Q \end{pmatrix} \succeq 0, \quad Q = P^{-1}, Y = GQ \\ & \begin{pmatrix} 1 & F_i Q \\ Q F_i^T & Q \end{pmatrix} \succeq 0, \quad i = 1, \dots, 2n, \quad F = \begin{pmatrix} (\mathbb{X}^1)^{-1} \\ -(\mathbb{X}^2)^{-1} \end{pmatrix} \\ & \begin{pmatrix} \begin{pmatrix} 1 & \frac{1}{v_1^i} Y_i \\ \frac{1}{v_1^i} Y_i^T & Q \end{pmatrix} & \mathbb{O} \\ \mathbb{O} & \begin{pmatrix} 1 & -\frac{1}{v_2^i} Y_i \\ -\frac{1}{v_2^i} Y_i^T & Q \end{pmatrix} \end{pmatrix} \succeq 0, \quad i = 1, \dots, m \end{aligned}$$

Si maintenant P est diagonal, alors nous avons le corollaire suivant.

Corollaire 2.2 *Pour P diagonal, $\Omega(x)$ est positivement invariant par rapport au système (2.14) et*

$$\Omega(x) \subseteq \mathbb{E}_x \cap \mathbb{U}_u$$

si et seulement s'il existe un gain G et une matrice $P = P^T$ définie positive tels que les LMIs suivantes soient faisables.

$$\begin{aligned} & \begin{pmatrix} Q & Q + Y^T B^T \\ Q + BY & Q \end{pmatrix} \succeq 0, \quad Q = P^{-1}, Y = GQ \\ & \begin{pmatrix} \begin{pmatrix} I & F^1 Q \\ Q F^1 & Q \end{pmatrix} & \mathbb{O} \\ \mathbb{O} & \begin{pmatrix} I & F^2 Q \\ Q F^2 & Q \end{pmatrix} \end{pmatrix} \succeq 0, \quad F^1 = (\mathbb{X}^1)^{-1}, F^2 = -(\mathbb{X}^2)^{-1} \\ & \begin{pmatrix} \begin{pmatrix} 1 & \frac{1}{v_1^i} Y_i \\ \frac{1}{v_1^i} Y_i^T & Q \end{pmatrix} & \mathbb{O} \\ \mathbb{O} & \begin{pmatrix} 1 & -\frac{1}{v_2^i} Y_i \\ -\frac{1}{v_2^i} Y_i^T & Q \end{pmatrix} \end{pmatrix} \succeq 0, \quad i = 1, \dots, m \end{aligned}$$

Remarque 2.1 John a démontré en 1948 dans [41] que tout convexe compact d'intérieur non vide de \mathbb{R}^n contenait un ellipsoïde de volume maximal. Il est bien connu que le volume d'un ellipsoïde est donné par :

$$vol(\Omega(x)) = vol(Sph) |\det(Q)| = vol(Sph) \prod_{i=1}^n \lambda_i, \quad Q = P^{-1}, \quad (2.32)$$

où $vol(Sph)$ est le volume de la boule unité fermée dans \mathbb{R}^n et λ_i les valeurs propres de la matrice Q . Donc, nous pouvons remarquer que pour chercher le plus grand ellipsoïde $\Omega(x)$ contenu dans $\mathbb{E}_x \cap \mathbb{U}_v$, il faut agrandir son volume. Pour ce faire, nous pouvons agir sur :

1. la plus petite des valeurs propres λ_{min} de la matrice Q . Le problème d'optimisation correspondant s'écrit :

$$\begin{aligned} & \max \lambda_{min} \\ & \text{sous les contraintes } Q \succ 0 \\ & F(Q) \succ 0 \end{aligned}$$

où $F(Q)$ est la contrainte LMI en Q du théorème 2.1.

2. le déterminant de la matrice Q . En effet, puisque :

$$\log(\det(Q)) = \sum_{i=1}^n \log(\lambda_i)$$

alors, le problème d'optimisation correspondant s'écrit :

$$\begin{aligned} & \max \log(\det(Q)) \\ & \text{sous les contraintes } Q \succ 0 \\ & F(Q) \succ 0 \end{aligned}$$

3. la trace $trace(Q) = \sum_{i=1}^n \lambda_i$. Le problème d'optimisation correspondant s'écrit :

$$\begin{aligned} & \max trace(Q) \\ & \text{sous les contraintes } Q \succ 0 \\ & F(Q) \succ 0 \end{aligned}$$

Ces problèmes d'optimisation LMI sont convexes en Q et classiques. Le choix du critère dépend du problème traité. Les avantages respectifs de chaque critère sont mentionnés dans [42]. De plus, tous les problèmes LMI mentionnés ci-dessus peuvent être résolus avec des outils sous MaTlab. ▶

2.4.0.3 Application aux carrefours à feux

Dans cette section, nous exposerons quelques exemples afin de montrer l'efficacité de l'approche proposée. Notons que les solutions satisfaisant les LMIs sont obtenues en utilisant le solveur LMI *SeDuMi* [43] et la boîte à outils Matlab *YALMIP* qui fournit une interface simple pour les solveurs LMI les plus utilisés [44].

2.4.0.4 Cas de deux phases

Les performances de la commande proposée sont évaluées par simulation sur une intersection à deux phases et à cycle constant ayant les paramètres suivants : $s_1 = 0.3$ veh/s ; $s_2 = 0.4$ veh/s ; $q_1 = 0.2$ veh/s ; $q_2 = 0.133$ veh/s ; $c = 120$ s ; $g_{e2max} = 90$ s ; $g_{e2min} = 30$ s ; $y_1^* = 70$ veh ; $y_2^* = 66$ veh. Dans ce cas, la commande nominale a pour valeur $g_{e2N} = u_N = 40$ s.

Une solution satisfaisant les LMIs du théorème 2.1 est donnée par :

$$Q = \begin{pmatrix} 1.1279 & 0 \\ 0 & 1.1382 \end{pmatrix}; Y = \begin{pmatrix} -1.2290 & 1.6386 \end{pmatrix}; G = \begin{pmatrix} -1.0896 & 1.4396 \end{pmatrix}$$

Ainsi

$$u(k) = v(k) + u_N = Gx(k) + u_N = Gy(k) - Gy_N + u_N$$

Les résultats obtenus par l'application de cette approche (figure 2.1) montrent bien que les longueurs des files d'attente y_1, y_2 restent inférieurs au niveau correspondant aux capacités des voies y_1^*, y_2^* respectivement. Remarquons aussi que les courbes de y_1 et y_2 évoluent de manière symétrique jusqu'au cycle 80. Ensuite elles se stabilisent sur les valeurs 30 et 26 respectivement. Ce fait est expliqué par le fait que la commande g_{e2} est stabilisante du système et atteint sa valeur nominale après 80 cycles, c'est-à-dire $g_{e2} = g_{e2N}$ et par conséquent $y = y_N = (31, 26)^T$. Ainsi, cette commande satisfait les contraintes sur les files d'attente et empêche donc la congestion. En outre, les avantages de la commande sont dûs au fait qu'elle est calculée à partir d'une information très riche, à savoir, à chaque instant, le contrôleur évalue la durée des feux verts suivant les longueurs des files d'attente au sein de l'intersection pour chaque cycle k . Cette situation nous permet d'utiliser efficacement les informations disponibles sur le système afin d'élaborer une commande de feux en temps réel. Ceci lui confère une légitimité par rapport aux objectifs fixés pour notre investigation.

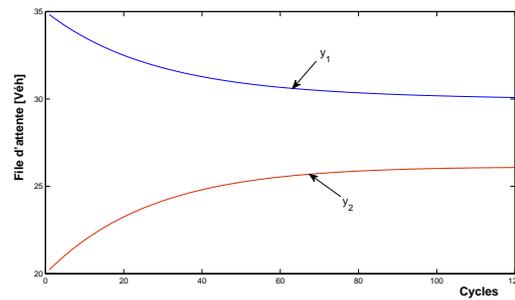


FIGURE 2.1 – Evolution des files d’attente.

Dans le tableau 2.1 suivant, nous reportons les valeurs du vert effectif g_{e_2} correspondants aux 120 cycles. Remarquons que g_{e_2} augmente entre 38.43 s et 57.59 s tout en respectant les conditions aux limites $30 \leq g_{e_i} \leq 90$.

cycles	1	2	3	4	5	6	7	8	9	10
g_{e_2}	38.43	45.79	40.89	39.12	42.43	45.89	51.74	47.52	53.14	50.29
cycles	11	12	13	14	15	16	17	18	19	20
g_{e_2}	48.46	48.15	46.90	49.11	45.73	46.18	45.97	44.30	46.52	43.94
cycles	21	22	23	24	25	26	27	28	29	30
g_{e_2}	45.80	45.68	43.73	46.35	45.41	47.02	47.94	45.75	44.62	46.38
cycles	31	32	33	34	35	36	37	38	39	40
g_{e_2}	49.51	51.52	51.68	50.64	52.65	50.39	50.42	53.04	54.32	52.61
cycles	41	42	43	44	45	46	47	48	49	50
g_{e_2}	54.05	57.11	54.72	55.47	51.81	51.34	53.39	51.70	49.14	52.94
cycles	51	52	53	54	55	56	57	58	59	60
g_{e_2}	55.03	54.66	54.36	55.94	52.26	50.94	51.97	49.64	50.13	50.97
cycles	61	62	63	64	65	66	67	68	69	70
g_{e_2}	54.65	57.59	55.96	52.64	56.40	52.71	51.14	51.02	52.34	51.97
cycles	71	72	73	74	75	76	77	78	79	80
g_{e_2}	54.42	51.35	48.66	49.10	50.74	47.64	45.70	45.36	44.16	41.12
cycles	81	82	83	84	85	86	87	88	89	90
g_{e_2}	40.04	40	40	40	40	40	40	40	40	40
cycles	91	92	93	94	95	96	97	98	99	100
g_{e_2}	40	40	40	40	40	40	40	40	40	40
cycles	101	102	103	104	105	106	107	108	109	110
g_{e_2}	40	40	40	40	40	40	40	40	40	40
cycles	111	112	113	114	115	116	117	118	119	120
g_{e_2}	40	40	40	40	40	40	40	40	40	40

TABLE 2.1 – Variation du feu vert effectif g_{e_2} en fonction du cycle.

2.4.0.5 Cas de trois phases

Dans cet exemple, nous appliquons notre approche à une intersection à trois phases. Les données de la simulation sont les suivantes : $s_1 = 0.48$ veh/s ; $s_2 = 0.35$ veh/s ; $s_3 = 0.55$ veh/s ; $q_1 = 0.16$ veh/s ; $q_2 = 0.11$ veh/s ; $q_3 = 0.18$ veh/s ; $c = 120$ s ; $g_{e1max} = 50$ s ; $g_{e1min} = 35$ s ; $g_{e2max} = 50$ s ; $g_{e2min} = 35$ s ; $y_1^* = 70$ veh ; $y_2^* = 50$ veh ; $y_3^* = 60$ veh ; $y_N = (42, 14, 39)^T$. Dans ce cas, nous avons :

$$B = \begin{pmatrix} -0.48 & 0 \\ 0 & -0.35 \\ 0.55 & 0.55 \end{pmatrix} ; L = \begin{pmatrix} 19.20 \\ 14.35 \\ -44.85 \end{pmatrix}$$

Le vecteur de commande nominale a pour valeur $u_N = (40, 41)$. En résolvant les LMIs du théorème 2.1, nous obtenons :

$$Q = \begin{pmatrix} 1.0128 & 0 & 0 \\ 0 & 1.0983 & 0 \\ 0 & 0 & 1.0734 \end{pmatrix} ; Y = \begin{pmatrix} 0.8323 & -0.3108 & -0.4684 \\ -0.2527 & 0.7744 & -0.9337 \end{pmatrix}$$

$$G = \begin{pmatrix} 0.8218 & -0.2830 & -0.4364 \\ -0.2496 & 0.7051 & -0.8699 \end{pmatrix}$$

Les résultats de la simulation sont donnés dans la figure 2.2. Nous pouvons immédiatement observer dans ce cas aussi que la commande $u(k)$ empêche les longueurs des files d'attente d'atteindre les capacités des voies et évite donc la congestion.

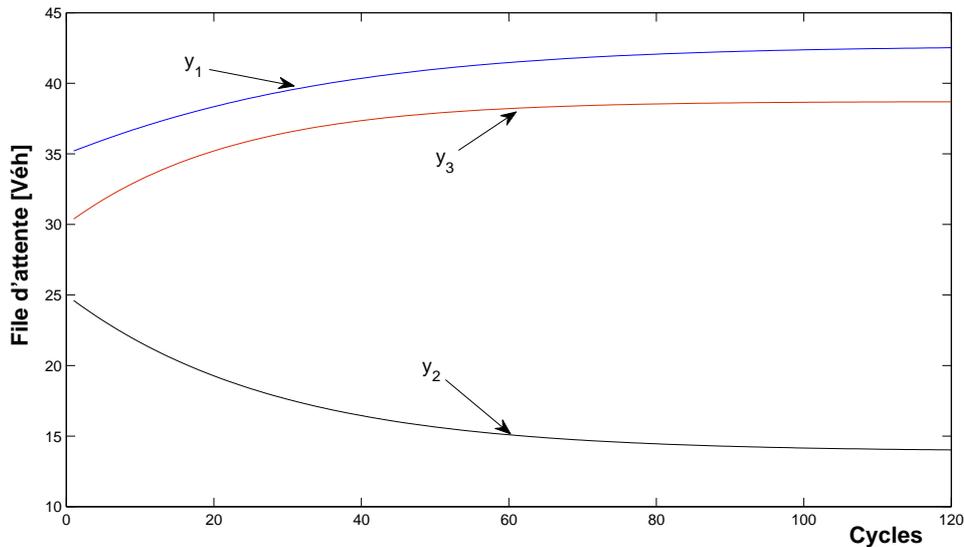


FIGURE 2.2 – Evolution des files d'attente.

En outre, le tableau 2.2 illustre les variations correspondantes des feux verts effectifs g_{e_1} et g_{e_2} pendant les premiers 20 cycles. Nous pouvons constater dans ce cas aussi que les valeurs de g_{e_1} et g_{e_2} varient légèrement tout en respectant les contraintes.

cycles	1	2	3	4	5	6	7	8	9	10
g_{e_1}	44.48	47.31	47.91	45.59	41.40	45.72	47.63	45.11	42.31	45.46
g_{e_2}	47.94	47.82	44.05	42.95	45.27	40.33	41.73	40.90	37.41	40.46
cycles	11	12	13	14	15	16	17	18	19	20
g_{e_1}	41.70	39.24	42.31	42.05	45.74	43.97	43.36	41.69	44.21	43.23
g_{e_2}	41.44	44.05	44.86	43.80	43.12	42.15	41.62	42.91	44.75	44.13

TABLE 2.2 – Variation des feux verts effectifs (g_{e_1}, g_{e_2}) en fonction du cycle.

En résumé, la stratégie de commande proposée dans cette section permet aux durées des feux de varier d'un cycle à un autre en fonction de la demande. Cependant, pour que la stratégie de commande s'adapte de manière efficace aux variations permanentes du trafic, il est nécessaire que le contrôle ait un degré important d'adaptabilité pour répondre dans des très courts délais à une demande très fluctuante et aléatoire. A cette fin, la durée du cycle constitue un degré de liberté dont on dispose pour cet objectif.

Dans la section suivante, nous abordons le cas où la durée du cycle est variable dans le temps et est considérée comme une variable de commande. La durée de cycle ainsi que les durées des feux peuvent varier d'un cycle à un autre en fonction de la demande enregistrée en amont.

2.4.1 Cas où le cycle est variable

Dans le cas où le cycle est variable, le modèle d'état du carrefour est de la forme (voir (1.10) chapitre 1) :

$$y(k+1) = y(k) + Bu(k) \quad (2.33)$$

où $y \in \mathbb{R}^n$ et $u \in \mathbb{R}^n$ avec :

$$\mathbb{E}_y = \{y \in \mathbb{R}_+^n / 0 \leq y(k) \leq y^*, \quad \forall k \geq 0, \quad \forall 0 \leq y_0 \leq y^*\} \quad (2.34)$$

$$\mathbb{U}_u = \{u \in \mathbb{R}_+^n / u_{min} \leq u \leq u_{max}; \quad u_{min} > 0\} \quad (2.35)$$

Comme dans la section précédente, l'objectif de la commande est alors de trouver un retour d'état et une matrice définie positive $P = P^T$ tels que $\Omega(y)$ soit positivement invariant par rapport au système 2.33 et $\Omega(y) \subseteq \mathbb{E}_y \cap \mathbb{U}_u$

Remarque 2.2 Ce problème de contrôle est le même que celui décrit dans les sections précédentes sauf que les variables de contrôle dans ce cas sont le temps effectif vert $g_e(k)$

et la durée du cycle $c(k)$. Ceci suggérerait que les résultats obtenus précédemment restent valables pour le cas où le cycle des feux est variable. Or, il n'en est rien, car dans cette nouvelle formulation, la matrice B est une matrice carrée ($B \in \mathbb{R}^{n \times n}$) et l'existence de la commande nominale u_N dépend fortement du rang de la matrice B . C'est la raison pour laquelle nous allons séparer l'étude, dans la suite du chapitre, en fonction du $\text{rang}(B)$. ►

2.4.1.1 Cas où $\text{rang}(B) < n$

Dans ce cas, nous pouvons supposer l'existence d'une commande $u_N \in \mathbb{U}_u$ (situation d'équilibre y_N du système) telle que :

$$Bu_N = 0 \quad (2.36)$$

Notons que la solvabilité de (2.36) avec $u \in \mathbb{R}^n$ est bien connue dans le cours de l'algèbre linéaire par l'utilisation de la méthode de Cramer ou la décomposition en valeurs singulières. Alors, le sérieux problème que pose la solvabilité de (2.36) est de prouver l'existence d'un vecteur positif et, le cas échéant, donner une forme explicite de la solution. Ce problème a été résolu dans l'annexe A.

Posons maintenant comme nouveau vecteur de commande :

$$v(k) = u(k) - u_N \quad (2.37)$$

et comme nouveau vecteur d'état :

$$x = y - y_N$$

Les équations (2.33) et (2.37) donnent alors le système suivant :

$$x(k+1) = x(k) + Bv(k); \quad x \in \mathbb{R}^n; v \in \mathbb{R}^n \quad (2.38)$$

Comme dans le cas du cycle constant, si nous posons :

$$v_2 = u_N - u_{\min}; \quad v_1 = u_{\max} - u_N$$

et :

$$x_2 = y_N; \quad x_1 = y^* - y_N$$

les contraintes sur $u(k)$ se transportent sur $v(k)$ en :

$$\mathbb{U}_v = \{v \in \mathbb{R}^n / -v_2 \leq v \leq v_1; \quad v_1, v_2 > 0\} \quad (2.39)$$

et l'ensemble \mathbb{E}_y devient :

$$\mathbb{E}_x = \{x \in \mathbb{R}^n / -x_2 \leq x \leq x_1; \quad x_1, x_2 > 0\} \quad (2.40)$$

La commande du système dynamique (2.38) aura toujours pour expression :

$$v(k) = Gx(k) \quad (2.41)$$

où $G \in \mathbb{R}^{n \times n}$. Dans ce cas, le système (2.38) en boucle fermée est donné par :

$$x(k+1) = (I + BG)x(k) \quad (2.42)$$

Dans cette situation, nous nous ramenons au cas où le cycle est constant (voir section 2.3.1).

2.4.1.2 Cas où $\text{rang}(B) = n$

Si $\text{rang}(B) = n$, alors la matrice B est non-singulière. Par conséquent, aucune commande nominale $u_N \neq 0$ solution de $Bu_N = 0$ ne peut exister. Pour résoudre ce problème, nous allons prendre le changement de variable suivant :

$$v = u - u_{min} \quad (2.43)$$

Dans ce cas, les contraintes sur $u(k)$ se transportent sur $v(k)$ en :

$$\mathbb{U}_v = \{v \in \mathbb{R}^n / 0 \leq v \leq q; \quad q > 0\} \quad (2.44)$$

avec $q = u_{max} - u_{min}$. L'ensemble \mathbb{E}_y demeure inchangé :

$$\mathbb{E}_y = \{y \in \mathbb{R}^n / 0 < y \leq y^*; \quad y^* > 0\}$$

Dans ces conditions, puisque \mathbb{E}_y et \mathbb{U}_v ont des frontières qui passent par 0, le domaine $\Omega(y)$ sera restreint à l'orthant positif :

$$\Omega(y) = \{y \in \mathbb{R}_{+*}^n \setminus y^T P y \leq 1\} \quad (2.45)$$

Ceci fait, avec le changement de variable (2.43), le système (2.33) aura pour équation :

$$y(k+1) = y(k) + Bv(k) + Bu_{min} \quad (2.46)$$

Choisissons maintenant une matrice G telle que la matrice $(I + BG)^T P (I + BG) - P$ soit semi-définie négative. Ceci est équivalent à la résolution de la LMI suivante (voir section 2.4.0.1) :

$$\begin{pmatrix} Q & Q + Y^T B^T \\ Q + BY & Q \end{pmatrix} \succeq 0, \quad Q = P^{-1}, \quad Y = GQ$$

Considérons alors le retour d'état suivant :

$$v = Gy + \alpha \quad (2.47)$$

où α est une commande auxiliaire. Le système (2.46) en boucle fermée devient :

$$y(k+1) = Ay(k) + Bw \quad (2.48)$$

où

$$A = (I + BG); \quad w = \alpha + u_{min}$$

Ceci fait, nous allons à présent voir comment choisir w (et par conséquent α) pour que $\Omega(y)$ soit positivement invariant par rapport au système 2.48. En effet, d'après le lemme (2.1), l'invariance positive de $\Omega(y)$ est équivalente à :

$$\Delta \mathcal{V}(y) = \mathcal{V}(y(k+1)) - \mathcal{V}(y) \leq 0$$

Ceci implique :

$$\Delta \mathcal{V}(y) = y^T [A^T P A - P] y + f(w) \leq 0$$

où

$$f(w) = 2y^T A^T P B^T w + w^T B P B^T w \quad (2.49)$$

Or, puisque $A^T P A - P$ est semi-définie négative, l'invariance positive de $\Omega(y)$ est assurée si w est choisi tel que $f(w) \leq 0$. En effet, remarquons que puisque $\text{rang}(B) = n$, alors la matrice $B P B^T$ est inversible. Soit θ le vecteur défini par :

$$\theta = -2(B P B^T)^{-1} B P A y \quad (2.50)$$

ce qui donne :

$$2B P A y = -(B P B^T) \theta$$

Introduisons maintenant ce vecteur dans (2.49), il vient :

$$f(w) = -2\theta^T (B P B^T) w + w^T B P B^T w + \theta^T B P B^T \theta - \theta^T B P B^T \theta$$

ce qui implique :

$$f(w) = (w - \theta)^T (B P B^T) (w - \theta) - \theta^T B P B^T \theta$$

Ainsi, si $w = \theta$ nous avons :

$$f(w) = -\theta^T B P B^T \theta \leq 0 \quad (2.51)$$

car la matrice $B P B^T$ est définie positive. Ainsi, l'invariance positive de $\Omega(y)$ est assurée si $w = -2(B P B^T)^{-1} B P A y$, et par conséquent :

$$\alpha = w - u_{min} = -2(B P B^T)^{-1} B P A y - u_{min} = -2(B^T)^{-1} A y - u_{min}$$

Dans ce cas, nous avons :

$$v = [G - 2(B^T)^{-1}A]y - u_{min}$$

La proposition suivante résume ces résultats.

Proposition 2.1 *Si la matrice G est choisie telle que la matrice $A^T P A - P$ soit semi-définie négative, alors le retour d'état :*

$$v = [G - 2(B^T)^{-1}A]y - u_{min} = Hy - u_{min} \quad (2.52)$$

garantit l'invariance positive de $\Omega(y)$ par rapport au système (2.48).

Regardons maintenant les nouvelles conditions sur l'inclusion $\Omega(y) \subseteq \mathbb{E}_y \cap \mathbb{U}_v$. Commençons d'abord par $\Omega(y) \subseteq \mathbb{E}_y$. En effet, comme dans la section 2.3.1, désignons par \mathbb{Y} la matrice diagonale de termes diagonaux y_i^* . Ainsi nous avons :

$$\mathbb{Y}^{-1} = \begin{pmatrix} \frac{1}{y_1^*} & 0 & \cdots & 0 \\ 0 & \frac{1}{y_2^*} & 0 & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{1}{y_n^*} \end{pmatrix}$$

Il vient alors :

$$y \leq y^* \Leftrightarrow \mathbb{Y}^{-1}y \leq e$$

où $e = (1, 1, \dots, 1)^T$ est le vecteur dont toutes ses composantes sont égales à 1. Il s'ensuit que l'ensemble \mathbb{E}_y peut s'écrire sous la forme suivante :

$$\mathbb{E}_y = \{y \in \mathbb{R}_+^n \mid \mathbb{Y}^{-1}y \leq e\} = \{y \in \mathbb{R}_+^n \mid Fy \leq e\}$$

où $F = \mathbb{Y}^{-1}$. L'ensemble \mathbb{E}_y peut aussi s'écrire sous la forme :

$$\mathbb{E}_y = \{y \in \mathbb{R}_+^n \mid F_i y \leq 1, \quad i = 1, \dots, n\} \quad (2.53)$$

où F_i est la $i^{\text{ième}}$ ligne de la matrice F . Ceci fait, comme dans la section 2.3.1, la condition $\Omega(y) \subseteq \mathbb{E}_y$ est équivalente à :

$$\max_{y \in \Omega(y)} \{F_i y\} \leq 1 \quad (2.54)$$

Il s'agit maintenant de connaître la valeur maximale de la fonction linéaire $F_i y$ sur l'ensemble $\Omega(y)$, c'est-à-dire :

$$\max F_i y$$

$$\text{sous la contrainte } y^T P y \leq 1, \quad -y < 0$$

Pour résoudre ce problème, nous allons appliquer les conditions de Kuhn et Tucker (voir section 2.2.3). En effet, il existe $n + 1$ contraintes h_j avec $h_1 = y^T P y$ et $h_j = -y_j$, $j = 2, \dots, n + 1$. Soit maintenant le Lagrangien L :

$$L(y, \lambda, \mu) = F_i y - \lambda(y^T P y - 1) + \mu^T y$$

Alors, si \bar{y} est la solution recherchée, il existe un scalaire $\bar{\lambda} \geq 0$ et un vecteur $\bar{\mu} \geq 0$ tels que :

$$\frac{\partial L}{\partial y} = F_i^T - 2\bar{\lambda} P \bar{y} + \bar{\mu} = 0 \quad (2.55)$$

$$\bar{\lambda}(y^T P y - 1) = 0 \Leftrightarrow \bar{\lambda} = 0, \text{ ou } y^T P y - 1 = 0 \quad (2.56)$$

$$\bar{\mu}^T \bar{y} = 0 \Leftrightarrow \bar{\mu} = 0, \text{ ou } \bar{y} = 0 \quad (2.57)$$

$$\frac{\partial L}{\partial \lambda} \geq 0 \Leftrightarrow \bar{y}^T P \bar{y} \leq 1 \quad (2.58)$$

$$\frac{\partial L}{\partial \mu} \geq 0 \Leftrightarrow \bar{y} \geq 0 \quad (2.59)$$

Pour déterminer les solutions de ce système, il faut envisager successivement tous les cas de figure possibles portant sur la saturation des contraintes et procéder par élimination.

Cas 1 : $\bar{y}^T P \bar{y} = 1$ et $\bar{y} = 0$ (les $n + 1$ contraintes sont saturées à l'optimum). Dans ce cas, on a $0 = 1$ ce qui est absurde. Donc, on ne peut pas avoir les $n + 1$ contraintes saturées à l'optimum simultanément.

Cas 2 : $\bar{y}^T P \bar{y} = 1$ et $\bar{y} > 0$ (seule la première contrainte est saturée à l'optimum). Dans ce cas, on a :

$$\text{rang}\left(\frac{\partial \bar{y}^T P \bar{y}}{\partial \bar{y}}\right) = \text{rang}(2P\bar{y}) = 1, \quad \forall \bar{y} \in \mathbb{R}^n$$

Ainsi, la contrainte de qualification est vérifiée. Il s'ensuit alors que dans ce cas nous avons $\bar{\lambda} > 0$ et $\bar{\mu} = 0$. Ainsi, en suivant les mêmes calculs de la section 2.3.1 nous obtenons :

$$\bar{y} = (F_i P^{-1} F_i^T)^{-\frac{1}{2}} P^{-1} F_i^T$$

qui est une solution possible du problème de maximisation de la fonction $F_i y$ sur l'ensemble $\Omega(y)$.

Cas 3 : $\bar{y}^T P \bar{y} < 1$ et $\bar{y} = 0$ (seules n contraintes sont saturées à l'optimum). La contrainte de qualification est vérifiée dans ce cas puisque :

$$\text{rang}\left(\frac{\partial y}{\partial y}\right) = \text{rang}(I) = n$$

Dans cette situation, on a $\bar{\lambda} = 0$ et $\bar{\mu} > 0$. D'où l'on peut inférer à partir de (2.55) que $F_i^T = -\bar{\mu} < 0$ ce qui viole le fait que $F_i^T \geq 0$.

Cas 4 : $\bar{y}^T P \bar{y} < 1$ et $\bar{y} > 0$ (aucune des $n + 1$ contraintes n'est saturée à l'optimum). Dans ce cas, on a $\bar{\lambda} = 0$ et $\bar{\mu} = 0$. Ainsi, l'équation (2.55) donne $F_i^T \bar{y} = 0$ et contredit le fait que $F_i \bar{y} \neq 0$.

Finalement, on en déduit que les conditions de Kuhn et Tucker admettent une seule solution :

$$\bar{y} = (F_i P^{-1} F_i^T)^{-\frac{1}{2}} P^{-1} F_i^T$$

Or, $F_i y$ est concave et les contraintes sont convexes, donc \bar{y} est un maximum global. Par conséquent, la valeur maximale de $F_i y$ sur l'ensemble $\Omega(y)$ est donnée par :

$$\max_{y \in \Omega(y)} \{F_i y\} = (F_i P^{-1} F_i^T)^{\frac{1}{2}}$$

Ainsi, d'après (2.54), la condition $\Omega(y) \subseteq \mathbb{E}_y$ est équivalente à :

$$(F_i P^{-1} F_i^T)^{\frac{1}{2}} \leq 1$$

qui à son tour équivalente à :

$$F_i P^{-1} F_i^T \leq 1$$

En suivant la même démarche de la section 2.3.1, cette inégalité est équivalente à la LMI suivante :

$$\begin{pmatrix} 1 & F_i Q \\ Q F_i^T & Q \end{pmatrix} \succeq 0, \quad Q = P^{-1}$$

En ce qui concerne l'inclusion $\Omega(y) \subseteq \mathbb{U}_v$, remarquons qu'avec le retour d'état (2.52), \mathbb{U}_v s'écrit sous la forme :

$$\mathbb{U}_v = \{y \in \mathbb{R}_{+*}^n / u_{min} \leq H y \leq u_{max}; \quad u_{min} > 0\}$$

Il est clair que $\Omega(y) \subseteq \mathbb{U}_v$ est équivalente à :

$$\min_{y \in \partial \mathbb{U}_v} \{y^T P y\} \geq 1 \tag{2.60}$$

où $\partial \mathbb{U}_v$ est la frontière de \mathbb{U}_v . Dans [45], il a été démontré que :

$$\min_{y \in \partial \mathbb{U}_v} \{y^T P y\} = \min_i \left\{ \frac{u_{min}^2}{H_i P^{-1} H_i^T}; \frac{u_{max}^2}{H_i P^{-1} H_i^T}, \quad i = 1, \dots, n \right\}$$

où H_i est la $i^{\text{ème}}$ ligne de la matrice H . Or, $u_{min} < u_{max}$, il s'ensuit que :

$$\min_{y \in \partial \mathbb{U}_v} \{y^T P y\} = \frac{u_{min}^2}{H_i P^{-1} H_i^T}, \quad i = 1, \dots, n$$

D'après (2.60), la condition d'inclusion $\Omega(y) \subseteq \mathbb{U}_v$ est équivalente à :

$$\frac{u_{min}^2}{H_i P^{-1} H_i^T} \geq 1, \quad i = 1, \dots, n$$

où d'une manière équivalente :

$$u_{min}^2 - H_i P^{-1} H_i^T \geq 0, \quad i = 1, \dots, n$$

En suivant la même démarche de la section 2.3.1, cette inégalité est équivalente à la LMI suivante :

$$\begin{pmatrix} u_{min}^2 & H_i Q \\ Q H_i^T & Q \end{pmatrix} \succeq 0$$

Finalement, en résumant tous les résultats cités plus haut, nous avons le théorème suivant :

Théorème 2.2 *Pour que $\Omega(y)$ soit positivement invariant par rapport au système (2.48) il suffit que la LMI suivante soit faisable :*

$$\begin{pmatrix} Q & Q + Y^T B^T \\ Q + BY & Q \end{pmatrix} \succeq 0, \quad Q = P^{-1}, Y = GQ \quad (2.61)$$

Dans ces conditions, avec le retour d'état $v = Hy - u_{min}$, l'inclusion $\Omega(y) \subseteq \mathbb{E}_y \cap \mathbb{U}_v$ est satisfaite si et seulement si les LMI suivantes sont faisables :

$$\begin{pmatrix} 1 & F_i Q \\ Q F_i^T & Q \end{pmatrix} \succeq 0, \quad i = 1, \dots, n \quad (2.62)$$

$$\begin{pmatrix} u_{min}^2 & H_i Q \\ Q H_i^T & Q \end{pmatrix} \succeq 0, \quad i = 1, \dots, n \quad (2.63)$$

Remarquons tout de suite que le gain G et la matrice P sont calculés à partir de la LMI (2.61). Par conséquent, les LMIs (2.62) et (2.63) sont imposées uniquement pour la vérification de l'inclusion $\Omega(y) \subseteq \mathbb{E}_y \cap \mathbb{U}_v$.

2.4.1.3 Application aux carrefours à feux

Cas de deux phases

La figure 2.3 représente l'évolution des files d'attente dans un carrefour à deux phases avec un cycle variable. La simulation est effectuée en utilisant les données suivantes : $s_1 = 0.6$ veh/s ; $s_2 = 0.4$ veh/s ; $q_1 = 0.3$ veh/s ; $q_2 = 0.2$ veh/s ; $c_{max} = 120$ s ; $c_{min} = 90$ s ; $g_{e_{2max}} = 70$ s ; $g_{e_{2min}} = 30$ s ; $y_1^* = 50$ veh ; $y_2^* = 40$ veh.

Tout d'abord il est important de calculer le rang de la matrice B :

$$B = \begin{pmatrix} q_1 - s_1 & s_1 \\ q_2 & -s_2 \end{pmatrix} = \begin{pmatrix} -0.3 & 0.6 \\ 0.2 & -0.4 \end{pmatrix}$$

Remarquons que $\text{rang}(B) = 1 < n = 2$. Ainsi, $u_N = (110, 55)^T$. Donc, d'après le théorème 2.1, il en résulte :

$$Q = \begin{pmatrix} 1.0612 & 0 \\ 0 & 1.1215 \end{pmatrix}; Y = \begin{pmatrix} 0.2741 & -0.1882 \\ -1.1853 & 0.7875 \end{pmatrix}; G = \begin{pmatrix} 0.2583 & -0.1678 \\ -1.1170 & 0.7022 \end{pmatrix}$$

Les résultats de la simulation montrent dans ce cas aussi que la commande atteint les objectifs souhaités.

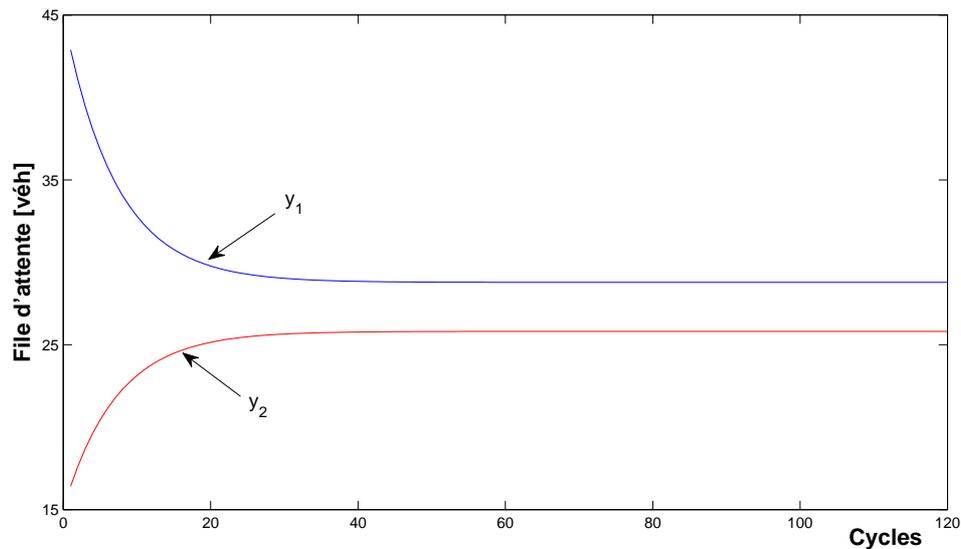


FIGURE 2.3 – Evolution des files d'attente.

Le tableau 2.3 présente une vue d'évolution de la durée du cycle $c(k)$ ainsi que le vert effectif g_{e_2} pendant les premiers 20 cycles.

cycles	1	2	3	4	5	6	7	8	9	10
$c(k)$	102.71	98.83	103.69	108.41	113.28	102.36	106.74	105.47	112.09	105.19
g_{e_2}	45.47	49.20	49.33	52.31	56.84	57.36	59.06	59.24	58.61	55.54
cycles	11	12	13	14	15	16	17	18	19	20
$c(k)$	101.36	98.97	99.82	98.68	102.57	100.10	97.85	106.21	104.39	104.91
g_{e_2}	56.20	49.61	55.94	42.15	50.73	52.74	59.90	54.81	50.76	52.03

TABLE 2.3 – Variation de la durée de cycle $c(k)$ et du feu vert effectif g_{e_2} en fonction du cycle.

Cas de trois phases

Dans cet exemple, nous réalisons une simulation pour un carrefour à trois phases et à cycle variable. Les données utilisées dans cet exemple sont définies comme suit :

$s_1 = 0.5 \text{ veh/s}$; $s_2 = 0.5 \text{ veh/s}$; $s_3 = 0.5 \text{ veh/s}$; $q_1 = 0.2 \text{ veh/s}$; $q_2 = 0.15 \text{ veh/s}$; $q_3 = 0.15 \text{ veh/s}$; $c_{max} = 140 \text{ s}$; $c_{min} = 100 \text{ s}$; $g_{e1_{max}} = 50 \text{ s}$; $g_{e1_{min}} = 30 \text{ s}$; $g_{e2_{max}} = 45 \text{ s}$; $g_{e2_{min}} = 30 \text{ s}$; $y_1^* = 60 \text{ veh}$; $y_2^* = 50 \text{ veh}$; $y_3^* = 55 \text{ veh}$; $y_N = (25, 15, 15)^T$.

Ainsi,

$$B = \begin{pmatrix} q_1 & -s_1 & 0 \\ q_2 & 0 & -s_2 \\ q_3 - s_3 & s_3 & s_3 \end{pmatrix} = \begin{pmatrix} 0.2 & -0.5 & 0 \\ 0.15 & 0 & -0.5 \\ -0.35 & 0.5 & 0.5 \end{pmatrix}$$

et $\text{rang}(B) = 2 < n$. Dans ce cas, nous avons $u_N = (115.50, 46.20, 35)^T$.

La résolution des conditions du théorème 2.1, donne immédiatement :

$$Q = \begin{pmatrix} 0.9204 & 0 & 0 \\ 0 & 0.9179 & 0 \\ 0 & 0 & 1.0775 \end{pmatrix}; Y = \begin{pmatrix} -1.6072 & -0.1468 & 1.3738 \\ 0.3769 & -0.4741 & -0.0550 \\ -0.7847 & 0.9945 & -0.3239 \end{pmatrix}$$

$$G = \begin{pmatrix} -1.7462 & -0.1599 & 1.2750 \\ 0.4095 & -0.5164 & -0.0510 \\ -0.8526 & 1.0834 & -0.3006 \end{pmatrix}$$

Nous constatons que les files d'attente respectent les contraintes comme le montre la figure 2.4.

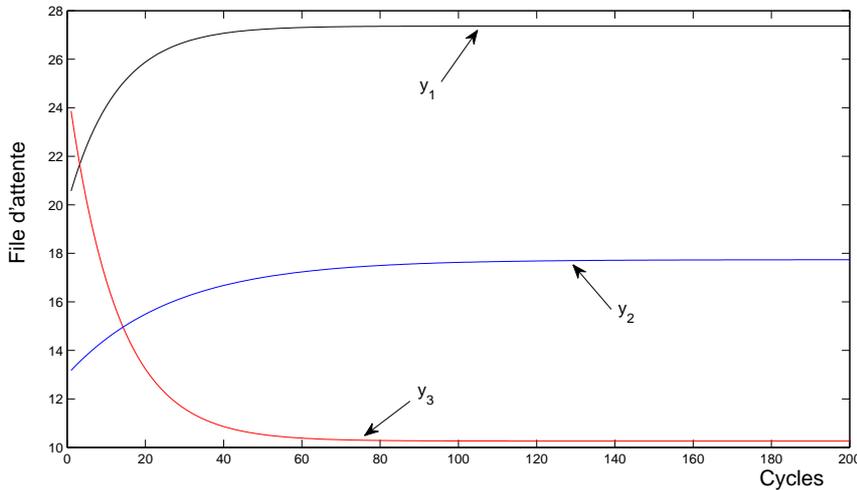


FIGURE 2.4 – Evolution des files d'attente.

Les valeurs des paramètres de commande pendant les premiers 20 cycles sont indiquées dans le tableau 2.4. Nous remarquons que la durée du cycle varie entre 115.99s et 133.21s tandis que g_{e1} , g_{e2} , quant à eux, varient dans les plages $[37.35, 48.92]$ et $[36.12, 45]$ respectivement.

cycles	1	2	3	4	5	6	7	8	9	10
$c(k)$	131.94	133.21	133.05	129.84	126.71	129.25	115.99	122.73	127.06	130.74
g_{e_1}	47.60	47.11	48.92	46.41	42.84	45.61	41.36	42.10	39.71	39.04
g_{e_2}	44.52	44.09	42.75	39.80	41.90	37.15	38.72	40.50	43.07	41.10
cycles	11	12	13	14	15	16	17	18	19	20
$c(k)$	130.55	129.80	126.14	122.57	125.71	121.35	126.60	128.14	132.90	129.64
g_{e_1}	41.18	38.96	38.09	42.25	43.80	40.03	37.35	39.21	39.66	42.91
g_{e_2}	43.80	45	41.72	38.68	36.12	39.56	42.15	41.50	39.27	39.10

TABLE 2.4 – Variation de la durée de cycle $c(k)$ et des feux verts effectifs g_{e_1} , g_{e_2} en fonction du cycle.

2.5 Conclusion

Dans ce chapitre, une approche basée sur les LMIs a été appliquée pour éviter la congestion du trafic urbain par les feux de signalisation en forçant les files d'attente à ne pas dépasser le niveau de trafic correspondant à l'optimum opérationnel des lignes. Cette étude a été faite grâce à la construction d'un domaine ellipsoïdal Ω positivement invariant, contenu dans le domaine des contraintes sur l'état et la commande. Deux cas ont été étudiés : celui où le cycle des feux est constant était d'abord analysé. Ensuite, nous nous sommes intéressés au cas où le cycle des feux est variable afin d'adapter la stratégie de commande de manière efficace aux variations permanentes du trafic. Dans les deux cas, la mise en évidence de l'approche LMI nous a permis d'énoncer des conditions nécessaires et suffisantes pour la résolution de notre problème de commande.

Cependant, il est à noter que puisque le domaine Ω est conditionné par la forme quadratique $x^T P x$, il ne permet pas d'étudier directement et sans passer par des changements de variables, des cas limites difficiles comme ceux des contraintes ayant des frontières qui passent par l'origine (comme c'est le cas des carrefours signalisés). En outre, dans le cas où les contraintes sont très dissymétriques, le domaine Ω sera très insignifiant autour de l'origine.

Pour remédier à ces inconvénients, nous proposons dans le chapitre suivant une autre approche permettant de considérer Ω comme un ensemble polyédral et de prendre non plus une partie de \mathbb{E}_x mais \mathbb{E}_x tout entier, c'est-à-dire $\Omega = \mathbb{E}_x$.

CHAPITRE 3

APPROCHE (A, B) -INVARIANCE

3.1 Introduction

Ce chapitre utilise la propriété d' (A, B) -invariance des ensembles \mathbb{E}_x et \mathbb{U}_v pour la régulation a priori des carrefours signalisés. La vérification de cette propriété permet de conclure sur la possibilité de restreindre l'état d'un système à l'intérieur d'un ensemble donné, au moyen d'une commande appropriée sans supposer aucune structure particulière des contraintes.

Comme il est souligné dans la conclusion du chapitre 2 sur les inconvénients de l'approche LMI, notre souci premier est de considérer le domaine des contraintes sur l'état comme un domaine candidat à l'invariance en boucle fermée au lieu de construire un domaine Ω candidat. L'avantage de cette approche est d'élargir l'ensemble des trajectoires se trouvant dans le champ des contraintes. Un autre avantage se fait sentir notamment dans le calcul de la commande, pour lequel nous proposons une méthode numérique, sans qu'elle soit conditionnée par la matrice P .

Cette méthode permet en outre d'intégrer directement les contraintes sur la commande dans la synthèse du régulateur. En effet, il suffit d'imposer à la fois en boucle fermée l'invariance du domaine \mathbb{E}_x et son inclusion dans le domaine des contraintes sur la commande \mathbb{U}_v pour assurer le respect des contraintes par les trajectoires du système.

Dans un premier temps, nous rappelons quelques définitions et résultats utiles dont nous aurons besoin dans la suite de l'exposé. Ensuite, nous précisons la définition du problème de commande qui nous concerne. Une fois le problème est bien défini, nous donnons sous forme de relations matricielles linéaires des conditions nécessaires et suffisantes

pour que l'ensemble des contraintes sur l'état soit à la fois un ensemble positivement invariant de notre système et inclus dans l'ensemble des contraintes sur la commande. Ces résultats sont ensuite appliqués aux modèles de transport établis dans le chapitre 1 pour le cas où le cycle des feux est constant ou variable.

3.2 Rappels mathématiques et résultats préliminaires

L'espace vectoriel \mathbb{R}^n muni de la relation d'ordre partiel \geq est un espace de Riesz [46], c'est-à-dire pour tout couple (x, y) d'éléments de \mathbb{R}^n ,

- $\sup(x, y)$ existe (le plus petit des majorants de $\{x, y\}$)
- $\inf(x, y)$ existe (le plus grand des minorants de $\{x, y\}$)

Définissons maintenant pour tout $x \in \mathbb{R}^n$:

$$x^+ = \sup(x, 0); \quad x^- = \sup(-x, 0) = (-x)^+ = -\inf(x, 0)$$

Il est clair que pour tout $x \in \mathbb{R}^n$, $x^+ > 0$ et $x^- > 0$ et :

$$x = x^+ - x^-; \quad |x| = x^+ + x^- \quad (3.1)$$

D'autre part, on a pour tout $x, y \in \mathbb{R}^n$:

$$\sup(x, y) = x + \sup(y - x, 0) = x + (y - x)^+ = \frac{1}{2}[x + y + |x - y|]$$

$$x + y = \sup(x, y) + \inf(x, y) \quad (\text{relation de Dedekind})$$

Lemme 3.1 Soit $x \in \mathbb{R}^n$ et soient $u, v \in \mathbb{R}_+^n$ tels que :

$$x = u - v$$

Alors nous avons :

$$x^+ \leq u, \quad x^- \leq v$$

Autrement dit, la décomposition de x en $x^+ - x^-$ est minimale.

Preuve: Nous avons :

$$x^+ = (u - v)^+ = \sup(u - v, 0) = \sup(u, v) - v$$

Donc, si $u > v$ alors il vient :

$$x^+ = u - v \leq u, \quad \text{car } u, v \geq 0$$

et si $u \leq v$, alors :

$$x^+ = v - v = 0 \leq u$$

Pour montrer $x^- \leq v$, il suffit de remarquer que :

$$x^- = (u - v)^- = -\inf(u - v, 0) = \sup(v - u, 0) = \sup(u, v) - u$$

et d'appliquer le raisonnement précédent. ■

De la même manière, pour toute matrice $A \in \mathbb{R}^{m \times n}$ on définit :

$$\begin{aligned} A^+ &= (A_{ij}^+); & \text{où } A_{ij}^+ &= \sup(A_{ij}, 0) \\ A^- &= (A_{ij}^-); & \text{où } A_{ij}^- &= -\inf(A_{ij}, 0) \end{aligned}$$

Il est clair que :

$$A^+ \geq 0; \quad A^- \geq 0; \quad A = A^+ - A^-; \quad |A| = A^+ + A^- \quad (3.2)$$

En nous servant du lemme 3.1, nous pouvons aisément établir le lemme suivant :

Lemme 3.2 Soit $x \in \mathbb{R}^n$ et soit $A \in \mathbb{R}^{m \times n}$, alors :

$$(Ax)^+ \leq A^+x^+ + A^-x^-; \quad (Ax)^- \leq A^+x^- + A^-x^+ \quad (3.3)$$

Preuve: Remarquons que :

$$Ax = (A^+ - A^-)(x^+ - x^-) = (A^+x^+ + A^-x^-) - (A^+x^- + A^-x^+)$$

Posons :

$$u = A^+x^+ + A^-x^-, \quad v = A^+x^- + A^-x^+$$

Ainsi, l'application du Lemme 3.1 donne immédiatement le résultat. ■

\mathcal{Z} -matrice et \mathcal{M} -matrice

Définition 3.1 (Fiedler et Pták [47]) Une matrice $A \in \mathbb{R}^{n \times n}$ est dite de classe \mathcal{Z} si et seulement si tous ses éléments non-diagonaux sont non-positifs. Ces matrices sont du type :

$$A = \begin{pmatrix} a_{11} & -a_{12} & -a_{13} & -a_{14} & \cdots \\ -a_{21} & a_{22} & -a_{23} & -a_{24} & \cdots \\ -a_{31} & -a_{32} & a_{33} & -a_{34} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}, \quad a_{ij} \geq 0. \quad \blacktriangleright$$

Les matrices de ce type ont été largement étudiées et plusieurs résultats sont apparus comme des conséquences des théorèmes développés sur ces matrices. Certains de ces théorèmes sont utilisés dans ce chapitre pour démontrer quelques résultats.

Théorème 3.1 (Fiedler et Pták [47]) Soit $A \in \mathcal{Z}$. A est dite \mathcal{M} -matrice si l'une des propriétés équivalentes suivantes est vérifiée :

1. Toutes les valeurs propres réelles de A sont positives.
2. Il existe un vecteur $x \geq 0$ tel que $Ax > 0$.
3. Il existe un vecteur $x > 0$ tel que $Ax > 0$.
4. Il existe un vecteur $x \geq 0$ tel que $A^t x > 0$.
5. Il existe un vecteur $x > 0$ tel que $A^t x > 0$.

Sur les systèmes positifs

Considérons le système linéaire suivant :

$$x(k+1) = Ax(k), \quad x(0) = x_0, \quad (3.4)$$

où $x(k) \in \mathbb{R}^n$ est le vecteur d'état du système à l'instant $k \in \mathbb{N}$ et x_0 est l'état initial.

Définition 3.2 le système (3.4) est non-négatif (resp. positif) si et seulement si pour toute condition initiale $x_0 \geq 0$ (resp. $x_0 > 0$) l'état $x(k) \geq 0$ (resp. $x(k) > 0$) pour tout $k \in \mathbb{N}$. ▶

Une condition équivalente à la positivité d'un système dynamique est donc de garantir l'invariance par (3.4) de l'orthant positif \mathbb{R}_+^n de l'espace d'état.

Lemme 3.3 ([48]) Pour que le système linéaire $x(k+1) = Ax(k)$ soit non-négatif (resp. positif) il est nécessaire et suffisant que la matrice A soit non-négative (resp. positive).

Preuve: La suffisance est triviale. Montrons alors la nécessité. En effet, soit $x_0 = e_j$ ou e_j est le vecteur dont toutes ses composantes sont égales à 0 sauf la $j^{\text{ième}}$ égale à 1. Il vient alors :

$$x(1) = Ae_j = a_j$$

où a_j et la $j^{\text{ième}}$ colonne de A . Par supposition $x(1) \geq 0$ d'où $a_j \geq 0$. ■

Sur l'inclusion des polyèdres

soient $Q \in \mathbb{R}^{q \times n}$ et $P \in \mathbb{R}^{p \times n}$ deux matrices. Définissons les ensembles suivants :

$$\begin{aligned} \mathbb{Q} &= \left\{ x \in \mathbb{R}^n \mid Qx \leq \phi \right\} \\ \mathbb{P} &= \left\{ x \in \mathbb{R}^n \mid Px \leq \psi \right\} \end{aligned}$$

Théorème 3.2 (Hennet [49]) Une condition nécessaire et suffisante pour que $\mathbb{Q} \subseteq \mathbb{P}$ est qu'il existe une matrice $K \in \mathbb{R}^{p \times q}$ telle que :

$$\begin{aligned} K_{ij} &\geq 0; \quad \forall i = 1 \cdots p, \quad j = 1 \cdots q \\ KQ &= P \\ K\phi &\leq \psi \end{aligned}$$

Le théorème 3.2 donne une condition nécessaire et suffisante du problème suivant : si x est solution du système d'inégalités $Qx \leq \phi$ alors x est aussi une solution du système d'inégalités $Px \leq \psi$

Remarque 3.1 Etant donné un ensemble :

$$\mathbb{Q} = \{x \in \mathbb{R}^n / Qx \leq \phi; \quad Q \in \mathbb{R}^{q \times n}\}$$

A partir du théorème 3.2, nous déduisons que \mathbb{Q} est positivement invariant par rapport au système linéaire autonome $x(k+1) = Ax(k)$ si et seulement s'il existe une matrice $K \in \mathbb{R}^{q \times q}$ telle que :

$$\begin{aligned} K_{ij} &\geq 0; \quad \forall i, j = 1 \cdots q \\ KQ &= QA \\ K\phi &\leq \phi \end{aligned}$$

Si $\phi = 0$, alors l'invariance positive de \mathbb{Q} équivaut aux conditions suivantes :

$$\begin{aligned} K_{ij} &\geq 0; \quad \forall i, j = 1 \cdots q \\ KQ &= QA \end{aligned} \quad \blacktriangleright$$

Sur ce, nous terminons notre aperçu général des rappels mathématiques pour passer aux sections suivantes à la définition du problème de la commande et sa solution.

3.3 Position du problème de la commande

3.3.1 Cas où le cycle est constant

Conformément à une notation déjà adoptée, on s'intéresse aux systèmes discrets linéaires définis par l'équation suivante :

$$x(k+1) = x(k) + Bu(k) + L \tag{3.5}$$

où $x \in \mathbb{R}_+^n$, $u \in \mathbb{R}_+^m$, $B \in \mathbb{R}^{n \times m}$ et $L \in \mathbb{R}^n$. Les contraintes sur l'état sont définies par :

$$\mathbb{E}_x = \{x \in \mathbb{R}^n / 0 \leq x(k) \leq x^*, \quad \forall k \geq 0, \quad \forall 0 \leq x_0 \leq x^*\} \quad (3.6)$$

L'ensemble des commandes admissibles quant à lui est défini par :

$$\mathbb{U}_u = \{u \in \mathbb{R}^m / u_{min} \leq u \leq u_{max}; \quad u_{min} > 0\} \quad (3.7)$$

Comme dans le chapitre précédent, nous supposons qu'il existe une commande $u_N \in \mathbb{U}_u$ (situation nominale, (voir remarque 1.1, chapitre 1) telle que :

$$Bu_N + L = 0 \quad (3.8)$$

Soulignons tout de suite que contrairement au chapitre 2 sur l'approche LMI, puisque l'ensemble des contraintes sur l'état est candidat à l'invariance positive en boucle fermée, alors aucun changement de variable sur l'état n'est nécessaire. Posons maintenant comme nouveau vecteur de commande :

$$v(k) = u(k) - u_N \quad (3.9)$$

Les équations (3.5), (3.8) et (3.9) donnent alors le système suivant :

$$x(k+1) = x(k) + Bv(k); \quad x \in \mathbb{R}^n; v \in \mathbb{R}^m \quad (3.10)$$

Notons tout de suite que la positivité de la commande v n'est plus demandée. Ainsi, si nous posons :

$$v_2 = u_N - u_{min}; \quad v_1 = u_{max} - u_N$$

les contraintes sur $u(k)$ se transportent sur $v(k)$ en :

$$\mathbb{U}_v = \{v \in \mathbb{R}^m / -v_2 \leq v \leq v_1; \quad v_1, v_2 > 0\} \quad (3.11)$$

Il est clair que $v \in \mathbb{U}_v$ si et seulement si $u \in \mathbb{U}_u$. La commande par retour d'état du système dynamique (3.10) aura toujours pour expression :

$$v(k) = Gx(k) \quad (3.12)$$

où $G \in \mathbb{R}^{m \times n}$. Dans ce cas, le système (3.10) en boucle fermée est donné par :

$$x(k+1) = (I + BG)x(k) \quad (3.13)$$

L'ensemble des états admissibles pour la commande pour G choisie telle que $\text{rang}(G) = m$ est :

$$\mathbb{U}_v = \{x \in \mathbb{R}^n / -v_2 \leq Gx \leq v_1; \quad v_1 > 0; v_2 > 0\} \quad (3.14)$$

Conformément à notre objectif de départ, le problème à étudier est de savoir comment chercher une loi de commande de telle sorte que l'état du système ne quitte jamais la région \mathbb{E}_x et respecte l'ensemble des états admissibles pour la commande \mathbb{U}_v .

Dans le chapitre 2, nous avons résolu ce problème par la restriction des conditions initiales de l'état en boucle fermée à un ensemble positivement invariant $\Omega(x)$ contenu dans $\mathbb{E}_x \cap \mathbb{U}_v$. Dans ce chapitre cependant, nous allons adopter une autre démarche, à savoir, assurer l'invariance positive de \mathbb{E}_x par rapport au système (3.13) et l'inclusion $\mathbb{E}_x \subseteq \mathbb{U}_v$. De cette manière si $x_0 \in \mathbb{E}_x$, alors on est sûr que $x(k) \in \mathbb{E}_x$ et $v(k) = Gx(k) \in \mathbb{U}_v$. Ainsi, l'objectif de la commande est le suivant :

Objectif : *Trouver un contrôleur par retour d'état $v = Gx$ tel que \mathbb{E}_x soit positivement invariant par rapport au système (3.13) et que :*

$$\mathbb{E}_x \subseteq \mathbb{U}_v \quad (3.15)$$

3.4 Solution du Problème de la commande

Comme dans le chapitre 2, nous allons séparer l'étude de l'invariance positive de \mathbb{E}_x et l'étude de l'inclusion $\mathbb{E}_x \subseteq \mathbb{U}_v$.

3.4.0.1 Sur l'invariance positive de \mathbb{E}_x

Remarquons tout d'abord qu'une condition nécessaire pour que l'état soit non-négatif est que la matrice $I + BG$ soit non-négative en vertu du lemme 3.3. Ainsi, une première condition sur le retour d'état $v(k) = Gx(k)$ est que la matrice de gain G doit être choisie telle que :

$$I + BG \geq 0 \quad (3.16)$$

Ceci fait, soit \mathbb{S} le cône défini par :

$$\mathbb{S} = \left\{ x \in \mathbb{R}^n \mid BGx \leq 0 \right\} \quad (3.17)$$

Proposition 3.1 *l'ensemble \mathbb{S} est positivement invariant par rapport au système (3.13).*

Preuve: Soit $x(k) \in \mathbb{S}$, donc, $BGx(k) \leq 0$, or $(I + BG) \geq 0$, d'où :

$$(I + BG)BGx(k) \leq 0$$

Il s'ensuit que :

$$(I + BG)BGx(k) = BG(I + BG)x(k) \leq 0$$

Ainsi, $x(k + 1) = (I + BG)x \in \mathbb{S}$, d'où l'invariance positive de \mathbb{S} . ■

Remarquons maintenant que si $x(k) \leq x^*$ alors $(I + BG)x(k) \leq (I + BG)x^*$, puisque $I + BG \geq 0$. Il vient alors :

$$x(k + 1) = (I + BG)x(k) \leq (I + BG)x^* = x^* + BGx^*$$

Nous en déduisons que si $x^* \in \mathbb{S}$, alors $BGx^* \leq 0$ si bien que :

$$x(k + 1) \leq x^*$$

Ainsi, l'ensemble \mathbb{E}_x sera positivement invariant. Donc, si nous imposons en plus au gain G de vérifier la relation suivante :

$$BGx^* \leq 0$$

\mathbb{E}_x sera positivement invariant par rapport au système (3.13), d'où le lemme suivant.

Lemme 3.4 *L'ensemble \mathbb{E}_x est positivement invariant par rapport au système (3.13) si et seulement si le retour d'état $v(k) = Gx$ vérifie :*

$$I + BG \geq 0 \tag{3.18}$$

$$BGx^* \leq 0 \tag{3.19}$$

Preuve: La suffisance est facilement déduite à partir du raisonnement plus haut. Montrons alors la nécessité. En effet, supposons que \mathbb{E}_x soit positivement invariant par rapport au système (3.13). Nous avons alors :

$$0 \leq x(k) \leq x^*, \quad \forall 0 \leq x_0 \leq x^*, \forall k \in \mathbb{N}$$

En vertu du lemme 3.3, le système (3.13) est nécessairement non-négatif. Par conséquent $I + BG \geq 0$. Posons maintenant $x_0 = x^*$. D'après l'invariance positive de \mathbb{E}_x , nous avons alors :

$$x(1) = (I + BG)x^* \leq x^*$$

d'où

$$BGx^* \leq 0$$

ce qui complète la preuve. ■

Nous allons voir à présent comment les résultats de Hennes du théorème 3.2 permettent de trouver les mêmes conditions du lemme 3.4. En effet, remarquons d'abord que l'ensemble \mathbb{E}_x peut s'écrire sous la forme :

$$\mathbb{E}_x = \left\{ x \in \mathbb{R}^n / \begin{pmatrix} I \\ -I \end{pmatrix} x \leq \begin{pmatrix} x^* \\ 0 \end{pmatrix} \right\} \quad (3.20)$$

Ainsi, la condition d'invariance positive du domaine \mathbb{E}_x peut se formuler ainsi :

$$x \in \mathbb{E}_x \Rightarrow (I + BG)x \in \mathbb{E}_x \quad (3.21)$$

En d'autres termes, si x est solution du système d'inégalités $\begin{pmatrix} I \\ -I \end{pmatrix} x \leq \begin{pmatrix} x^* \\ 0 \end{pmatrix}$ alors x est aussi une solution du système d'inégalités $\begin{pmatrix} I+BG \\ -(I+BG) \end{pmatrix} x \leq \begin{pmatrix} x^* \\ 0 \end{pmatrix}$. Posons $Q = \begin{pmatrix} I \\ -I \end{pmatrix}$, $P = \begin{pmatrix} I+BG \\ -(I+BG) \end{pmatrix}$ et $\phi = \psi = \begin{pmatrix} x^* \\ 0 \end{pmatrix}$. Ainsi, en vertu du théorème 3.2 et la remarque 3.1, la condition d'inclusion (3.21) équivaut à l'existence de deux matrices $H \in \mathbb{R}^{n \times n}$, $H' \in \mathbb{R}^{n \times n}$ telles que :

$$H_{ij} \geq 0; \quad \forall i, j = 1 \cdots n \quad (3.22)$$

$$H = I + BG \quad (3.23)$$

$$Hx^* \leq x^* \quad (3.24)$$

$$H'_{ij} \geq 0; \quad \forall i, j = 1 \cdots n \quad (3.25)$$

$$-H' = -(I + BG) \quad (3.26)$$

On en déduit à partir de (3.23) et (3.26) que seule la matrice H suffit pour assurer l'inclusion. Nous en déduisons alors à partir de (3.23) et (3.24) que le retour d'état $v = Gx$ doit d'abord être choisi tel que :

$$I + BG \geq 0$$

$$BGx^* \leq 0$$

Notons que la condition (3.19) peut aussi être déterminée par l'application du lemme 2.1 (voir chapitre 2). En effet, remarquons que l'inégalité suivante :

$$0 \leq x_i \leq x_i^* \quad \forall i = 1, 2, \dots, n$$

est équivalente à :

$$0 \leq \frac{x_i}{x_i^*} \leq 1 \quad \forall i = 1, 2, \dots, n$$

D'où l'on peut inférer :

$$\max_i \left\{ \frac{x_i}{x_i^*} \right\} \leq 1, \quad \forall i = 1, 2, \dots, n$$

Définissons maintenant la fonction $V(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ par :

$$V(x) = \max_i \left[\frac{x_i}{x_i^*} \right]$$

Donc, l'ensemble \mathbb{E}_x n'est autre que :

$$\mathbb{E}_x = \left\{ x \in \mathbb{R}_+^n \mid V(x) = \max_i \left[\frac{x_i}{x_i^*} \right] \leq 1 \right\}$$

Ainsi, la fonction $V(x)$ est la fonction frontière de \mathbb{E}_x . Posons $A = I + BG$. Donc, pour que \mathbb{E}_x soit positivement invariant, il suffit que (lemme 2.1) :

$$\Delta V(x) = V(Ax) - V(x) \leq 0$$

En effet :

$$\Delta V(x) = \max_i \left[\frac{(Ax)_i}{x_i^*} \right] - \max_i \left[\frac{x_i}{x_i^*} \right] = \max_i \left\{ \frac{(Ax)_i}{x_i^*} - \max_\ell \left[\frac{x_\ell}{x_\ell^*} \right] \right\}$$

Or, pour tout $i = 1, 2, \dots, n$, nous avons :

$$\begin{aligned} \frac{(Ax)_i}{x_i^*} &= \frac{1}{x_i^*} \sum_{j=1}^n a_{ij} x_j = \frac{1}{x_i^*} \sum_{j=1}^n a_{ij} x_j^* \frac{x_j}{x_j^*} \\ &\leq \frac{1}{x_i^*} \sum_{j=1}^n a_{ij} x_j^* \max_\ell \left[\frac{x_\ell}{x_\ell^*} \right] = \frac{(Ax^*)_i}{x_i^*} \max_\ell \left[\frac{x_\ell}{x_\ell^*} \right] \end{aligned}$$

Il vient alors :

$$\begin{aligned} \Delta V(x) &\leq \max_i \left\{ \frac{(Ax^*)_i}{x_i^*} \max_\ell \left[\frac{x_\ell}{x_\ell^*} \right] - \max_\ell \left[\frac{x_\ell}{x_\ell^*} \right] \right\} \\ &= \max_i \left\{ \left[\frac{(Ax^*)_i}{x_i^*} - 1 \right] \max_\ell \left[\frac{x_\ell}{x_\ell^*} \right] \right\} \end{aligned}$$

Ainsi, pour que $\Delta V(x) \leq 0$ il suffit que :

$$\frac{(Ax^*)_i}{x_i^*} - 1 \leq 0, \quad \forall i = 1, 2, \dots, n$$

Soit sous une forme matricielle :

$$(A - I)x^* \leq 0$$

Et puisque $A = I + BG$, il s'ensuit que :

$$BGx^* \leq 0$$

Ce qu'il fallait démontrer. ■

Remarque 3.2 Il vient immédiatement à partir du lemme 3.4 que le domaine \mathbb{E}_x ne peut être positivement invariant que pour les x^* appartenant au cône \mathbb{S} . Or, dire que $BGx^* \leq 0$, c'est dire que $\mathbb{S} \subseteq \mathbb{R}_+^n$ puisque $x^* > 0$. Se pose alors la question de savoir à quelle condition sur G l'ensemble \mathbb{S} est inclus dans \mathbb{R}_+^n . Dans ce qui va suivre, nous allons proposer une réponse à cette question. En effet, observons d'abord que nous pouvons écrire \mathbb{S} sous la forme :

$$\mathbb{S} = \left\{ x \in \mathbb{R}^n \mid -BGx \geq 0 \right\} \quad (3.27)$$

Donc, si $-(BG)^{-1}$ existe et $-(BG)^{-1} \geq 0$ alors pour $x \in \mathbb{S}$ nous avons $-BGx \geq 0$ ce qui implique $-(BG)^{-1}BGx \geq 0$ d'où $x \geq 0$ si bien que $\mathbb{S} \subseteq \mathbb{R}_+^n$. On se ramène alors au problème suivant : comment choisir le gain G pour que $-(BG)^{-1}$ existe et $-(BG)^{-1} \geq 0$.

Pour répondre à ce problème, nous allons préalablement donner une formulation un peu particulière du lemme suivant donné par Varga [50].

Lemme 3.5 ([50]) Soient $M, N \in \mathbb{R}^{n \times n}$ deux matrices telles que M^{-1} existe et est $M^{-1} \geq 0$ et $N \geq 0$. Posons $T = M - N$. Si T^{-1} existe et $T^{-1} \geq 0$ alors $\rho(M^{-1}N) < 1$.

D'où le corollaire suivant.

Corollaire 3.1 Soit $A \in \mathbb{R}^{n \times n}$ telle que $A \geq 0$. Posons $T = I - A$. Les conditions suivantes sont équivalentes.

1. T^{-1} existe et $T^{-1} \geq 0$
2. $\rho(A) < 1$

Preuve: 1. \Rightarrow 2. : c'est le lemme précédent. 2. \Rightarrow 1. : Si $A \geq 0$ et $\rho(A) < 1$ alors $(I - A)^{-1}$ existe et s'écrit :

$$(I - A)^{-1} = I + A + A^2 + \dots \geq 0$$

d'où $T^{-1} = (I - A)^{-1}$ existe et $T^{-1} \geq 0$. ■

Donc, pour $A = I + BG \geq 0$, le corollaire montre que si $\rho(I + BG) < 1$ alors $-(BG)^{-1}$ existe et $-(BG)^{-1} \geq 0$. Ainsi, pour que $\mathbb{S} \subseteq \mathbb{R}_+^n$, il suffit de choisir G telle que $\rho(I + BG) < 1$. Ceci peut être réalisé simplement par placement de pôles.

En outre, dire que $BGx^* \leq 0$, c'est dire qu'il existe un vecteur positif x^* tel que $-BGx^* \geq 0$. Or, puisque $I + BG \geq 0$ il s'ensuit que $-(BG)_{ij} \leq 0$ pour $i \neq j$ si bien que $-BG$ est une \mathcal{Z} -matrice (voir définition 3.1). Comme $-(BG)^{-1} \geq 0$, $-BG$ est une M -matrice (voir définition 3.1). ►

3.4.0.2 Sur l'inclusion $\mathbb{E}_x \subseteq \mathbb{U}_v$

Regardons maintenant les conditions supplémentaires que doit vérifier la matrice de gain G pour la condition d'inclusion $\mathbb{E}_x \subseteq \mathbb{U}_v$. En effet, remarquons d'abord que l'ensemble \mathbb{U}_v peut s'écrire sous la forme :

$$\mathbb{U}_v = \left\{ x \in \mathbb{R}^n / \begin{pmatrix} G \\ -G \end{pmatrix} x \leq \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}; \quad v_1, v_2 > 0 \right\} \quad (3.28)$$

La condition d'inclusion (3.15) peut alors se formuler ainsi :

$$x \in \mathbb{E}_x \Rightarrow Gx \in \mathbb{U}_v \quad (3.29)$$

En d'autres termes, si x est solution du système d'inégalités $\begin{pmatrix} I \\ -I \end{pmatrix} x \leq \begin{pmatrix} x^* \\ 0 \end{pmatrix}$ alors x est aussi une solution du système d'inégalités $\begin{pmatrix} G \\ -G \end{pmatrix} x \leq \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$. De la même manière que plus haut, posons $Q = \begin{pmatrix} I \\ -I \end{pmatrix}$, $P = \begin{pmatrix} G \\ -G \end{pmatrix}$, $\phi = \begin{pmatrix} x^* \\ 0 \end{pmatrix}$ et $\psi = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$. Ainsi, en vertu du théorème 3.2, la condition (3.29) équivaut à l'existence d'une matrice non-négative $K \in \mathbb{R}_+^{2m \times 2n}$ telle que :

$$KQ = P \quad (3.30)$$

$$K \begin{pmatrix} x^* \\ 0 \end{pmatrix} \leq \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \quad (3.31)$$

Posons :

$$K = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix}$$

Il s'ensuit à partir de (3.30) et (3.31) que :

$$K_{11} - K_{12} = G \quad (3.32)$$

$$K_{22} - K_{21} = G \quad (3.33)$$

$$K_{11}x^* \leq v_1 \quad (3.34)$$

$$K_{21}x^* \leq v_2 \quad (3.35)$$

d'où l'on peut inférer :

$$K_{11} - K_{12} = K_{22} - K_{21}$$

Remarquons tout de suite que les relations (3.34) et (3.35) peuvent se formuler ainsi :

$$\begin{pmatrix} K_{11} & O \\ O & K_{21} \end{pmatrix} \begin{pmatrix} x^* \\ x^* \end{pmatrix} \leq \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \quad (3.36)$$

Ceci fait, posons maintenant :

$$D = K_{11} - K_{12} = K_{22} - K_{21} \quad (3.37)$$

et définissons :

$$\begin{aligned} D^+ &= (D_{ij}^+); \quad \text{où } D_{ij}^+ = \sup(D_{ij}, 0) \\ D^- &= (D_{ij}^-); \quad \text{où } D_{ij}^- = -\inf(D_{ij}, 0) \end{aligned}$$

Il est clair que :

$$D^+ \geq 0; \quad D^- \geq 0; \quad D = D^+ - D^- \quad (3.38)$$

Nous avons alors le lemme suivant :

Proposition 3.2 *Les inégalités suivantes sont satisfaites :*

$$\begin{aligned} D_{ij}^+ &\leq \min\{(K_{11})_{ij}; (K_{22})_{ij}\} \\ D_{ij}^- &\leq \min\{(K_{12})_{ij}; (K_{21})_{ij}\} \end{aligned}$$

Preuve: Puisque la décomposition de D en D^+ et D^- est minimale (voir lemme 3.1) alors l'équation (3.37) donne :

$$\begin{aligned} D_{ij}^+ &\leq (K_{11})_{ij} \quad \text{et} \quad D_{ij}^+ \leq (K_{22})_{ij} \\ D_{ij}^- &\leq (K_{12})_{ij} \quad \text{et} \quad D_{ij}^- \leq (K_{21})_{ij} \end{aligned}$$

si bien que :

$$\begin{aligned} D_{ij}^+ &\leq \min\{(K_{11})_{ij}; (K_{22})_{ij}\} \\ D_{ij}^- &\leq \min\{(K_{12})_{ij}; (K_{21})_{ij}\} \end{aligned}$$

■

Il résulte de cette proposition et l'inégalité (3.36) que :

$$\begin{pmatrix} D^+ & O \\ O & D^- \end{pmatrix} \begin{pmatrix} x^* \\ x^* \end{pmatrix} \leq \begin{pmatrix} K_{11} & O \\ O & K_{21} \end{pmatrix} \begin{pmatrix} x^* \\ x^* \end{pmatrix} \leq \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \quad (3.39)$$

Par conséquent, les conditions (3.30) et (3.31) sont équivalentes à l'existence de deux matrices non-négatives $D_1 \in \mathbb{R}_+^{m \times n}$ et $D_2 \in \mathbb{R}_+^{m \times n}$ telles que :

$$D_1 - D_2 = G$$

$$\begin{pmatrix} D_1 & O \\ O & D_2 \end{pmatrix} \begin{pmatrix} x^* \\ x^* \end{pmatrix} \leq \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

En d'autres termes, la matrice de gain G doit être choisie telle que :

$$D_1 - D_2 = G \quad (3.40)$$

$$D_1 x^* \leq v_1 \quad (3.41)$$

$$D_2 x^* \leq v_2 \quad (3.42)$$

Finalement, en réunissant les conditions citées plus haut, nous avons le théorème suivant.

Théorème 3.3 *Le domaine \mathbb{E}_x est positivement invariant par rapport au système 3.13 et $\mathbb{E}_x \subseteq \mathbb{U}_v$ si et seulement s'il existe deux matrices non-négatives D_1 et D_2 telles que :*

$$D_1 - D_2 = G \quad (3.43)$$

$$D_1 x^* \leq v_1 \quad (3.44)$$

$$D_2 x^* \leq v_2 \quad (3.45)$$

$$I + BG \geq 0 \quad (3.46)$$

$$BGx^* \leq 0 \quad (3.47)$$

Nous pouvons à présent en venir à l'examen de ces conditions. Nous montrerons par une analyse fine comment obtenir les mêmes conditions établies dans [51]. En effet, soient $Y \in \mathbb{R}_+^{m \times n}$ et $Z \in \mathbb{R}_+^{m \times n}$ deux matrices non-négatives. Posons Y^i et Z^i les colonnes de Y et Z respectivement. Désignons par \mathbb{X}_{x^*} la matrice diagonale de termes diagonaux x_i^* . Ainsi nous avons :

$$\mathbb{X}_{x^*}^{-1} = \begin{pmatrix} \frac{1}{x_1^*} & 0 & \cdots & 0 \\ 0 & \frac{1}{x_2^*} & 0 & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{1}{x_n^*} \end{pmatrix}$$

La multiplication à droite de Y et Z par $\mathbb{X}_{x^*}^{-1}$ a pour effet de multiplier chaque colonne i des matrices Y et Z par $\frac{1}{x_i^*}$:

$$Y \mathbb{X}_{x^*}^{-1} = \left(Y^1 \frac{1}{x_1^*}, Y^2 \frac{1}{x_2^*}, \dots, Y^n \frac{1}{x_n^*} \right) \quad (3.48)$$

$$Z \mathbb{X}_{x^*}^{-1} = \left(Z^1 \frac{1}{x_1^*}, Z^2 \frac{1}{x_2^*}, \dots, Z^n \frac{1}{x_n^*} \right) \quad (3.49)$$

c'est-à-dire :

$$Y \mathbb{X}_{x^*}^{-1} = \begin{pmatrix} \frac{y_1^1}{x_1^*} & \frac{y_1^2}{x_2^*} & \cdots & \frac{y_1^n}{x_n^*} \\ \frac{y_2^1}{x_1^*} & \frac{y_2^2}{x_2^*} & \vdots & \frac{y_2^n}{x_n^*} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{y_m^1}{x_1^*} & \frac{y_m^2}{x_2^*} & \vdots & \frac{y_m^n}{x_n^*} \end{pmatrix}; \quad Z \mathbb{X}_{x^*}^{-1} = \begin{pmatrix} \frac{z_1^1}{x_1^*} & \frac{z_1^2}{x_2^*} & \cdots & \frac{z_1^n}{x_n^*} \\ \frac{z_2^1}{x_1^*} & \frac{z_2^2}{x_2^*} & \vdots & \frac{z_2^n}{x_n^*} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{z_m^1}{x_1^*} & \frac{z_m^2}{x_2^*} & \vdots & \frac{z_m^n}{x_n^*} \end{pmatrix}$$

La multiplication maintenant de $Y \mathbb{X}_{x^*}^{-1}$ et $Z \mathbb{X}_{x^*}^{-1}$ par x^* donne :

$$Y \mathbb{X}_{x^*}^{-1} x^* = \sum_{i=1}^n Y^i$$

$$Z \mathbb{X}_{x^*}^{-1} x^* = \sum_{i=1}^n Z^i$$

Ainsi, si nous choisissons D_1 et D_2 comme :

$$D_1 = YX_{x^*}^{-1}, \quad D_2 = ZX_{x^*}^{-1}$$

Il s'ensuit que :

$$D_1 x^* = \sum_{i=1}^n Y^i \quad (3.50)$$

$$D_2 x^* = \sum_{i=1}^n Z^i \quad (3.51)$$

Par conséquent, si nous imposons les conditions suivantes :

$$\sum_{i=1}^n Y^i \leq v_1 \quad (3.52)$$

$$\sum_{i=1}^n Z^i \leq v_2 \quad (3.53)$$

nous vérifions les inégalités (3.44), (3.45). En résumé, pour chercher D_1 et D_2 , il suffit de trouver $2n$ vecteurs non-négatifs $Y^1, \dots, Y^n, \in \mathbb{R}^m, Z^1, \dots, Z^n, \in \mathbb{R}^m$ tels que les relations (3.52) et (3.53) soient vérifiées. Ceci fait, remarquons maintenant que :

$$Gx^* = D_1 x^* - D_2 x^* = \sum_{i=1}^n Y^i - \sum_{i=1}^n Z^i = \sum_{i=1}^n (Y^i - Z^i)$$

Ainsi, pour que la condition (3.47) soit vérifiée, il suffit d'imposer l'inégalité suivante :

$$BGx^* = B \sum_{i=1}^n (Y^i - Z^i) \leq 0 \quad (3.54)$$

En ce qui concerne l'inégalité (3.46), observons d'abord que :

$$(I + BG)_{ij} = \begin{cases} 1 + b^i \frac{(Y^i - Z^i)}{x_i^*} & \text{pour } i = j; \\ b^i \frac{(Y^j - Z^j)}{x_j^*} & \text{pour } i \neq j; \end{cases}$$

où b_i est la i -ième ligne de la matrice B . Ainsi, pour vérifier l'inégalité (3.46), il suffit d'imposer les inégalités suivantes :

$$1 + b^i \frac{(Y^i - Z^i)}{x_i^*} \geq 0, \text{ pour } i = j$$

$$b^i \frac{(Y^j - Z^j)}{x_j^*} \geq 0, \text{ pour } i \neq j$$

En résumé nous avons le théorème suivant.

Théorème 3.4 *Considérons le système 3.13 avec les contraintes (3.6) et (3.11). S'il existe $2n$ vecteurs positifs $Y^1, \dots, Y^n, \in \mathbb{R}^m, Z^1, \dots, Z^n, \in \mathbb{R}^m$ tels que :*

$$B \sum_{i=1}^n (Y^i - Z^i) \leq 0 \quad (3.55)$$

$$\sum_{i=1}^n Y^i \leq v_1 \quad (3.56)$$

$$\sum_{i=1}^n Z^i \leq v_2 \quad (3.57)$$

$$1 + b^i \frac{(Y^i - Z^i)}{x_i^*} \geq 0, \quad \text{pour } i = j \quad (3.58)$$

$$b^i \frac{(Y^j - Z^j)}{x_j^*} \geq 0, \quad \text{pour } i \neq j \quad (3.59)$$

où b_i est la i -ième ligne de la matrice B , alors il existe deux matrices non-négatives D_1 et D_2 et un retour d'état $v(k) = (D_1 - D_2)x(k) = Gx(k)$ tels que \mathbb{E}_x soit positivement invariant par rapport au système (3.13) et que :

$$\mathbb{E}_x \subseteq \mathbb{U}_v \quad (3.60)$$

Preuve: Définissons D_1 et D_2 par :

$$D_1 = \left[\frac{1}{x_1^*} Y^1, \frac{1}{x_2^*} Y^2, \dots, \frac{1}{x_n^*} Y^n \right]; \quad D_2 = \left[\frac{1}{x_1^*} Z^1, \frac{1}{x_2^*} Z^2, \dots, \frac{1}{x_n^*} Z^n \right]$$

Il est clair que $D_1 \geq 0$ et $D_2 \geq 0$. Ceci fait, posons maintenant $G = D_1 - D_2$ et observons que (3.58) et (3.59) impliquent $I + BG \geq 0$. L'application maintenant du retour d'état $v = Gx$ au système donne :

$$x(k+1) = (I + BG)x(k) \quad (3.61)$$

Comme $I + BG \geq 0$, il est facile de montrer que $x_k \geq 0$ pour tout $x_0 \geq 0$. D'autre part, observons que :

$$\sum_{i=1}^n (Y^i - Z^i) = Gx^*$$

Ainsi, (3.55) implique :

$$B \sum_{i=1}^n (Y^i - Z^i) = BGx^* \leq 0. \quad (3.62)$$

Or, puisque $I + BG \geq 0$ et $x_0 \leq x^*$, il en résulte à partir de (3.62) :

$$x(1) = (I + BG)x_0 \leq (I + BG)x^* = x^* + BGx^* \leq x^*$$

si bien que :

$$x(2) = (I + BG)x(1) \leq (I + BG)x^* = x^* + BGx^* \leq x^*.$$

Par récurrence, nous pouvons déduire que :

$$x(k) = (I + BG)x(k - 1) \leq (I + BG)x^* = x^* + BGx^* \leq x^*$$

Ceci prouve le premier objectif de la commande, c'est-à-dire l'invariance positive de \mathbb{E}_x .

Reste à montrer que :

$$-v_2 \leq Gx(k) \leq v_1$$

En effet, multiplions l'inégalité $0 \leq x(k) \leq x^*$ par D_1 , il vient :

$$0 \leq D_1x(k) \leq D_1x^* \quad (3.63)$$

De même, la multiplication de l'inégalité par $-D_2$ donne :

$$-D_2x^* \leq -D_2x(k) \leq 0 \quad (3.64)$$

L'addition maintenant de (3.63) et (3.64) implique :

$$-D_2x^* \leq (D_1 - D_2)x(k) \leq D_1x^* \quad (3.65)$$

Or, les relations (3.56) et (3.57) impliquent $\sum_{i=1}^n Y^i = D_1x^* \leq v_1$ et $\sum_{i=1}^n Z^i = D_2x^* \leq v_2$, il s'ensuit que :

$$-v_2 \leq (D_1 - D_2)x(k) = Gx(k) \leq v_1 \quad (3.66)$$

ce qui complète la preuve. ■

Soulignons qu'une méthode simple de résolution de ce problème de régulation sous contraintes consiste donc, par exemple à déterminer la matrice G en résolvant par la méthode du simplexe le programme linéaire suivant :

$$\begin{aligned} & \text{Minimiser } \epsilon \\ & \text{tel que } \epsilon \geq 0 \\ & B \sum_{i=1}^n (y_i - z_i) - \epsilon \leq 0 \\ & \sum_{i=1}^n y_i - \epsilon v_1 \leq 0 \\ & \sum_{i=1}^n z_i - \epsilon v_2 \leq 0 \\ & 1 + b^i \frac{(y_i - z_i)}{x_i^*} - \epsilon \geq 0, \quad \text{pour } i = j \\ & b^i \frac{(y_j - z_j)}{x_j^*} - \epsilon \geq 0, \quad \text{pour } i \neq j \end{aligned}$$

Remarque 3.3 Si nous imposons au retour d'état, non seulement d'assurer l'invariance positive de l'ensemble \mathbb{E}_x et $\mathbb{E}_x \subseteq \mathbb{U}_v$, mais aussi d'assurer la stabilité asymptotique du système, il faut alors que l'inégalité (3.55) soit stricte. En effet, dans ce cas nous aurons $B \sum_{i=1}^n (y^i - z^i) = BGx^* < 0$. Cela posé, remarquons que la stabilité d'un système linéaire autonome $x(k+1) = Ax(k)$ est équivalente à la stabilité de son système dual $x(k+1) = A^T x(k)$. Ainsi, considérons la fonction de Lyapunov candidate suivante $V(x) = x^T x^*$ pour le système dual en boucle fermée :

$$x(k+1) = (I + BG)^T x(k)$$

Il vient alors :

$$\Delta V(x) = x^T (I + BG)x^* - x^T x^* = x^T (BG)x^* < 0$$

ce qui prouve que le système dual est stable, et par conséquent le système $x(k+1) = (I + BG)x(k)$ l'est aussi. ▶

Dans le cas où la commande est un scalaire, le théorème précédent prend une formulation différente.

Corollaire 3.2 [Cas $v \in \mathbb{R}$] *Considérons le système (3.10) avec les contraintes (3.6) et (3.11). S'il existe 2 vecteurs positifs $y; z \in \mathbb{R}^n$, tels que :*

$$B \sum_{i=1}^n (y_i - z_i) \leq 0 \tag{3.67}$$

$$\sum_{i=1}^n y_i \leq v_1 \tag{3.68}$$

$$\sum_{i=1}^n z_i \leq v_2 \tag{3.69}$$

$$1 + b^i \frac{(y_i - z_i)}{x_i^*} \geq 0, \quad \text{pour } i = j \tag{3.70}$$

$$b^i \frac{(y_j - z_j)}{x_j^*} \geq 0, \quad \text{pour } i \neq j \tag{3.71}$$

où b_i est la i -ième composante du vecteur B , alors il existe deux vecteurs non-négatifs d_1 et d_2 et un retour d'état $v(k) = (d_1 - d_2)x(k) = Gx(k)$ tels que \mathbb{E}_x soit positivement invariant pour le système (3.13) et que

$$\mathbb{E}_x \subseteq \mathbb{U}_v \tag{3.72}$$

Sur ce, nous terminons notre aperçu général du sujet pour passer à l'application des ces résultats aux modèles de carrefours.

3.4.0.3 Application aux carrefours à feux

3.4.0.4 Cas de deux phases

Supposons que l'intersection à deux phases ait pour données : $s_1 = 0.3 \text{ veh/s}$; $s_2 = 0.4 \text{ veh/s}$; $q_1 = 0.2 \text{ veh/s}$; $q_2 = 0.133 \text{ veh/s}$; $c = 120 \text{ s}$; $g_{e2max} = 90 \text{ s}$; $g_{e2min} = 30 \text{ s}$; $x_1^* = 80 \text{ veh}$; $x_2^* = 70 \text{ veh}$. Dans ce cas, nous avons :

$$B = \begin{pmatrix} 0.3 \\ -0.4 \end{pmatrix}; \quad L = \begin{pmatrix} -12 \\ 16 \end{pmatrix}$$

Dans cette situation, la commande nominale a pour valeur $g_{e2N} = u_N = 40s$. Ainsi, d'après le corollaire 3.2 et en utilisant la programmation linéaire, nous obtenons :

$$y = \begin{pmatrix} 0.655 \\ 6.655 \end{pmatrix}; \quad z = \begin{pmatrix} 4.431 \\ 2.879 \end{pmatrix}; \quad d_1 = (0.0082 \quad 0.0951); \quad d_2 = (0.0554 \quad 0.0411)$$

$$G = \begin{pmatrix} -0.0472 & 0.0539 \end{pmatrix}$$

si bien que :

$$u(k) = v(k) + u_N = Gx(k) + u_N = \begin{pmatrix} -0.0472 & 0.0539 \end{pmatrix} x(k) + 40$$

Les performances de la commande sont montrées sur la figure 3.1. Comme nous pouvons le voir, les longueurs des files d'attente x_1, x_2 restent inférieurs au niveau correspondant à la capacité des voies x_1^*, x_2^* respectivement. Nous remarquons aussi que les files d'attente se stabilisent à partir du cycle 40 aux valeurs 26 et 24 respectivement. Ainsi, cette commande satisfait les contraintes (3.6) sur les files d'attente et empêche donc la congestion.

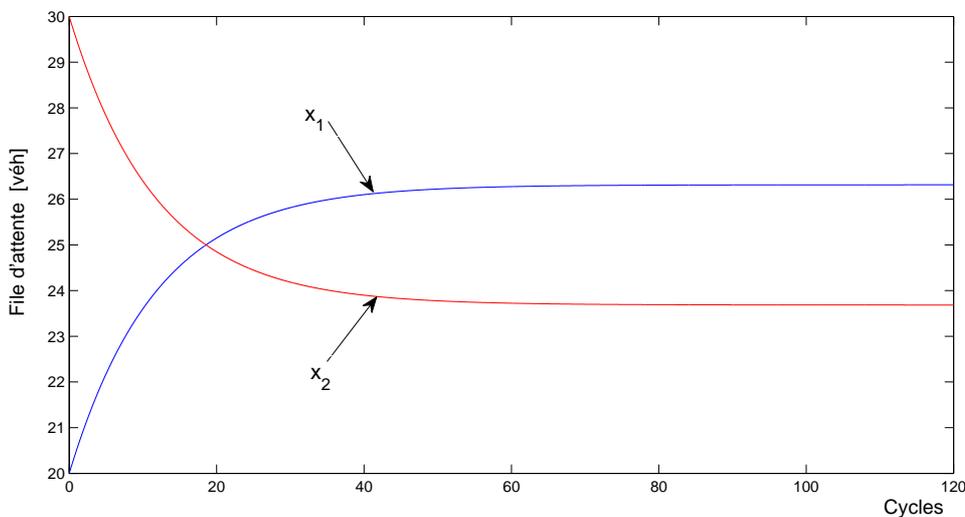


FIGURE 3.1 – Evolution des files d'attente.

Le tableau 3.1 suivant présente les variations du vert effectif g_{e_2} sur les 20 premiers cycles. Remarquons que g_{e_2} augmente entre 46.90 s et 79.86 s d'une façon continue, tout en respectant les conditions aux limites $30 \leq g_{e_i} \leq 90$ définies dans (3.7).

cycles	1	2	3	4	5	6	7	8	9	10
g_{e_2}	46.90	49.80	49.91	50.03	51.65	53.96	54.20	56.47	56.54	60.19
cycles	11	12	13	14	15	16	17	18	19	20
g_{e_2}	66.97	67.04	69.62	69.65	70.17	72.56	74.01	74.25	78.79	79.86

TABLE 3.1 – Variation du feu vert effectif g_{e_2} en fonction du cycle.

3.4.0.5 Cas de trois phases

Dans le cas d'une intersection à trois phases, les données de la simulation sont les suivantes : $s_1 = 0.48$ veh/s ; $s_2 = 0.35$ veh/s ; $s_3 = 0.55$ veh/s ; $q_1 = 0.16$ veh/s ; $q_2 = 0.11$ veh/s ; $q_3 = 0.18$ veh/s ; $c = 120$ s ; $g_{e_{1max}} = 50$ s ; $g_{e_{1min}} = 35$ s ; $g_{e_{2max}} = 50$ s ; $g_{e_{2min}} = 35$ s ; $x_1^* = 70$ veh ; $x_2^* = 95$ veh ; $x_3^* = 90$ veh. Dans ce cas, nous avons :

$$B = \begin{pmatrix} -0.48 & 0 \\ 0 & -0.35 \\ 0.55 & 0.55 \end{pmatrix} ; L = \begin{pmatrix} 19.20 \\ 14.35 \\ -44.85 \end{pmatrix}$$

Le vecteur de commande nominale a pour valeur $u_N = (40, 41)$. L'application de la programmation linéaire donne :

$$y_1 = \begin{pmatrix} 3.776 \\ 0.684 \end{pmatrix} ; y_2 = \begin{pmatrix} 0.509 \\ 4.340 \end{pmatrix} ; y_3 = \begin{pmatrix} 0.248 \\ 0.296 \end{pmatrix} ; z_1 = \begin{pmatrix} 0.208 \\ 2.568 \end{pmatrix} ; z_2 = \begin{pmatrix} 2.353 \\ 0.238 \end{pmatrix}$$

$$z_3 = \begin{pmatrix} 1.972 \\ 2.514 \end{pmatrix} ; D_1 = \begin{pmatrix} 0.0540 & 0.0102 & 0.0041 \\ 0.0098 & 0.0868 & 0.0049 \end{pmatrix} ; D_2 = \begin{pmatrix} 0.0030 & 0.0471 & 0.0471 \\ 0.0367 & 0.0048 & 0.0048 \end{pmatrix}$$

$$G = \begin{pmatrix} 0.0510 & -0.0369 & -0.0429 \\ -0.0269 & 0.0820 & 0.0002 \end{pmatrix}$$

Les résultats de la simulation sont présentés sur la figure 3.2. Nous pouvons immédiatement observer dans ce cas aussi que la commande $u(k) = Gx(k) + u_N$ respecte les contraintes sur les longueurs des files d'attente.

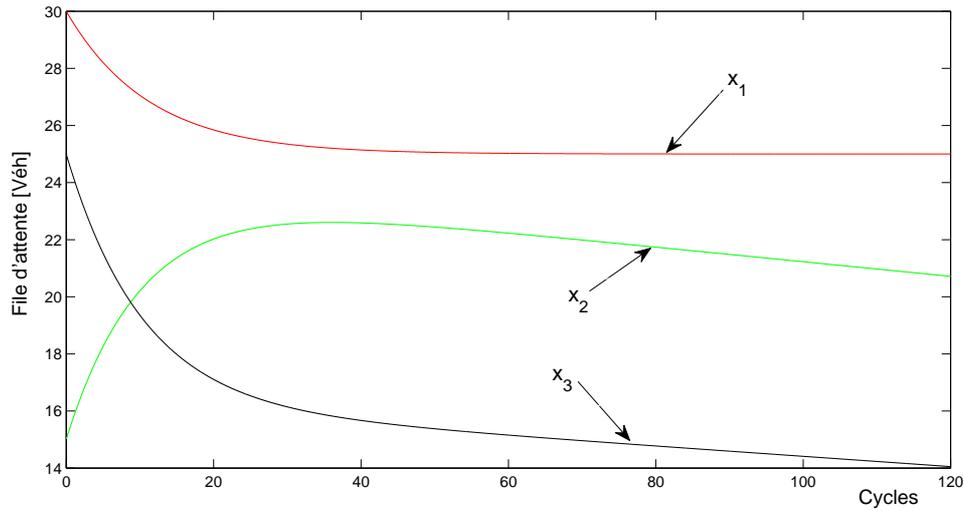


FIGURE 3.2 – Evolution des files d'attente.

En outre, le tableau 3.2 illustre les variations correspondantes des feux verts effectifs g_{e_1} , g_{e_2} pendant les 20 premiers cycles. Nous pouvons constater facilement que les valeurs de g_{e_1} et g_{e_2} varient d'un cycle à un autre tout en respectant les contraintes (3.7).

cycles	1	2	3	4	5	6	7	8	9	10
g_{e_1}	46.84	49.71	50	49.21	49.82	45.54	47.22	48.21	44.32	42.59
g_{e_2}	42.28	48.91	48.16	47.85	44.74	45.53	45.50	43.78	42.71	40.96
cycles	11	12	13	14	15	16	17	18	19	20
g_{e_1}	42.74	43.41	43.70	41.96	41.64	39.45	41.73	38.96	38.96	40.78
g_{e_2}	41.28	39.92	43.84	44.95	45.04	39.16	37.06	39.31	42.16	42.30

TABLE 3.2 – Variation des feux verts effectifs (g_{e_1} , g_{e_2}) en fonction du cycle.

3.4.1 Cas où le cycle des feux est variable

Comme dans le chapitre 2, dans le cas où le cycle est variable, nous nous intéressons au modèle d'état suivant :

$$x(k+1) = x(k) + Bu(k) \quad (3.73)$$

où $x \in \mathbb{R}_+^n$, $u \in \mathbb{R}_+^m$ et $B \in \mathbb{R}^{n \times m}$. Les contraintes sur l'état sont toujours définies par :

$$\mathbb{E}_x = \{x \in \mathbb{R}_+^n / 0 \leq x(k) \leq x^*, \quad \forall k \geq 0, \quad \forall 0 \leq x_0 \leq x^*\} \quad (3.74)$$

L'ensemble des commande admissible quant à lui est défini par :

$$\mathbb{U}_u = \{u \in \mathbb{R}_+^m / u_{min} \leq u \leq u_{max}; \quad u_{min} > 0\} \quad (3.75)$$

Comme dans le chapitre 2, nous allons séparer l'étude, dans la suite, en fonction du $\text{rang}(B)$.

3.4.1.1 Cas où $\text{rang}(B) < n$

Dans ce cas, nous pouvons supposer l'existence d'une commande $u_N \in \mathbb{U}_u$ (situation d'équilibre du système) telle que :

$$Bu_N = 0 \quad (3.76)$$

Posons maintenant comme nouveau vecteur de commande :

$$v(k) = u(k) - u_N \quad (3.77)$$

Les équations (3.73), (A.1) et (3.77) donnent alors le système suivant :

$$x(k+1) = x(k) + Bv(k); \quad x \in \mathbb{R}^n; v \in \mathbb{R}^n \quad (3.78)$$

Comme dans le cas du cycle constant, si nous posons :

$$v_2 = u_N - u_{\min}; \quad v_1 = u_{\max} - u_N$$

les contraintes sur $u(k)$ se transportent sur $v(k)$ en :

$$\mathbb{U}_v = \{v \in \mathbb{R}^n / -v_2 \leq v \leq v_1; \quad v_1, v_2 > 0\} \quad (3.79)$$

La commande du système dynamique (3.78) aura toujours pour expression :

$$v(k) = Gx(k) \quad (3.80)$$

où $G \in \mathbb{R}^{n \times n}$. Dans ce cas, le système (3.78) en boucle fermée est donné par :

$$x(k+1) = (I + BG)x(k) \quad (3.81)$$

Dans cette situation, nous nous ramenons au cas où le cycle est constant (voir section 2.3). Cependant, le théorème 3.4 prend ici une formulation différente.

Théorème 3.5 *Considérons le système (3.78) avec les contraintes (3.74) et (3.79). S'il existe 2 vecteurs positifs $y; z \in \mathbb{R}^n$, tels que :*

$$B(y - z) \leq 0 \quad (3.82)$$

$$y \leq v_1 \quad (3.83)$$

$$z \leq v_2 \quad (3.84)$$

$$1 + b_{ii} \frac{(y_i - z_i)}{x_i^*} \geq 0, \quad \text{pour } i = j \quad (3.85)$$

$$b_{ij} \frac{(y_j - z_j)}{x_j^*} \geq 0, \quad \text{pour } i \neq j \quad (3.86)$$

où b_{ij} sont les éléments de la matrice B , alors il existe deux matrices non-négatives D_1 et D_2 et un retour d'état $v(k) = (D_1 - D_2)^T x(k) = Gx(k)$ tels que \mathbb{E}_x soit positivement invariant pour le système (3.81) et que :

$$\mathbb{E}_x \subseteq \mathbb{U}_v \quad (3.87)$$

Preuve: La preuve est similaire à celle du théorème 3.4 si nous prenons les matrices D_1 et D_2 sous les formes diagonales suivantes :

$$D_1 = \begin{pmatrix} \frac{y_1}{x_1^*} & 0 & \cdots & 0 \\ 0 & \frac{y_2}{x_2^*} & \vdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \vdots & \frac{y_n}{x_n^*} \end{pmatrix}; \quad D_2 = \begin{pmatrix} \frac{z_1}{x_1^*} & 0 & \cdots & 0 \\ 0 & \frac{z_2}{x_2^*} & \vdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \vdots & \frac{z_n}{x_n^*} \end{pmatrix}$$

3.4.1.2 Cas où $\text{rang}(B) = n$

Si $\text{rang}(B) = n$, alors la matrice B est non-singulière. Par conséquent, aucune commande nominale $u_N \neq 0$ solution de $Bu_N = 0$ ne peut exister. Une façon de contourner ce problème est de prendre le changement de variable suivant :

$$v(k) = u(k) - u_{min} \quad (3.88)$$

Dans ce cas, les équations (3.73) et (3.88) donnent alors le système suivant :

$$x(k+1) = x(k) + Bv(k) + L; \quad x \in \mathbb{R}^n; v \in \mathbb{R}^n \quad (3.89)$$

où $L = Bu_{min}$. De cette façon, les contraintes sur $u(k)$ se transportent sur $v(k)$ en :

$$\mathbb{U}_v = \{v \in \mathbb{R}^n / 0 \leq v \leq v_1; \quad v_1 > 0\} \quad (3.90)$$

où $v_1 = u_{max} - u_{min}$. La commande du système dynamique (3.89) aura toujours pour expression :

$$v(k) = Gx(k) \quad (3.91)$$

où $G \in \mathbb{R}^{n \times n}$. Dans ce cas, le système (3.89) en boucle fermée est donné par :

$$x(k+1) = (I + BG)x(k) + L \quad (3.92)$$

Exprimons dans cette situation les nouvelles conditions que doit vérifier la matrice G pour l'invariance positive du domaine \mathbb{E}_x et pour la condition d'inclusion $\mathbb{E}_x \subseteq \mathbb{U}_v$. En effet, puisque l'ensemble \mathbb{E}_x peut s'écrire sous la forme :

$$\mathbb{E}_x = \left\{ x \in \mathbb{R}^n / \begin{pmatrix} I \\ -I \end{pmatrix} x \leq \begin{pmatrix} x^* \\ 0 \end{pmatrix} \right\} \quad (3.93)$$

Alors, la condition d'invariance positive du domaine \mathbb{E}_x peut se formuler ainsi :

$$x \in \mathbb{E}_x \Rightarrow (I + BG)x + L \in \mathbb{E}_x \quad (3.94)$$

En d'autres termes, si x est solution du système d'inégalités $\begin{pmatrix} I \\ -I \end{pmatrix} x \leq \begin{pmatrix} x^* \\ 0 \end{pmatrix}$ alors x est aussi une solution du système d'inégalités $\begin{pmatrix} I+BG \\ -(I+BG) \end{pmatrix} x \leq \begin{pmatrix} x^*-L \\ L \end{pmatrix}$. Posons $Q = \begin{pmatrix} I \\ -I \end{pmatrix}$, $P = \begin{pmatrix} I+BG \\ -(I+BG) \end{pmatrix}$, $\phi = \begin{pmatrix} x^* \\ 0 \end{pmatrix}$ et $\psi = \begin{pmatrix} x^*-L \\ L \end{pmatrix}$. Ainsi, en vertu du théorème 3.2, l'invariance positive du domaine \mathbb{E}_x équivaut à l'existence d'une matrices non-négative $H \in \mathbb{R}^{2n \times 2n}$, telle que :

$$HQ = P \quad (3.95)$$

$$H \begin{pmatrix} x^* \\ 0 \end{pmatrix} \leq \begin{pmatrix} x^* - L \\ L \end{pmatrix} \quad (3.96)$$

Posons :

$$H = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix}$$

Il s'ensuit à partir de (3.95) (3.96) que :

$$H_{11} - H_{12} = I + BG \quad (3.97)$$

$$H_{22} - H_{21} = I + BG \quad (3.98)$$

$$H_{11}x^* \leq x^* - L \quad (3.99)$$

$$H_{21}x^* \leq L \quad (3.100)$$

d'où l'on peut inférer

$$I + BG = H_{11} - H_{12} = H_{22} - H_{21}$$

Remarquons tout de suite que les relations (3.99) et (3.100) peuvent se formuler ainsi :

$$\begin{pmatrix} H_{11} & O \\ O & H_{21} \end{pmatrix} \begin{pmatrix} x^* \\ x^* \end{pmatrix} \leq \begin{pmatrix} x^* - L \\ L \end{pmatrix} \quad (3.101)$$

Définissons maintenant les matrices H^+ et H^- de la manière suivante :

$$\begin{aligned} H^+ &= (H_{ij}^+); & \text{où } H_{ij}^+ &= \sup\{(I + BG)_{ij}, 0\} \\ H^- &= (H_{ij}^-); & \text{où } H_{ij}^- &= -\inf\{(I + BG)_{ij}, 0\} \end{aligned}$$

Ainsi, nous avons :

$$H^+ \geq 0; \quad H^- \geq 0; \quad I + BG = H^+ - H^- \quad (3.102)$$

Par conséquent, la proposition 3.2 montre que 3.101 est équivalente à :

$$\begin{pmatrix} H^+ & O \\ O & H^- \end{pmatrix} \begin{pmatrix} x^* \\ x^* \end{pmatrix} \leq \begin{pmatrix} x^* - L \\ L \end{pmatrix} \quad (3.103)$$

En d'autres termes, la matrice de gain G doit être choisie telle que :

$$H^+ x^* \leq x^* - L \quad (3.104)$$

$$H^- x^* \leq L \quad (3.105)$$

Regardons maintenant les conditions supplémentaires que doit vérifier la matrice de gain G pour la condition d'inclusion $\mathbb{E}_x \subseteq \mathbb{U}_v$. En effet, l'ensemble \mathbb{U}_v peut s'écrire sous la forme :

$$\mathbb{U}_v = \left\{ x \in \mathbb{R}^n / \begin{pmatrix} G \\ -G \end{pmatrix} x \leq \begin{pmatrix} v_1 \\ 0 \end{pmatrix};; v_1 > 0 \right\} \quad (3.106)$$

la condition d'inclusion $\mathbb{E}_x \subseteq \mathbb{U}_v$ peut alors se formuler ainsi :

$$x \in \mathbb{E}_x \Rightarrow Gx \in \mathbb{U}_v \quad (3.107)$$

En d'autre termes, si x est solution du système d'inégalités $\begin{pmatrix} I \\ -I \end{pmatrix} x \leq \begin{pmatrix} x^* \\ 0 \end{pmatrix}$ alors x est aussi une solution du système d'inégalités $\begin{pmatrix} G \\ -G \end{pmatrix} x \leq \begin{pmatrix} v_1 \\ 0 \end{pmatrix}$. De la même manière que plus haut, posons $Q = \begin{pmatrix} I \\ -I \end{pmatrix}$, $P = \begin{pmatrix} G \\ -G \end{pmatrix}$, $\phi = \begin{pmatrix} x^* \\ 0 \end{pmatrix}$ et $\psi = \begin{pmatrix} v_1 \\ 0 \end{pmatrix}$. Ainsi, en vertu du théorème 3.2 la condition (3.107) équivaut à l'existence d'une matrice non-négative $K \in \mathbb{R}_+^{2n \times 2n}$ telle que :

$$KQ = P \quad (3.108)$$

$$K \begin{pmatrix} x^* \\ 0 \end{pmatrix} \leq \begin{pmatrix} v_1 \\ 0 \end{pmatrix} \quad (3.109)$$

Posons

$$K = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix}$$

Il s'ensuit à partir de (3.108) (3.109) que :

$$K_{11} - K_{12} = G \quad (3.110)$$

$$K_{22} - K_{21} = G \quad (3.111)$$

$$K_{11} x^* \leq v_1 \quad (3.112)$$

$$K_{21} x^* \leq 0 \quad (3.113)$$

Remarquons tout de suite que l'inégalité $K_{21} x^* \leq 0$ ne peut être vérifiée que pour $K_{21} = 0$ car $x^* > 0$ et $K_{21} \geq 0$. Il s'ensuit dans ces conditions que :

$$K_{11} - K_{12} = K_{22} = G$$

Or, puisque $K_{22} \geq 0$, il suffit de prendre $K_{12} = 0$, d'où l'on peut inférer :

$$K_{11} = K_{22} = G$$

Nous en déduisons alors que le retour d'état $v = Gx$ doit être choisi tel que :

$$G \geq 0 \quad (3.114)$$

$$Gx^* \leq v_1 \quad (3.115)$$

Finalement, en réunissant les conditions citées plus haut, la matrice de gain G doit être choisie telle que :

$$G \geq 0 \quad (3.116)$$

$$Gx^* \leq v_1 \quad (3.117)$$

$$H^+ x^* \leq x^* - L \quad (3.118)$$

$$H^- x^* \leq L \quad (3.119)$$

avec

$$H_{ij}^+ = \sup\{(I + BG)_{ij}, 0\}, \quad H_{ij}^- = -\inf\{(I + BG)_{ij}, 0\}$$

Si en plus le gain G est choisi tel que $I + BG \geq 0$ alors dans ce cas nous avons :

$$H^+ = I + BG, \quad H^- = O$$

et les conditions (3.116)-(3.119) deviennent :

$$G \geq 0 \quad (3.120)$$

$$Gx^* \leq v_1 \quad (3.121)$$

$$BGx^* + L \leq 0 \quad (3.122)$$

$$L \geq 0 \quad (3.123)$$

Ce qui donne le théorème suivant.

Théorème 3.6 *Supposons que $L \geq 0$. Alors, s'il existe n vecteurs positifs $y^1, \dots, y^n \in \mathbb{R}^n$ tels que :*

$$B \sum_{i=1}^n y^i + L \leq 0 \quad (3.124)$$

$$\sum_{i=1}^n y^i \leq v_1 \quad (3.125)$$

$$1 + b_i \frac{y^i}{x_i^*} \geq 0; \quad \text{pour } i = j; \quad (3.126)$$

$$b_i \frac{y^j}{x_j^*} \geq 0, \text{ pour } i \neq j \quad (3.127)$$

où b_i est la i -ième ligne de la matrice B , alors il existe une commande positive $v(k) = Gx(k)$ telle que \mathbb{E}_x soit positivement invariant par rapport au système (3.92) et que :

$$\mathbb{E}_x \subseteq \mathbb{U}_v \quad (3.128)$$

Preuve: Définissons la matrice G par $G = [\frac{1}{x_1^*}y^1, \dots, \frac{1}{x_2^*}y^n] \geq 0$ et observons que (3.126) et (3.127) impliquent $I + BG \geq 0$. Ceci fait, l'application maintenant du retour d'état $v = Gx$ au système donne :

$$x(k+1) = (I + BG)x(k) + L \quad (3.129)$$

Comme $I + BG \geq 0$ et $L \geq 0$, il est facile de montrer que $x_k \geq 0$ pour tout $x_0 \geq 0$. D'autre part, observons que $\sum_{i=1}^n y^i = Gx^*$. Ainsi, (3.124) implique :

$$B \sum_{i=1}^n y^i + L = BGx^* + L \leq 0. \quad (3.130)$$

Or, puisque $I + BG \geq 0$ et $x_0 \leq x^*$, il en résulte à partir de (3.130) :

$$\begin{aligned} x(1) &= (I + BG)x_0 + L \\ &\leq (I + BG)x^* + L \\ &= x^* + BGx^* + L \leq x^*, \end{aligned}$$

si bien que :

$$\begin{aligned} x(2) &= (I + BG)x(1) + L \\ &\leq (I + BG)x^* + L \\ &= x^* + BGx^* + L \leq x^*. \end{aligned}$$

Par récurrence nous pouvons déduire que :

$$\begin{aligned} x(k) &= (I + BG)x(k-1) + L \\ &\leq (I + BG)x^* + L \\ &= x^* + BGx^* + L \leq x^*. \end{aligned}$$

Ceci prouve le premier objectif de la commande. Reste à montrer que $0 \leq v(k) \leq v_1$. En effet, puisque $0 \leq x(k) \leq x^*$ et $G > 0$, alors il vient à partir de (3.125) que $0 \leq v(k) = Gx(k) \leq Gx^* = \sum_{i=1}^n y^i \leq v_1$. Ceci termine la preuve. ■

3.4.1.3 Application aux carrefours à feux

3.4.1.4 Cas de deux phases

Sur la figure 3.3 est représentée l'évolution des files d'attente dans un carrefour à deux phases avec un cycle variable. Nous avons réalisé cette simulation en utilisant les données suivantes : $s_1 = 0.6$ veh/s ; $s_2 = 0.4$ veh/s ; $q_1 = 0.3$ veh/s ; $q_2 = 0.2$ veh/s ; $c_{max} = 120$ s ; $c_{min} = 90$ s ; $g_{e_{2max}} = 70$ s ; $g_{e_{2min}} = 30$ s ; $x_1^* = 80$ veh ; $x_2^* = 70$ veh. Dans ce cas, nous avons :

$$B = \begin{pmatrix} q_1 - s_1 & s_1 \\ q_2 & -s_2 \end{pmatrix} = \begin{pmatrix} -0.3 & 0.6 \\ 0.2 & -0.4 \end{pmatrix}$$

Remarquons que $\text{rang}(B) = 1 < n = 2$. Alors, d'après le théorème 3.5, il en résulte :

$$y = \begin{pmatrix} 9.601 \\ 11.002 \end{pmatrix} ; z = \begin{pmatrix} 1.014 \\ 6.708 \end{pmatrix} ; D_1 = \begin{pmatrix} 0.120 & 0 \\ 0 & 0.157 \end{pmatrix} ; D_2 = \begin{pmatrix} 0.012 & 0 \\ 0 & 0.095 \end{pmatrix} ;$$

$$G = \begin{pmatrix} 0.1073 & 0 \\ 0 & 0.0613 \end{pmatrix}$$

Les résultats des simulations montrent clairement que la dissipation des files d'attente, dans le cas où le cycle est variable, se fait d'une façon plus efficace que dans le cas où le cycle est constant. Cela s'explique par le fait que le cycle variable représente un degré de liberté dont la commande dispose.

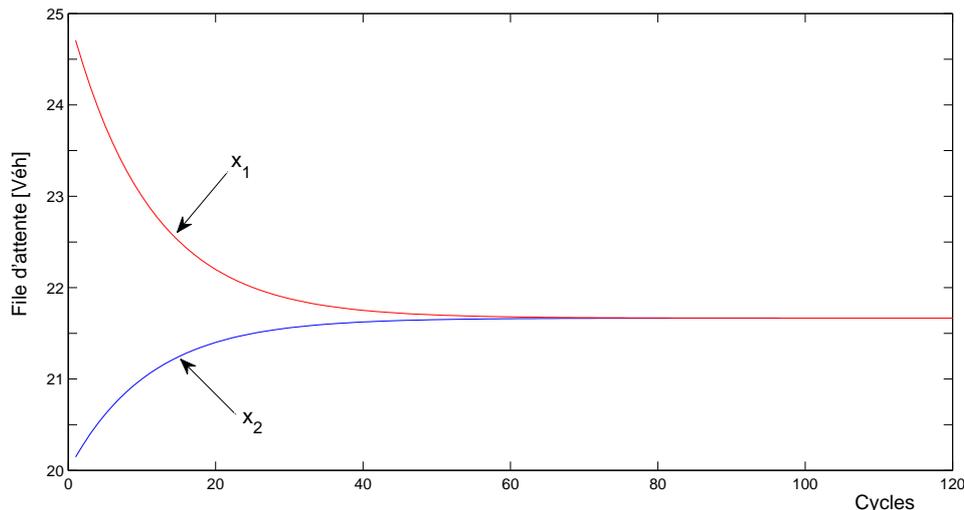


FIGURE 3.3 – Évolution des files d'attente.

Le tableau 3.3 suivant permet d'avoir une vue sur l'évolution de la durée du cycle $c(k)$ ainsi que le vert effectif g_{e_2} pendant les 20 premiers cycles.

cycles	1	2	3	4	5	6	7	8	9	10
$c(k)$	104.05	105.91	106.02	103.24	99.72	103.15	109.54	109.02	106.72	102.52
g_{e_2}	64.05	62.82	59.60	59.48	58.05	55.48	54.51	54.43	51.78	50.05
cycles	11	12	13	14	15	16	17	18	19	20
$c(k)$	98.38	97.48	100	99.91	99.05	101.71	105.92	104.26	101.47	98.59
g_{e_2}	49.54	48.55	48.20	48.25	47.66	45.59	44.42	41.80	41.41	40.96

TABLE 3.3 – Variation de $c(k)$ et de g_{e_2} en fonction de k .

3.4.1.5 Cas de trois phases

Cette simulation, dont les résultats sont présentés sur la figure 3.4, a été réalisée avec les données suivantes : $s_1 = 0.4 \text{ veh/s}$; $s_2 = 0.35 \text{ veh/s}$; $s_3 = 0.4 \text{ veh/s}$; $q_1 = 0.2 \text{ veh/s}$; $q_2 = 0.15 \text{ veh/s}$; $q_3 = 0.3 \text{ veh/s}$; $c_{max} = 140 \text{ s}$; $c_{min} = 100 \text{ s}$; $g_{e_{1max}} = 50 \text{ s}$; $g_{e_{1min}} = 30 \text{ s}$; $g_{e_{2max}} = 45 \text{ s}$; $g_{e_{2min}} = 30 \text{ s}$; $x_1^* = 90 \text{ veh}$; $x_2^* = 85 \text{ veh}$; $x_3^* = 55 \text{ veh}$. Ainsi,

$$B = \begin{pmatrix} q_1 & -s_1 & 0 \\ q_2 & 0 & -s_2 \\ q_3 - s_3 & s_3 & s_3 \end{pmatrix} = \begin{pmatrix} 0.2 & -0.4 & 0 \\ 0.15 & 0 & -0.35 \\ -0.1 & 0.4 & 0.4 \end{pmatrix}$$

et $\text{rang}(B) = 3 = n$. Il s'ensuit à partir du théorème 3.6 que :

$$y_1 = \begin{pmatrix} 7.390 \\ 12.124 \\ 1.876 \end{pmatrix} ; y_2 = \begin{pmatrix} 8.947 \\ 2.642 \\ 9.157 \end{pmatrix} ; y_3 = \begin{pmatrix} 15.774 \\ 0.569 \\ 0.267 \end{pmatrix} ; G = \begin{pmatrix} 0.1232 & 0.1790 & 0.2868 \\ 0.2021 & 0.0528 & 0.0104 \\ 0.0313 & 0.1831 & 0.0049 \end{pmatrix}$$

Comme le montre la figure 3.4, les files d'attente respectent les contraintes (3.74). Ainsi, la congestion peut être évitée cette fois-ci en appliquant la commande $u(k) = Gx(k) + u_{min}$.

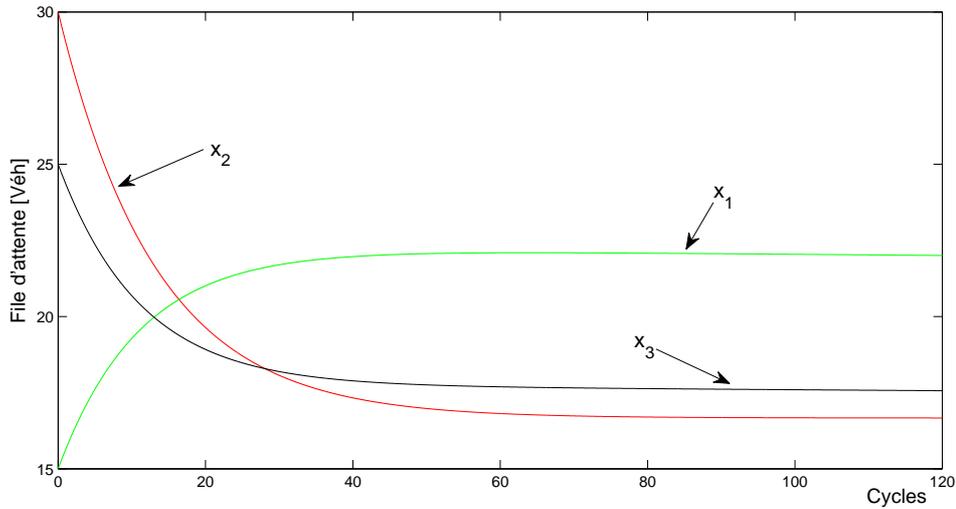


FIGURE 3.4 – Evolution des files d’attente.

Un échantillon des valeurs des paramètres de commande pendant les 20 premiers cycles est indiqué dans le tableau 3.4. Nous remarquons que la durée du cycle varie entre 116.52s et 137.24s tandis que g_{e_1} , g_{e_2} , quant à eux, varient dans les plages $[36.52s, 49.80s]$ et $[33.96s, 45s]$ s respectivement. Ce qui montre que les contraintes (3.75) sont satisfaites.

cycles	1	2	3	4	5	6	7	8	9	10
$c(k)$	116.52	118.76	121.43	126.42	128.19	131.97	126.42	131.87	137.24	134.83
g_{e_1}	49.80	48.17	45.91	41.53	41.43	39.77	38.66	38.25	39.21	39.56
g_{e_2}	35.52	38.09	39.43	39.90	41.21	40.83	42.49	45	45	45
cycles	11	12	13	14	15	16	17	18	19	20
$c(k)$	131.83	129.85	132.27	133.41	126.42	127.84	120.72	123.18	124.89	122.26
g_{e_1}	37.66	37.50	36.52	39.92	41.75	42.03	42.15	40.67	41.14	40.08
g_{e_2}	44.10	43.15	45	42.80	44.35	39.74	40.12	40.83	34.14	33.96

TABLE 3.4 – Variation de $c(k)$ et de g_{e_1} , g_{e_2} en fonction de k .

3.5 Conclusion

Dans ce chapitre, basé sur la caractérisation sous la forme de relations matricielles linéaires de l’inclusion des domaines polyédraux donnée par Hennet [49], nous avons donné des conditions nécessaires et/ou suffisantes pour la régulation à priori des carrefours signalisés. Nous avons ainsi montré que, pour une intersection isolée, la congestion peut être évitée en utilisant une commande par retour d’état ayant une structure appropriée. Ces résultats s’avèrent plus généraux que ceux de l’approche LMI relativement aux

domaines polyédraux et permettent d'étudier des cas limites difficiles comme ceux des domaines ayant des frontières qui passent par l'origine. Deux principaux objectifs ont été atteints avec la commande proposée : d'une part, lorsque la circulation n'est pas en excès, le passage des véhicules ne peut jamais dépasser le niveau de trafic correspondant à l'optimum opérationnel des lignes. D'autre part, afin de respecter les contraintes physiques inhérentes à la signalisation des carrefours, la commande proposée respecte les conditions aux limites.

Bien que les résultats obtenus dans le chapitre 2 et 3 constituent un cadre théorique pour la régulation du trafic, ils restent néanmoins dépendants des paramètres du trafic comme le flux d'entrée q supposé constant. Cependant, dû aux différents changements qui affectent le trafic (accidents, conditions atmosphériques, comportement des conducteurs ...) ce paramètre est sujet à des variations temporels. Ainsi, la connaissance, la prédiction et la prévision de ce paramètre représentent un atout capital pour la commande. Dans ce contexte, les approches proposées sont enrichies dans le chapitre suivant par la technique des réseaux de neurones pour l'estimation temps réel du paramètre q .

CHAPITRE 4

RÉSEAUX DE NEURONES POUR LA PRÉVISION DES FLUX DU TRAFIC

4.1 Introduction

La fourniture d'informations précises en temps réel et la prévision à court terme des paramètres de trafic tels que les débits, les vitesses des voyageurs et le taux d'occupation, sont des sujets de recherche qui ont suscité un intérêt considérable dans la littérature. La réussite de la mise en œuvre de la stratégie de commande dépend, dans une grande partie, de la qualité et l'exactitude des informations données au système de commande du trafic.

Depuis le début des années 1980, les chercheurs ont utilisé une grande variété de spécifications pour modéliser les caractéristiques du trafic et produire des prévisions à court terme. La littérature dans ce domaine se concentre majoritairement sur les prévisions basées sur les modèles des séries temporelles, qui vont de l'ARIMA (modèle auto-régressif à moyenne mobile intégrée) aux réseaux de neurones artificiels (RNA).

Dans ce sens, ce chapitre propose d'enrichir les stratégies de commandes proposées précédemment par une approche neuronale pour la prévision des flux du trafic afin de donner aux commandes un caractère temps réel. Dans un premier temps, nous rappelons certaines notions relatives aux séries temporelles et aux réseaux de neurone (leurs types, leurs algorithmes d'apprentissage et leurs démarches de modélisation). Dans un deuxième temps, nous abordons le domaine d'application essentiel du chapitre, celui du développement de modèles neuronaux pour la prévision des flux d'entrée du trafic sur un carrefour réel. Deux modèles sont utilisés. D'abord le perceptron multi-couches et ensuite la carte

auto-organisatrice (SOM). Enfin, nous intégrons ces modèles neuronaux dans les stratégies de commande proposées dans les chapitres précédents afin de montrer l'intérêt de l'approche neuronale dans la prévision et la régulation du trafic.

4.2 Séries temporelles

Une série temporelle, appelée aussi série chronologique, est une suite d'observations $\{x_t, 1 \leq t \leq n\}$ d'une même variable à des différentes dates [52]. Cette variable peut être de n'importe quelle nature : économique, démographique, biologique ou météorologique. Tout ce qui est chiffrable et qui varie au cours du temps peut être représenté par une série temporelle. Généralement, l'intervalle de temps entre deux observations est considéré constant dans une même série.

Les séries temporelles sont généralement représentées sur des graphiques de valeurs (ordonnées) en fonction du temps (abscisses). Lorsqu'une série est stable autour de sa moyenne, on parle dans ce cas de **série stationnaire**. Dans le cas contraire, on parle de **série non stationnaire**. Lorsqu'une série croît sur l'ensemble de l'échantillon et donc possède une moyenne qui n'est pas constante, on parle de **tendance**. On parle de **phénomène saisonnier** lorsqu'on observe des phénomènes qui se reproduisent à des périodes régulières.

Les séries temporelles occupent une place importante dans tous les domaines de l'observation ou de la collecte de données [53]. Dans le domaine du trafic routier, les variables temporelles du trafic, à l'exemple du débit et le taux d'occupation, sont mesurées par des capteurs implantés sur des axes routiers. Ces mesures constituent des séries temporelles. Par exemple, la figure 4.1 montre une série temporelle qui décrit le débit de trafic à l'entrée d'une intersection pendant une journée.

Il est bien utile de disposer de quelques indices numériques qui résument une série temporelle (x_1, \dots, x_n) . En effet, toute tentative de modélisation nécessite au préalable une analyse descriptive des données. Cette analyse permet non seulement de se familiariser avec les données qui seront manipulées tout au long de l'étude, mais aussi de détecter les incohérences dans les données et/ou les valeurs atypiques. Les indices descriptifs les plus utilisées sont :

- les indices de tendance centrale ou de position (la moyenne) ;
- les indices de dispersion ou de variabilité (l'écart-type) ;
- les indices de dépendance.

Les indices de dépendance renseignent sur le lien existant entre les données x_t . On retrouve :

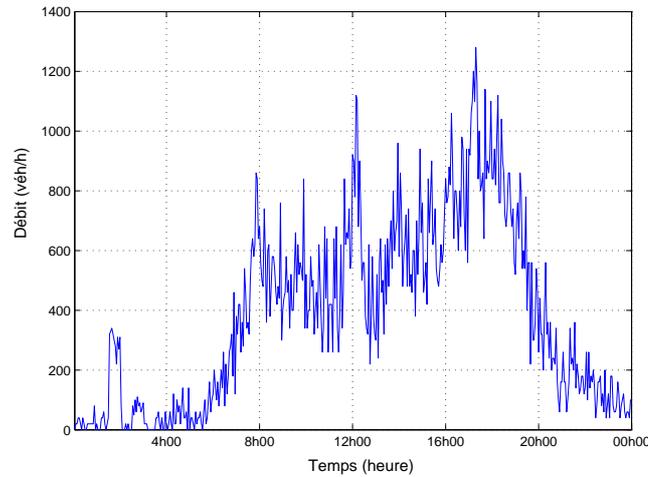


FIGURE 4.1 – Série temporelle de débit du trafic

Auto-covariance : l'auto-covariance empirique d'ordre 1 renseigne sur la dépendance entre deux données successives :

$$\hat{\sigma}_n(1) = \frac{1}{n} \sum_{t=1}^{n-1} (x_t - \bar{x}_n)(x_{t+1} - \bar{x}_n) \quad (4.1)$$

où \bar{x}_n est la moyenne. L'auto-covariance empirique d'ordre 2 renseigne sur la dépendance entre deux données écartées de deux pas de temps :

$$\hat{\sigma}_n(2) = \frac{1}{n} \sum_{t=1}^{n-2} (x_t - \bar{x}_n)(x_{t+2} - \bar{x}_n) \quad (4.2)$$

De la même manière, nous pouvons définir l'auto-covariance empirique d'ordre h :

$$\hat{\sigma}_n(h) = \frac{1}{n} \sum_{t=1}^{n-h} (x_t - \bar{x}_n)(x_{t+h} - \bar{x}_n) \quad (4.3)$$

On appelle *fonction d'auto-covariance* (empirique) la fonction qui à h associe $\hat{\sigma}_n(h)$.

Fonction d'auto-corrélation ACF : les auto-corrélations empiriques sont les quotients des covariances empiriques par la variance empirique :

$$\hat{r}_n(h) = \frac{\hat{\sigma}_n(h)}{\hat{\sigma}_n(0)} \quad (4.4)$$

où $\hat{\sigma}_n(0) = \sum_{t=1}^n (x_t - \bar{x}_n)^2$ est l'écart-type. On appelle *fonction d'auto-corrélation* (empirique) la fonction qui à h associe $\hat{r}_n(h)$.

Fonction d'auto-corrélation partielle PACF : il s'agit de la suite de valeurs $\alpha_h^{(h)} = \text{corr}(x_t, x_{t-h} | x_{t-1}, \dots, x_{t-h+1})$, qui correspondent au dernier coefficient dans une

régression linéaire de x_t et ses h dernières valeurs :

$$x_t = \alpha_0^{(h)} + \alpha_1^{(h)} x_{t-1} + \alpha_2^{(h)} x_{t-2} + \dots + \alpha_h^{(h)} x_{t-h} + \epsilon_t \quad (4.5)$$

4.2.1 Modélisation linéaire des séries temporelles mono-variable

On retrouve quatre modèles linéaires pour les séries temporelles mono-variable.

Les modèles auto-régressifs AR_p

Les modèles auto-régressifs sont construits à partir de l'idée que l'observation au temps t s'exprime linéairement par les observations précédentes. Un modèle auto-régressif X_t d'ordre p est défini comme suit :

$$X_t = \sum_{j=1}^p a_j X_{t-j} + \epsilon_t \quad (4.6)$$

où a_1, a_2, \dots, a_p sont des réels fixés, et ϵ_t est un bruit blanc centré de variance σ^2 (c'est-à-dire une suite de chocs aléatoires indépendants). L'observation X_t au temps t est alors la somme d'un choc aléatoire à l'instant t , indépendant de l'historique, et d'une fonction linéaire de son passé $\sum_{j=1}^p a_j X_{t-j}$, qui peut être vue comme la prédiction de X_t à partir des p dernières observations passées.

Les modèles à moyenne mobile MA_q

Les modèles à moyenne mobile (Moving Average) sont construits à partir de l'idée que l'observation au temps t s'exprime linéairement par les observations d'un bruit blanc ; ils sont définis par la relation suivante :

$$X_t = \epsilon_t + b_1 \epsilon_{t-1} + \dots + b_q \epsilon_{t-q} \quad (4.7)$$

où b_1, b_2, \dots, b_q sont des réels fixés et ϵ_{t-q} sont des bruits blancs centrés de variance σ^2 .

Les modèles auto-régressifs de moyenne mobile $ARMA_{pq}$

Cette classe plus générale de modèles définit des processus sous la forme d'une récurrence auto-régressive avec un second membre de type moyenne mobile. Dans les modèles ARMA, la valeur prise au temps t par la variable étudiée est une fonction linéaire de ses valeurs passées et des valeurs présentes ou passées d'un bruit blanc. Un modèle auto-régressif de moyenne mobile d'ordres p et q s'écrit sous la forme :

$$X_t = \sum_{k=1}^p a_k X_{t-k} + \sum_{j=0}^q b_j \epsilon_{t-j} \quad (4.8)$$

où les ϵ_{t-j} sont des bruits blancs centrés de variance σ^2 . Notons que les modèles ARMA sont destinés à modéliser des processus stationnaires.

Les modèles auto-régressifs à moyennes mobiles intégrées ARIMA_{pdq}

Les modèles ARIMA sont la généralisation des modèles ARMA pour des processus non stationnaires admettant une tendance. Cette catégorie de modèles a été popularisée et formalisée par Box et Jenkins [54]. L'estimation des modèles ARIMA suppose que l'on travaille sur une série stationnaire. Ceci signifie que la moyenne de la série ainsi que la variance sont constantes dans le temps. La meilleure méthode pour éliminer toute tendance est de différencier, c'est-à-dire de remplacer la série originale par la série des différences adjacentes. Une série temporelle qui a besoin d'être différenciée pour atteindre la stationnarité est considérée comme une version intégrée d'une série stationnaire (d'où le terme intégrée).

Un modèle ARIMA est étiqueté comme modèle ARIMA(p, d, q), dans lequel : p est le nombre de termes auto-régressifs, d est le nombre de différences et q est le nombre de moyennes mobiles.

Soit Δ l'opérateur de différenciation qui associe au processus (X_t) le processus ($X_t - X_{t-1}$). Nous rappelons que le processus obtenu en différenciant deux fois de suite, c'est-à-dire $X_t - 2X_{t-1} + X_{t-2}$ est noté Δ^2 . Ainsi, un processus (X_t) est un processus ARIMA(p, d, q) si le processus :

$$Y_t = \Delta^d X_t \quad (4.9)$$

est un processus ARMA(p, q). Donc, le modèle ARIMA revient à appliquer un modèle ARMA sur le processus différencié :

$$Y_t = ARMA(p, q) \quad \Leftrightarrow \quad X_t = ARIMA(p, d, q)$$

4.3 Rappels sur les réseaux de neurones artificiels (RNA)

En 1943, McCulloch et Pitts [55] ont proposé le premier modèle de base d'un neurone artificiel. Bien que très simple, il constitue l'élément de base des réseaux de neurones. La figure 4.2 montre un modèle de neurone formel.

Les entrées au neurone sont notées par x_i . Les w_i caractérisent les poids synaptiques et indiquent la force du lien entre l'entrée et le neurone. Les signaux d'entrée sont multipliés par ceux-ci avant d'être additionnés. Un poids positif est associé à une synapse excitatrice tandis qu'un poids négatif est associé à une synapse inhibitrice.

Un additionneur est utilisé pour faire la somme pondérée des entrées plus un certain décalage constant de biais θ . Le résultat est ensuite passé à travers une fonction $f()$ généralement non-linéaire qu'on appelle *fonction d'activation*. Celle-ci limite l'amplitude

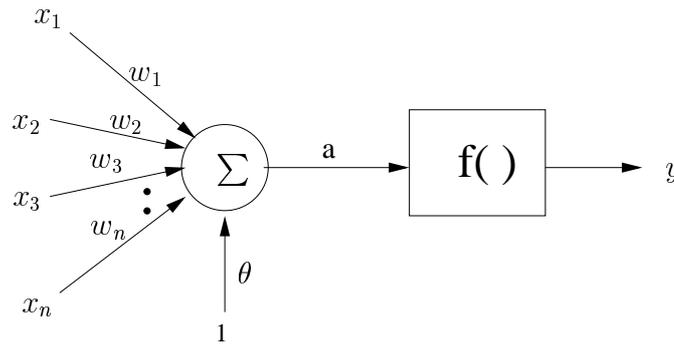


FIGURE 4.2 – Représentation du neurone formel

de sortie du neurone dans un certain intervalle borné, usuellement de $[0,1]$ ou bien encore de $[-1,1]$. On peut représenter mathématiquement le traitement que fait le neurone élémentaire par les équations suivantes :

$$a = \sum_{i=1}^n w_i \cdot x_i + \theta \quad (4.10)$$

$$y = f(a) \quad (4.11)$$

Les poids w_i et θ constituent les paramètres libres du système et doivent être ajustés. Dans les premiers travaux fondateurs de McCulloch et Pitts, la fonction d'activation utilisée pour les neurones était une simple fonction de seuillage brutale de type *signe*. Dès lors, la sortie du réseau ne pouvait prendre que des valeurs binaires (-1 ou +1). Cependant, de nos jours, il est plus usuel d'utiliser une fonction continue de forme sigmoïdale. C'est de loin la fonction la plus utilisée dans la littérature sur les réseaux de neurones artificiels (RNA). Elle a les avantages d'être continue, limitante pour la sortie (squashing) et différentiable. Algébriquement, elle s'exprime sous l'une de ces formes :

$$y(x) = \frac{1}{1 + e^{-x}}, \quad \text{log-sigmoïde} \quad (4.12)$$

$$y = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad \text{tangente hyperbolique} \quad (4.13)$$

Elles diffèrent principalement par leur intervalle de sortie, soient $[0,1]$ et $[-1,1]$ respectivement. Pour plus de détails sur les réseaux de neurones, consulter la référence [56].

4.3.1 Architectures d'un réseau de neurones

Nous entendons par *architecture* ou *topologie* d'un réseau de neurones artificiels la manière selon laquelle les neurones sont organisés. D'une façon générale, l'architecture

des réseaux de neurones artificiels peut aller d'une connectivité totale (chacun des neurones du réseau est relié à tous les autres) à une connectivité locale où les neurones ne sont liés qu'à leurs plus proches voisins.

Deux classes différentes d'architectures de réseaux de neurones peuvent être distinguées [57] :

1. Les réseaux *non bouclés* (*feed-forward*)
2. Les réseaux *récurrents* ou *bouclés*

Les réseaux non bouclés

Cette classe se distingue par l'absence de toute boucle de rétroaction de la sortie vers l'entrée, d'où l'appellation "feed-forward". En d'autres termes, la propagation des signaux s'y fait uniquement dans le sens de l'entrée vers la sortie.

Ce type de réseaux comprend deux groupes d'architectures : les réseaux mono-couche et les réseaux multi-couches. Ils diffèrent par l'existence ou non de neurones intermédiaires appelés *neurones cachés* entre les unités d'entrées et les unités de sorties appelées *nœuds sources*.

Réseaux non bouclés mono-couche (perceptron) : Ce type de réseaux possède une *couche d'entrée* recevant les stimuli à traiter par l'intermédiaire des nœuds sources. La figure 4.3 montre l'exemple d'un réseau non bouclé mono-couche à 4 nœuds d'entrée et 3 nœuds de sortie. La désignation mono-couche est attribuée à la couche de sortie (nœuds de calcul). La couche d'entrée n'est pas comptée dans ce sens vu qu'il n'y a pas de calcul fait par les nœuds sources, ils servent uniquement à recevoir les signaux d'entrée et à les transmettre à la couche suivante.

Réseaux non bouclés multi-couches : Ce type de réseaux non bouclés se caractérise par la présence d'une ou de plusieurs *couches cachées* dont les nœuds de calcul correspondants s'appellent *neurones cachés* ou *unités cachées*. Les couches cachées s'interposent entre l'entrée du réseau et sa sortie. Leur rôle est d'effectuer un prétraitement des signaux d'entrée et de transmettre les résultats correspondants à la couche de sortie où sera déterminée la réponse finale du réseau avant qu'elle soit transmise au milieu extérieur.

Dans ce type de réseaux, les entrées des neurones d'une couche particulière proviennent uniquement des sorties de la couche adjacente précédente. Les réseaux les plus fréquemment utilisés de cette catégorie sont les perceptrons multi-couches PMC (Multi-layered Perceptrons - *MLP*) [58].

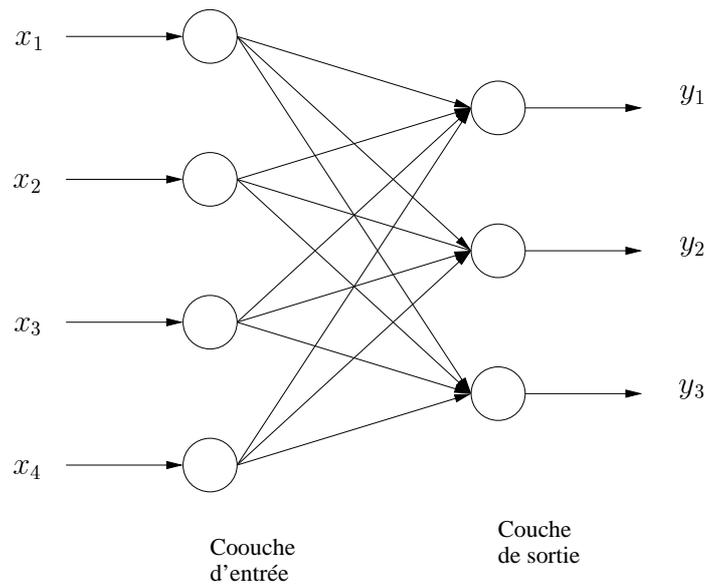


FIGURE 4.3 – Réseau non bouclé mono-couche (perceptron)

Dans la figure 4.4 est montré l'exemple d'un réseau à une seule couche cachée comportant 5 unités d'entrée, 4 unités cachées et 3 unités de sortie (réseau 5-4-3). Ce réseau est dit *complètement connecté* dans le sens où chaque noeud d'une couche est connecté à tous les noeuds de la couche adjacente suivante. Si éventuellement, des connexions manquaient entre des neurones de deux couches voisines, le réseau serait dit *partiellement connecté*.

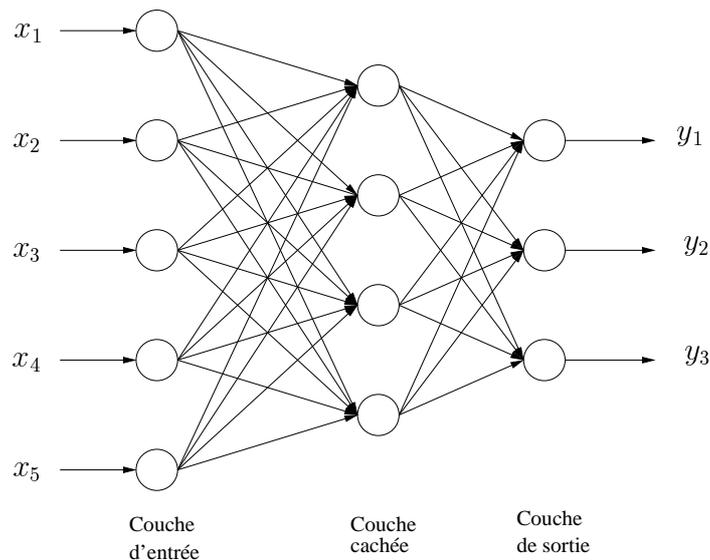


FIGURE 4.4 – Le perceptron multi-couches

Réseaux récurrents

Les réseaux *récurrents* se distinguent des réseaux non bouclés par le fait qu'ils contiennent au moins une boucle de contre-réaction des nœuds de sortie vers les nœuds d'entrée (ou, au moins d'une couche vers une couche précédente, adjacente ou non).

Dans la figure 4.5 est illustré l'exemple d'un réseau de neurones récurrent ayant 2 unités d'entrée, 3 unités cachées et 2 unités de sortie. Dans ce réseau, les connexions de rétroaction proviennent aussi bien des unités cachées que des unités de sortie.

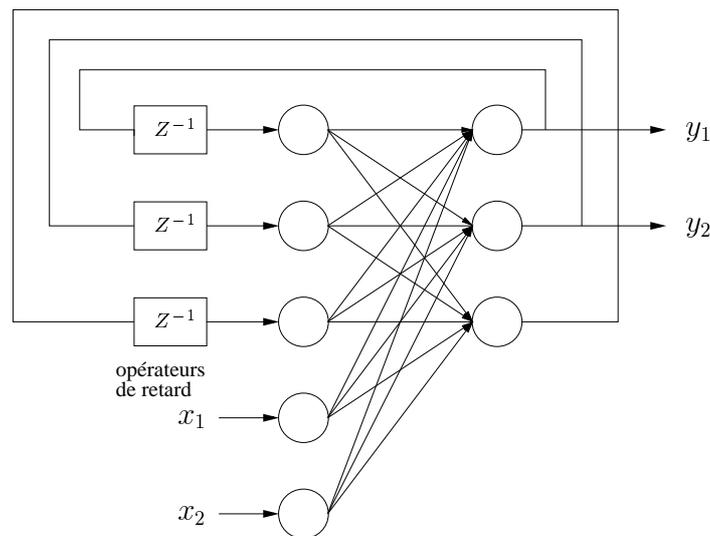


FIGURE 4.5 – Réseau récurrent avec neurones cachés

Contrairement aux réseaux non bouclés, les propriétés dynamiques du réseau récurrent sont importantes. En effet, la présence des boucles de contre-réaction implique l'insertion, dans le graphe correspondant à la topologie du réseau, de branches particulières composées d'*opérateurs de retard* (les blocs Z^{-1} de la figure 4.5). Ces derniers résultent en un comportement dynamique non linéaire du réseau dû à la nature non linéaire des neurones. Dans certains cas, les états des neurones subissent un processus de relaxation faisant évoluer le réseau vers un état stable dans lequel ces états ne peuvent plus changer. Dans d'autres cas, le changement des états des neurones de sortie est important de façon à ce que le comportement dynamique constitue la sortie du réseau. Des exemples de ce type de réseaux sont le réseau de Hopfield [59] et la machine de Boltzmann [60].

4.3.2 Apprentissage des réseaux de neurones

Parmi les propriétés importantes des réseaux de neurones, nous pouvons citer leur capacité à *apprendre* de leur environnement et à *améliorer* leur performance par apprentissage. Ce dernier se fait à travers un processus itératif d'ajustement appliqué aux poids

synaptiques et aux seuils. Intuitivement, le réseau devient de mieux en mieux informé sur son environnement après chaque itération du processus d'apprentissage. Le type d'apprentissage est déterminé par la manière selon laquelle les paramètres du réseau (poids et seuils) sont changés. Le processus d'apprentissage implique la séquence suivante [57] :

1. Le réseau de neurones est *stimulé* par son environnement,
2. Le réseau de neurones *subit des changements* suite à ce stimulus,
3. Le réseau de neurones *répond d'une façon nouvelle* à l'environnement à cause du changement qui a eu lieu dans sa structure interne.

L'ensemble des règles permettant d'adapter un réseau de neurones à un problème donné est appelé *algorithme d'apprentissage*. Plusieurs algorithmes d'apprentissage peuvent être trouvés dans la littérature, chacun ayant ses propres avantages et inconvénients. Fondamentalement, les algorithmes d'apprentissage diffèrent par la méthode d'ajustement des poids synaptiques et des seuils, mais la plupart d'entre eux sont basés sur la minimisation d'une fonction d'erreur appelée *fonction de coût*.

L'apprentissage se fait à partir d'une base d'exemples du problème traité. Il s'agit essentiellement de modifier les paramètres du réseau (notamment les poids synaptiques et seuils) pour qu'il donne la bonne réponse aux stimuli d'entrée de la base d'exemples qui lui sont présentés.

Deux principales stratégies d'apprentissage peuvent être distinguées :

1. L'apprentissage **supervisé** qui est caractérisé par la présence d'un "professeur" qui possède une connaissance approfondie de l'environnement dans lequel évolue le réseau de neurones. En pratique, les connaissances de ce professeur prennent la forme d'un ensemble de N couples (entrée, réponse attendue) que nous noterons $\{(p_1, d_1), (p_2, d_2), \dots, (p_N, d_N)\}$, où p_i désigne un stimulus (entrée) et d_i la cible pour ce stimulus, c'est-à-dire les sorties désirées du réseau. Chaque couple (p_i, d_i) correspond donc à un cas d'espèce de ce que le réseau devrait produire (la cible) pour un stimulus donné. Pour cette raison, l'apprentissage supervisé est aussi qualifié d'apprentissage par des exemples [61].

L'apprentissage supervisé est illustré d'une manière conceptuelle dans la figure 4.6. L'environnement est inconnu du réseau. Celui-ci produit à chaque pas d'apprentissage un stimulus $p(t)$ qui est acheminé à la fois au professeur et au réseau. Grâce à ses connaissances intrinsèques, le professeur produit une sortie désirée $d(t)$ pour ce stimulus. On suppose que cette réponse est optimale. Elle est ensuite comparée (par soustraction) avec la sortie du réseau $y(t)$ pour produire un signal d'erreur $e(t)$

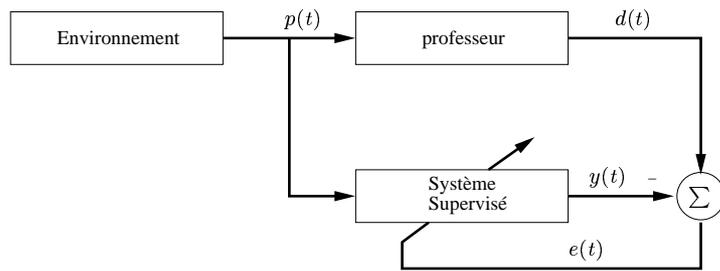


FIGURE 4.6 – Schéma bloc de l'apprentissage supervisé

qui est réinjecté dans le réseau pour modifier son comportement via une procédure itérative qui, éventuellement, lui permet de simuler la réponse du professeur. Autrement dit, la connaissance de l'environnement par le professeur est graduellement transférée vers le réseau jusqu'à l'atteinte d'un certain critère d'arrêt. Par la suite, on peut éliminer le professeur et laisser le réseau fonctionner de façon autonome.

2. L'apprentissage **non supervisé** ou **compétitif**, appelé aussi **auto-organisation** [62] est caractérisé par l'absence complète du professeur, c'est-à-dire qu'on ne dispose pas d'un signal d'erreur comme dans le cas supervisé. Nous ne disposons donc que d'un environnement qui fournit des stimuli (entrées), et d'un réseau qui doit apprendre sans intervention externe (figure 4.7). En assimilant les stimuli de l'environnement à une description de son état interne, la tâche du réseau est alors de modéliser cet état le mieux possible. Pour y arriver, il importe d'abord de définir une mesure de la qualité pour ce modèle, et de s'en servir par la suite pour optimiser les paramètres libres du réseau, c'est-à-dire ses poids synaptiques. A la fin de l'apprentissage, le réseau a développé une habilité à former des représentations internes des stimuli de l'environnement permettant d'encoder les caractéristiques de ceux-ci et, par conséquent, de créer automatiquement des classes de stimuli similaires. L'apprentissage non-supervisé s'appuie généralement sur un processus compétitif permettant d'engendrer un modèle où les poids synaptiques des neurones représentent des prototypes de stimuli.

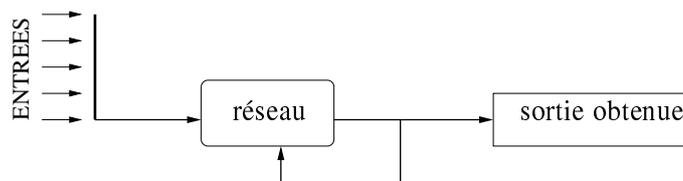


FIGURE 4.7 – Schéma bloc de l'apprentissage non supervisé

Une fois que la phase d'apprentissage est complétée avec succès, le réseau pourra être utilisé pour obtenir la réponse à des vecteurs d'entrée *inconnus*, c'est-à-dire, non rencon-

très durant l'apprentissage, ce qui constitue la phase dite de *rappel* ou de *généralisation*. Cette capacité, de réagir correctement à des motifs d'entrée non appris, donnerait aux réseaux de neurones une robustesse intrinsèque face aux fautes survenant sur le motif d'entrée [57].

4.3.3 Les algorithmes d'apprentissage supervisé

La définition d'une fonction de coût est primordiale pour les algorithmes d'apprentissage, car celle-ci sert à mesurer l'écart entre la sortie du modèle et la sortie désirée sur la dernière couche du réseau. Une gamme de fonctions de coût peut être utilisée pour optimiser les performances d'un réseau de neurones. La fonction de coût la plus utilisée est l'erreur quadratique moyenne, elle peut être écrite comme suit :

$$E(W) = \frac{1}{N} \sum_{i=1}^N E_i = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^J (d_j^{(i)} - y_j^{(i)})^2 \quad (4.14)$$

avec

- W est le vecteur de poids,
- N est le nombre d'exemples de la base d'apprentissage,
- J est le nombre de neurones de la couche de sortie,
- $y_j^{(i)}$ est la valeur de sortie du neurone j de la dernière couche obtenue lors de la présentation de l'exemple i ,
- $d_j^{(i)}$ est la valeur désirée à la sortie du neurone j .

Les algorithmes d'apprentissage utilisés nécessitent que $E(W)$ soit dérivable par rapport aux poids. Le principe de ces méthodes est de se placer en un point initial, de trouver une direction de descente du coût dans l'espace des paramètres W , puis de se déplacer d'un pas dans cette direction. On atteint un nouveau point et l'on itère la procédure jusqu'à satisfaction d'un critère d'arrêt. Ainsi, à l'itération $n + 1$, on calcule :

$$W^{(n+1)} = W^{(n)} + \eta d^{(n)} \quad (4.15)$$

η est le pas de la descente ou le pas d'apprentissage et $d^{(n)}$ est la direction de descente. Les différents algorithmes se distinguent par le choix de ces deux quantités.

La méthode de descente de gradient

L'algorithme de descente de gradient (steepest descent, en anglais) consiste à choisir comme direction de descente l'opposé du gradient de la fonction de coût ($d^{(n)} = -\nabla E(W^{(n)})$). Ainsi, à l'itération $n + 1$, la modification des poids est donnée par :

$$W^{(n+1)} = W^{(n)} - \eta \nabla E(W^{(n)}) \quad (4.16)$$

où ∇ représente le gradient de la fonction de coût par rapport aux poids W . Le choix de η doit être ni trop petit (pour la rapidité de la descente), ni trop grand (pour assurer la descente). Le calcul du gradient se fait en utilisant *l'algorithme de la rétro-propagation de l'erreur* (back-propagation) (voir Annexe B).

La méthode de descente de gradient est un algorithme efficace loin d'un minimum. Cependant, il devient très lent à l'approche d'un minimum. Dans la plupart des situations, l'inconvénient associé à la convergence lente motive l'utilisation de méthodes plus sophistiquées. Des exemples de telles méthodes seront donnés dans la suite.

La méthode de Newton

La méthode de Newton utilise la dérivée seconde de la fonction de coût pour atteindre plus rapidement le minimum. La modification des paramètres (poids) s'écrit ainsi :

$$W^{(n+1)} = W^{(n)} - H^{-1}(W^{(n)})\nabla E(W^{(n)}) \quad (4.17)$$

La direction de descente est $-H^{-1}(W^{(n)})\nabla E(W^{(n)})$ où $H^{-1}(W^{(n)})$ est l'inverse du hessien de la fonction de coût. Le pas d'apprentissage est constant fixé à un. Cet algorithme converge en une seule itération pour une fonction quadratique. C'est donc un algorithme qui est inefficace loin du minimum de la fonction et très efficace près du minimum. Dans la pratique, le calcul du hessien et surtout de son inverse est à la fois complexe et source d'instabilités numériques [63] ; on utilise de préférence une méthode de *quasi-Newton*.

La méthode de quasi-Newton

La méthode de quasi-Newton consiste à approcher l'inverse du hessien plutôt que de calculer sa valeur exacte. La modification des paramètres s'écrit :

$$W^{(n+1)} = W^{(n)} - \eta^{(n)} M(W^{(n)})\nabla E(W^{(n)}) \quad (4.18)$$

La suite $M(W^{(n)})$ est construite de façon à la faire converger vers l'inverse du hessien avec M^0 égale à la matrice identité. Cette suite est construite grâce à la méthode dite BFGS (Broyden-Fletcher-Goldfarb-Shanno) [64],[65],[66],[67], dont la vitesse de convergence est beaucoup plus grande que celle de la méthode du gradient. De plus, elle est relativement insensible au choix du pas, qui peut être déterminé économiquement par la méthode de Goldstein.

La méthode de Levenberg-Marquardt

L'algorithme de Levenberg-Marquardt [68],[69] consiste à modifier les paramètres selon la relation suivante :

$$W^{(n+1)} = W^{(n)} - [H^{(n)} + \eta^{(n)} . I]^{-1} \nabla E(W^{(n)}) \quad (4.19)$$

avec $I =$ Matrice Identité

Cette méthode est particulièrement astucieuse car elle s'adapte d'elle-même à la forme de la fonction de coût. Elle effectue un compromis entre la direction du gradient et la direction donnée par la méthode de Newton. En effet, si $\eta^{(n)}$ est grand, on reconnaît la méthode du gradient(dans ce cas la valeur du pas est donnée par $1/\eta^{(n)}$) et si $\eta^{(n)}$ est petit, la modification des paramètres correspond à celle de la méthode de Newton.

4.4 Le problème de sur-ajustement (overfitting)

Si l'on considère un ensemble d'apprentissage et une fonction de coût quadratique, il est toujours possible d'obtenir une fonction de coût aussi petite que l'on veut sur l'ensemble d'apprentissage, à condition de mettre suffisamment de neurones cachés. Cependant, le but de l'apprentissage n'est pas d'apprendre exactement la base d'apprentissage, mais le modèle sous-jacent qui a servi à engendrer les données. Or, si la fonction apprise par le réseau de neurones est ajustée trop finement aux données, elle apprend les particularités de la base d'apprentissage au détriment du modèle sous-jacent : le réseau de neurones est *sur-ajusté*.

On distingue deux familles de méthodes pour prévenir le sur-ajustement : les méthodes **d'élagage** (*pruning*) et les méthodes de **régularisation**.

L'élagage (*pruning* en anglais) est l'opération de suppression de connexions ou d'unités dont la présence est jugée inutile, voire nuisible à la généralisation. Une fois l'apprentissage du réseau achevé, la question est alors : tous les poids synaptiques ou tous les neurones participent-ils effectivement à la fonction réalisée par le réseau ? Autrement dit, quels sont les poids ou unités qui doivent être supprimés ?

Plusieurs méthodes existent pour l'élagages des poids dans un perceptron multi-couches (PMC). Des techniques telle que OBS (Optimal Brain Surgeon) [70] commencent par effectuer l'apprentissage d'un grand réseau. L'élagage des poids synaptiques est alors fondé sur la sensibilité de la fonction de coût à la variation des poids synaptiques : si la fonction de coût est peu sensible vis à vis d'un poids synaptique, celui-ci est supprimé et un nouvel apprentissage du réseau est effectué.

Les méthodes de régularisation, par opposition, ne cherchent pas à limiter la complexité du réseau, mais elles peuvent contrôler la valeur des poids pendant l'apprentissage [63]. Il devient possible d'utiliser des modèles avec un nombre élevé de poids et donc un modèle complexe, même si le nombre d'exemples d'apprentissage est faible.

Plusieurs méthodes de régularisation existent dans la littérature, comme la pénalisation de la fonction de coût (*weight decay*) qui ajoute un terme de pénalisation Ω à la fonction de coût $E(W)$ afin de favoriser les solutions les plus régulières :

$$E' = E + \alpha\Omega \quad (4.20)$$

Le terme α est un paramètre réalisant un compromis entre le fonction de coût et le terme de pénalisation Ω .

Lorsque les poids du réseau sont grands en valeur absolue, les sigmoïdes des neurones cachés sont saturées, si bien que les fonctions modélisées peuvent avoir des variations brusques. Pour obtenir des fonctions régulières, il faut travailler avec la partie linéaire des sigmoïdes, ce qui implique d'avoir des poids dont la valeur absolue est faible. La méthode de régularisation du *weight decay* limite la valeur absolue des poids en utilisant :

$$\Omega = \frac{1}{2} \sum_{i=1}^p w_i^2 \quad (4.21)$$

L'apprentissage s'effectue en minimisant :

$$\begin{aligned} E' &= E + \alpha\Omega \\ &= E + \frac{\alpha}{2} \sum_{i=1}^p w_i^2 \end{aligned} \quad (4.22)$$

où p est le nombre de poids que comporte le réseau.

4.5 Modélisation non linéaire basée sur les réseaux de neurones

D'une manière générale, les réseaux de neurones font partie de la classe des modèles non linéaires s'écrivant :

$$y = f(x_1, x_2, \dots, x_p) \quad (4.23)$$

où x_1, x_2, \dots, x_p sont p variables indépendantes représentant les entrées du modèle et y est la sortie du modèle. En ce sens, le réseau de neurones est fonctionnellement équivalent à un modèle de régression non linéaire.

Pour le problème de prévision des séries temporelles, les entrées sont généralement les dernières observations (les observations passées) de la série et la sortie est la valeur future. Ainsi, le RNA effectue la relation entrée/sortie suivante :

$$\hat{y}(t+1) = f(y(t), y(t-1), \dots, y(t-p+1)) \quad (4.24)$$

où \hat{y} est la sortie du modèle, $y(t)$ est l'observation à l'instant t , p est le nombre d'entrées et f est la fonction non linéaire réalisée par le réseau de neurones. Ainsi, le RNA est équivalent à un modèle auto-régressif non linéaire pour les problèmes de prévision des séries temporelles (Figure 4.8). Ce modèle est appelé NNAR (NNAR est l'acronyme en anglais de Neural Network Auto-Regressive).

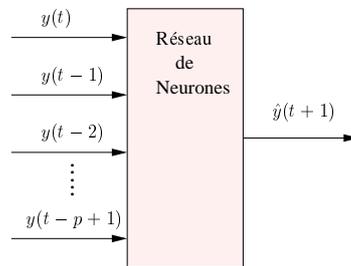


FIGURE 4.8 – Structure du modèle NNAR

Plusieurs architectures des réseaux de neurones appliquées au problème de la prévision des séries temporelles peuvent être trouvées dans la littérature. La plus populaire est le perceptron multi-couches (PMC). La figure 4.9 montre un exemple d'architecture d'un perceptron multi-couches (PMC) qui peut être utilisé pour des problèmes de prédiction des séries temporelles.

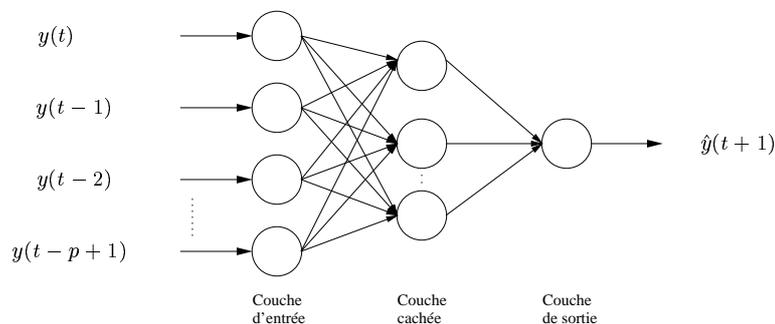


FIGURE 4.9 – Architecture de réseau de neurones pour la prédiction.

Pour un problème de prévision des séries temporelles, le vecteur d'entrée se compose d'un nombre fixe d'observations retardées de la série. Supposons que nous ayons N observations y_1, y_2, \dots, y_N dans l'ensemble d'apprentissage et nous avons besoin de prévision

d'une étape à l'avance (one-step ahead forecasting). Alors, en utilisant un RNA avec n nœuds d'entrée, nous avons $N - n$ motifs d'entraînement. Le premier motif sera composé de y_1, y_2, \dots, y_n comme entrées et y_{n+1} comme cible (la sortie désirée). Le second motif contient y_2, y_3, \dots, y_{n+1} comme entrées et y_{n+2} comme la sortie désirée et ainsi de suite jusqu'au dernier motif contenant $y_{N-n}, y_{N-n+1}, \dots, y_{N-1}$ pour les entrées et y_N pour la cible. Typiquement, la fonction de coût à minimiser pendant le processus d'apprentissage est :

$$E = \frac{1}{2} \sum_{i=n+1}^N (\hat{y}_i - y_i)^2 \quad (4.25)$$

avec \hat{y}_i est la sortie du réseau et y_i est la sortie désirée.

Démarche de modélisation par réseaux de neurones

La démarche générale de la modélisation d'un système dynamique par réseaux de neurones comprend plusieurs étapes [71],[72]. La procédure est décrite dans la figure 4.10.

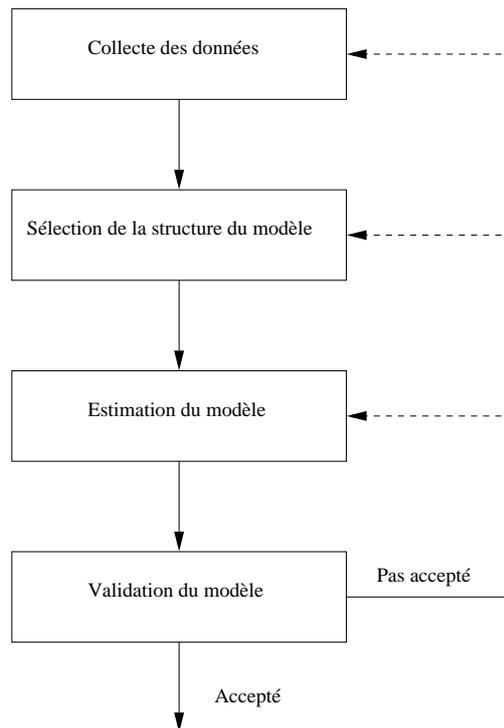


FIGURE 4.10 – La procédure de modélisation

Collecte des données : Parce que les réseaux de neurones contiennent de nombreux paramètres ajustables (les poids), on doit s'assurer de collecter autant de données que possible. Le traitement intelligent des données collectées est souvent beaucoup plus impor-

tant que d'essayer un grand nombre de différentes structures de modèle et d'algorithmes d'apprentissage. Différents types de traitement peuvent être considérés pour extraire les informations les plus valables à partir des données mesurées et les rendre appropriées pour la modélisation par réseaux de neurones.

- Mise à l'échelle : Il est recommandé de supprimer la moyenne et mettre à l'échelle tous les signaux à la même variance. Si les signaux sont mesurés dans différentes unités physiques et sans être mis à l'échelle, il y a une tendance à ce que le signal de plus grande ampleur soit trop dominant. En outre, la mise à l'échelle rend l'algorithme d'apprentissage numériquement robuste et conduit à une convergence plus rapide [73]. Enfin, l'expérience a montré qu'il tend simplement à donner de meilleurs modèles.
- Suppression de valeurs aberrantes : Il est également recommandé d'éliminer les valeurs aberrantes de l'ensemble de données (ou alternativement d'insérer des valeurs interpolées du signal de la sortie) [74]. Les valeurs aberrantes ont souvent un impact fatal sur le modèle entraîné.

Sélection de la structure du modèle neuronal : Cette étape consiste à sélectionner une structure du modèle. Malheureusement, ce problème est beaucoup plus difficile dans le cas non linéaire que dans le cas linéaire dus aux nombreux degrés de liberté. Non seulement il est nécessaire de choisir un vecteur de régression (ordre du modèle), mais aussi une architecture du réseau est requise. Le choix de la structure dépend d'un certain nombre de critères :

- type de problème à résoudre : classification, diagnostic, contrôle, prévision, . . .
- système considéré : statique ou dynamique, linéaire ou non linéaire, . . .
- contraintes de développement : temps disponible, expérience de l'utilisateur, outils disponibles (logiciels et matériels), coûts, . . .

La structure du modèle NNAR (Figure 4.8) devrait toujours être le premier choix de la structure. L'avantage d'un modèle NNAR est qu'aucun des régresseurs ne dépend des sorties passées du modèle, ce qui garantit que le prédicteur reste stable. Non seulement cela facilite l'apprentissage, mais il en résulte aussi des modèles plus robustes [74].

Estimation du modèle : Cela comprend deux tâches :

- Sélectionner un critère ou fonction de coût indiquant comment les poids devraient être déterminés à partir des données. C'est une fonction qui évalue d'une manière simple la précision d'un modèle neuronal donné. L'objectif de l'apprentissage est donc de choisir le modèle qui a la plus petite valeur du critère. Le type de base de

la fonction de coût a été donnée par (4.14). Elle est basée sur l'erreur quadratique moyenne entre les sorties désirées et les sorties de réseau.

- Choisir une méthode de recherche itérative pour minimiser le critère (un algorithme d'apprentissage). Les différentes méthodes d'apprentissage ont été présentées dans la section 4.3.3.

Validation du modèle : Le modèle obtenu n'est valable, en toute rigueur, que pour l'expérience utilisée. Il faut donc vérifier qu'il est compatible avec d'autres formes d'entrées afin de représenter correctement le fonctionnement du processus à identifier. La plupart des tests requièrent un jeu de données qui n'était pas utilisé en apprentissage. Un tel jeu de test ou de validation doit, si possible, couvrir la même gamme de fonctionnement que le jeu d'apprentissage. Plusieurs approches permettent de valider un réseau de neurones sur de telles données : visualisations, histogramme, corrélations et tests statistiques.

Revenir en arrière dans la procédure : Les chemins allant en arrière, du bloc de validation aux étapes précédentes, indiquent que la procédure est exécutée de manière itérative. Il est nécessaire de revenir dans la procédure afin de déterminer un certain nombre de modèles différents, d'essayer différentes structures de modèles et même, dans le pire des cas, refaire l'opération de collection des données.

4.6 Application des réseaux de neurones à la prévision des flux du trafic

Nous allons maintenant appliquer la procédure de développement d'un modèle neuronal pour la prévision des flux du trafic d'entrées d'un carrefour à feux. Anticipons sur la présentation qui va suivre pour noter que deux structures de modèle neuronal vont être choisies pour la prévision des flux du trafic : le perceptron multi-couches et la carte auto-organisatrice. Pour chaque modèle, nous allons suivre les étapes de modélisation décrites plus haut.

4.6.1 Description du site

Les données disponibles proviennent d'un carrefour situé sur le boulevard Anatole France (figure 4.11) qui est une artère de la ville de Belfort. Le carrefour fonctionne seulement avec deux phases et un cycle c de durée 90 secondes. Les lignes qui sont à très haute fréquence sont les lignes 1, 2 et 3 (voir la figure). La ligne 4 est relativement peu chargée.

Les lignes 2 et 3 du boulevard Anatole France ont quasiment les même caractéristiques. C'est la raisons pour laquelle nous nous intéressons uniquement aux lignes 1 et 2. Chaque ligne est caractérisée par son taux de flux de saturation qui est défini comme le nombre maximal de véhicules qui peuvent traverser une section en une unité de temps. Dans le cas de notre carrefour, nous avons relevé approximativement les valeurs suivantes : $s_1 = 0.3$ véhicule/seconde et $s_2 = 0.4$ véhicule/seconde, ce qui correspond à 1080 véhicules/heure pour la ligne 1 et 1440 véhicules/heure pour la ligne 2. De plus, la longueur de chaque ligne est fixée à $100m$ et dont les capacités sont approximativement $x_{1,max} = 50$ véhicules et $x_{2,max} = 66$ véhicules.

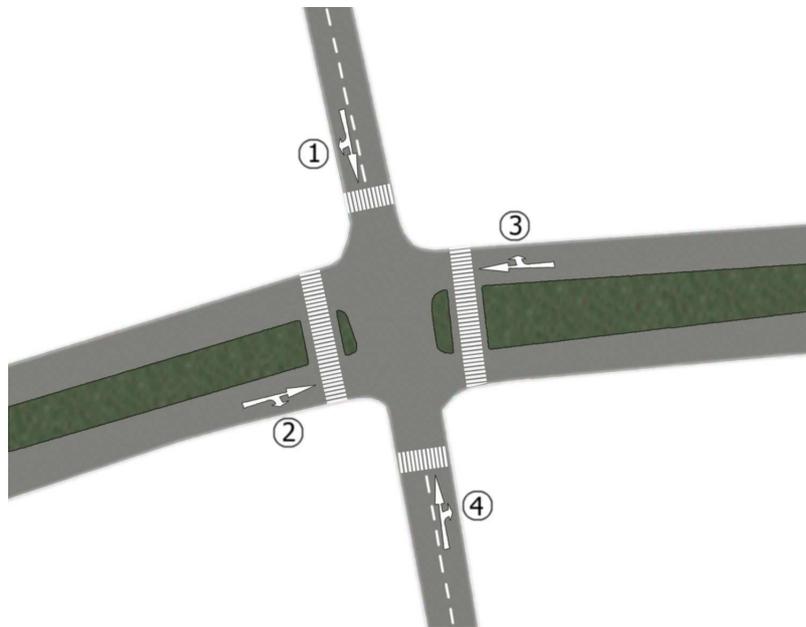


FIGURE 4.11 – Intersection-Etude de cas.

4.6.2 Collecte des données des séries temporelles

Les détecteurs à boucle d'induction sont les principales sources de mesure. Ils fournissent principalement des informations sur le flux des véhicules. Dans cette étude, les données ont été recueillies chaque 3 minutes par 2 boucles d'induction pendant 5 jours (du lundi au vendredi) en octobre 2010. Le débit considère également différents types de véhicules : véhicules légers, motos, autobus et poids-lourds. Lorsque les données ont été acquises, il a fallu commencer par une inspection visuelle afin de déterminer si un filtrage supplémentaire est nécessaire, ou s'il y a des valeurs aberrantes qui doivent être enlevées. Ensuite, les données devaient être normalisées (moyenne nulle et variance 1) avant l'apprentissage. Nous avons remarqué qu'il y avait deux lacunes dans la collecte des données : la période de 10h55 jusqu'à 11h28 et la période de 20h55 jusqu'à 21h28.

Les données manquantes sont remplacées par les données les plus proches en temps dans la journée. Enfin, les données de chaque série temporelle devaient être divisées en deux sous-ensembles. Les premiers (4 jours) sont utilisées comme l'ensemble d'apprentissage et les autres (1 jour) sont utilisées comme l'ensemble de validation.

4.6.3 Structure du modèle : cas du perceptron multi-couches (PMC)

Des perceptrons multi-couches (PMC) sont choisis dans un premier temps comme étant les structures de nos modèles neuronaux de prévision des flux q_1 et q_2 . Ainsi, la question fondamentale lorsqu'on tente de prédire les données d'une série temporelle du trafic par un PMC statique est de trouver une façon d'intégrer ses caractéristiques temporelles. En l'absence d'une mémoire interne qui pourrait représenter le comportement temporel, les PMC doivent être fournis par des entrées qui engendrent de telles caractéristiques.

En faisant ce choix, la sélection de la structure des modèles est essentiellement réduite à traiter les deux points suivants :

- Sélection des entrées du réseau (ordre du modèle)
- Sélection d'une architecture interne du réseau.

Sélection de l'ordre du Modèle (l'espace de retard) : Bien souvent, le fait de sélectionner quelques paramètres en plus ou en moins dans l'architecture du réseau a peu d'importance sur la performance du modèle. Toutefois, un mauvais choix de l'espace de retard, c'est à dire, le nombre de signaux retardés utilisés comme régresseurs, peut avoir un impact désastreux sur cette performance [74]. Un espace de retard trop petit implique évidemment que les dynamiques essentielles ne seront pas modélisés, alors qu'un espace de retard trop grand peut aussi être un problème.

Une approche souvent utilisée consiste à réutiliser les structures d'entrée (ordres) des modèles auto-régressifs linéaires (AR) comme entrées pour le réseaux de neurone. L'avantage attractif de cette approche est qu'il s'agit d'une exploitation naturelle des structures des modèles AR linéaires bien connus. Ainsi, elle permet une compréhension de l'espace de retard en calculant les ordres des modèles AR linéaires. Ces derniers sont calculés à l'aide de la fonction d'auto-corrélation partielle (PACF). La courbe d'auto-corrélation partielle est examinée pour aider à identifier l'ordre. Nous recherchons alors le point sur la courbe où les auto-corrélations partielles deviennent essentiellement zéro. Placer un intervalle de confiance à 95% pour la signification statistique est utile à cet effet. L'intervalle de confiance approximatif à 95% pour les auto-corrélations partielles est de $\pm 2/\sqrt{N}$, où N est la longueur (nombre de points) de la série temporelle en cours d'ana-

lyse. La figure 4.12 montre les résultats obtenus où les deux lignes tiretées représentent l'intervalle de confiance approximatif à 95%.

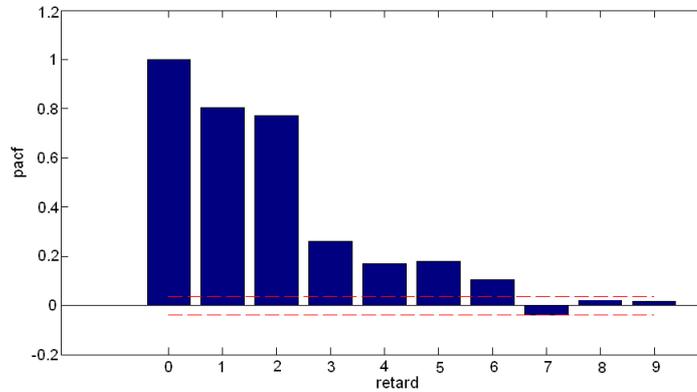
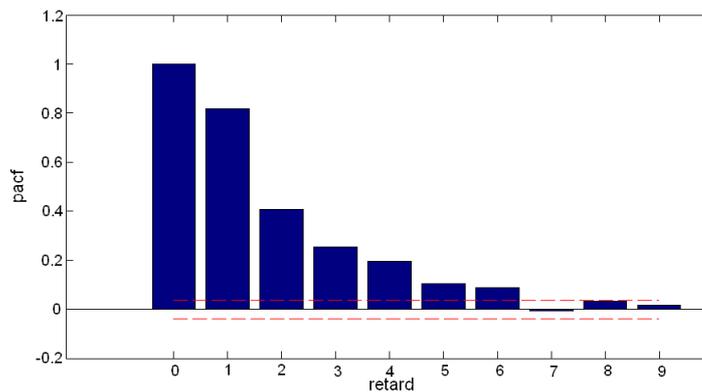
(a) PACF pour q_1 (b) PACF pour q_2

FIGURE 4.12 – La fonction d'auto-corrélation partielle (PACF)

Les ordres choisis pour les modèles de q_1, q_2 sont montrés dans le tableau 4.1.

Modèle	q_1	q_2
Ordre du modèle	6	6

TABLE 4.1 – Ordres des modèles des flux d'entrée du trafic

Sélection d'une architecture interne du réseau : L'architecture interne d'un réseau neuronal constitue sa structure formée du nombre de couche cachées et du nombre de neurones dans chaque couche cachée. Deux réseaux de neurones PMCs sont utilisés dans cette étude (un réseau pour la prévision de chaque débit du trafic q_i).

Le nombre de couches cachées choisi pour chaque modèle est égale à 1. Le nombre de

neurones dans cette couche cachée sera choisi et optimisé par la suite au procédure d'élagage. La couche de sortie de chaque réseau contient un seul neurone. Les fonction d'activation des neurones de la couche cachée sont de type tangente hyperbolique et ceux des neurones de sortie sont du type linéaire. Des travaux précédents ont montré que ce couple de fonctions permet d'approximer quasiment tous les types de relations non linéaires [75]. Une fois la structure du modèle neuronal choisie, nous passons à la prochaine étape qui est l'apprentissage des réseaux.

Apprentissage : Le but de l'apprentissage est de déterminer les paramètres du modèle, appelés poids, qui vont générer un modèle capable de donner les sorties les plus proches des sorties réelles du système. Pour cette étude, le critère choisi pour l'apprentissage est l'erreur quadratique moyenne.

L'apprentissage est effectué en utilisant la méthode de *Levenberg-Marquardt* (voir paragraphe 4.3.3). Cette méthode permet une convergence rapide. En plus, elle définit un algorithme très robuste et simple à mettre en œuvre. Le nombre maximum d'itérations est choisi égale à 700. L'apprentissage des modèles neuronaux a été réalisé en utilisant la boîte à outils Matlab NNSYSID (Neural Network Based System Identification Toolbox) [71].

Pour trouver le nombre optimal de neurones dans la couche cachée pour les deux modèles neuronaux, nous partons d'une architecture composée d'un seul neurone dans la couche cachée et nous ajoutons des neurones un par un en évaluant la qualité de prédiction. Pour chaque nombre d'unités cachées, le réseau est entraîné 5 à 7 fois, en commençant par différentes initialisations des poids à chaque fois car le réseau a généralement plus d'un minimum local. Quand il semble être évident que l'erreur de test augmente, nous nous arrêtons et choisissons une structure du modèle qui est légèrement plus grande que la structure optimale (2 à 5 unités cachés supplémentaires) de sorte que l'erreur de biais est ignorée. Les résultats sont présentés dans le tableau 4.2. Ainsi, nous avons choisi 11 et 15 neurones cachés pour les modèles de q_1 et q_2 respectivement. Ensuite, nous avons utilisé l'élagage pour faire face à l'usage excessif de poids.

Modèle	q_1	q_2
Nombre de neurones cachés	11	15

TABLE 4.2 – Nombre de neurones cachées dans les modèles neuronaux

La spécification du processus d'apprentissage des deux modèles neuronaux développés, c'est-à-dire l'algorithme d'apprentissage, le critère d'erreur, le critère d'arrêt, le

nombre maximal de cycles d'apprentissage et le weight decay sont résumés dans le tableau 4.3

Spécification d'apprentissage	
Algorithme d'apprentissage	Levenberg-Marquardt
Critère d'erreur	Erreur quadratique moyenne (MSE)
Nombre max d'itérations	700
Arrêt d'apprentissage	MSE= 10^{-5} ; No.itérations=700
Pas d'apprentissage	0.2
weight decay	10^{-3}

TABLE 4.3 – Spécification d'apprentissage

Paramètre de régularisation (weight decay) : Un problème important est le choix du paramètre de la régularisation (le decay) α (voir équation (4.20)). Dans les expériences précédentes, nous avons utilisé $\alpha = 10^{-3}$. Ce choix est obtenu en effectuant plusieurs expériences d'apprentissage avec différentes valeurs de α . Nous avons remarqué que la qualité de prévision des réseaux de neurones ne diffère pas significativement tant que $\alpha \leq 0.01$. Nous continuons donc à utiliser $\alpha = 10^{-3}$.

Élagage des réseaux (pruning) : Cette méthode permet de déterminer l'architecture optimale du réseau en supprimant les poids superflus selon la stratégie OBS (optimal brain surgeon) (voir la section 4.4). La session d'élagage élimine les poids du réseau un par un. Le réseau est réentraîné avec un maximum de 50 itérations et sa capacité de généralisation est testée après chaque élimination de poids. C'est la stratégie la plus lente mais la plus sûre. Nous avons utilisé l'algorithme programmé par [76] qui permet d'effectuer des réapprentissage entre chaque suppression de poids. La figure 4.13 montre le résultat de l'élagage des réseaux initiaux. Les traits en continu correspondent à des valeurs de poids positives et ceux en pointillés correspondent à des valeurs négatives. La ligne verticale passant par les neurones représente le biais. Le tableau 4.4 montre une comparaison entre le nombre de neurones et le nombre de poids des réseaux initiaux et ceux des réseaux finaux obtenus par l'élagage.

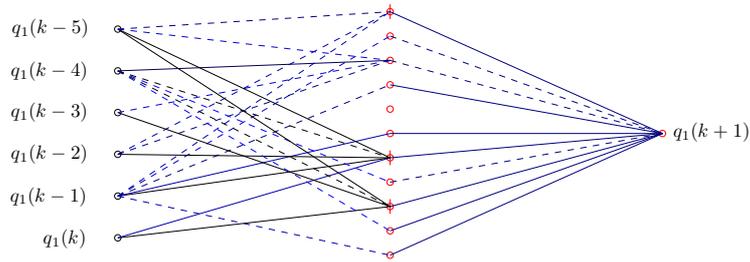
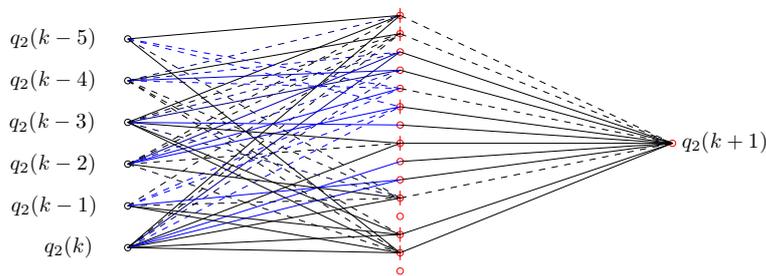
(a) La structure finale du modèle pour q_1 (b) La structure finale du modèle pour q_2

FIGURE 4.13 – Les structures finales des modèles neuronaux après l'élagage

Modèle	Structure	Nombre de neurones cachés	Nombre de poids
q_1	Initiale	11	89
	Finale	10	35
q_2	Initiale	15	121
	Finale	13	70

TABLE 4.4 – Comparaison entre les réseaux initiaux et les réseaux finaux

Lorsque l'architecture optimale des deux réseaux est trouvée, les réseaux doivent être entraînés sans régularisation (le decay = 0).

Une fois que l'étape de l'estimation des modèles neuronaux a été réalisée, nous passons à la validation des modèles.

Validation des modèles

Dans la phase de validation, les modèles estimés sont évalués sur leur adéquation à représenter le système sous-jacent. Pour valider les modèles, un deuxième ensemble de données différent de celui de l'apprentissage est utilisé. Cet ensemble de données contient les points correspondant à une journée de fonctionnement de l'intersection routière. La validation se fonde sur trois analyses : la visualisation des prévisions, la recherche de corrélations et les tests statistiques.

Visualisation des prévisions : La représentation graphique qui compare les prévisions des modèles neuronaux aux mesures réelles donne une idée de la précision des prévisions dans les différents régimes de la plage de fonctionnement du système. Elle permet de voir comment le modèle décrit le comportement dynamique du système.

Les figures 4.14 et 4.15 montrent la comparaison entre les valeurs réelles et les prédictions à un pas en avant (3 minutes) des flux d'entrée q_1 et q_2 avec les erreurs de prédiction respectivement. On remarque que les résultats sont assez satisfaisants et les valeurs prédites sont proches des valeurs mesurées .

Cependant, ce test graphique permet seulement de se donner une première idée. Il faut ensuite calculer l'histogramme de la fonction d'erreur pour confirmer les résultats. En effet, l'histogramme de l'erreur de prédiction pour les flux du trafic q_1 et q_2 est représenté sur la figure 4.16. Sa forme est presque symétrique et la majorité des valeurs de l'erreur (ou résidu) concentrées autour de zéro. C'est une preuve en plus que la plupart des informations concernant la dynamique des flux du trafic dans le carrefour ont été incorporées dans les modèles neuronaux.

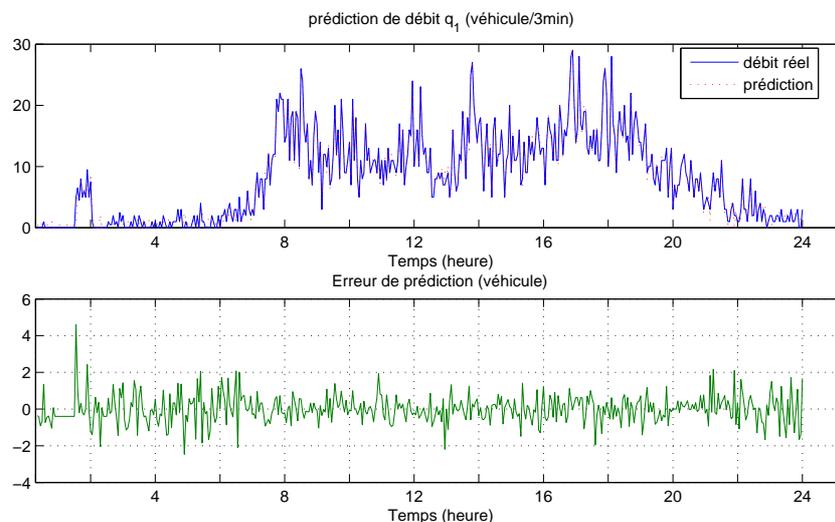


FIGURE 4.14 – La visualisation de la prédiction à un pas en avant de flux du trafic q_1

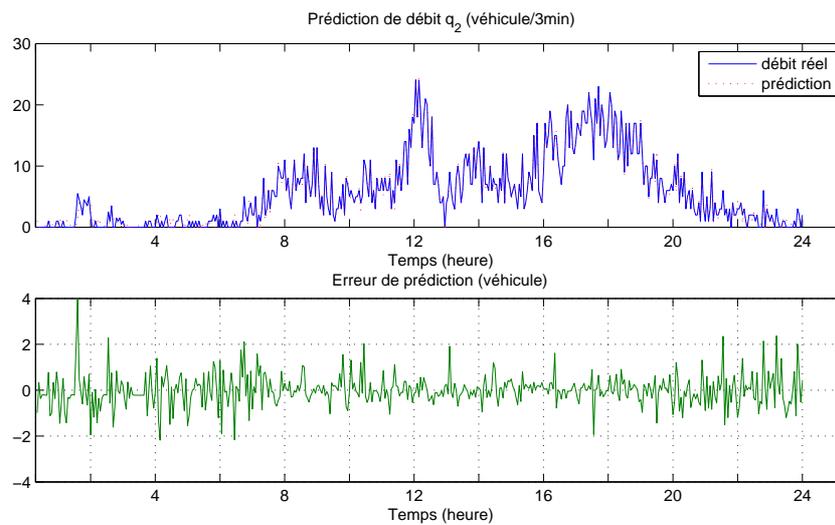
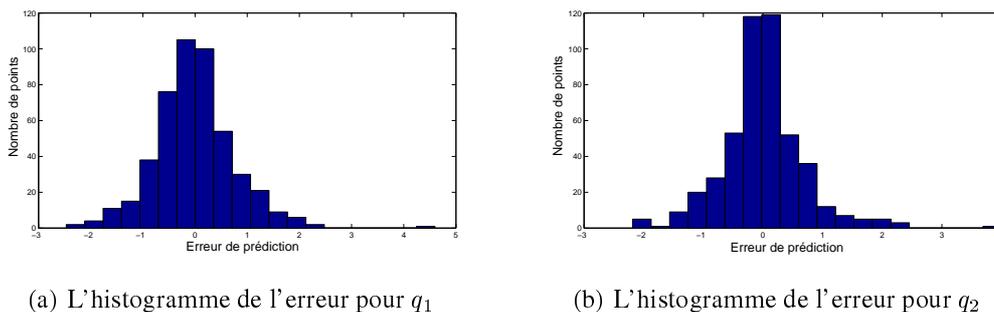
FIGURE 4.15 – La visualisation de la prévision à un pas en avant de flux du trafic q_2 (a) L'histogramme de l'erreur pour q_1 (b) L'histogramme de l'erreur pour q_2

FIGURE 4.16 – L'histogramme de l'erreur de prédiction pour les flux du trafic

Corrélation : Si les résidus (c'est à dire, les erreurs de prévision $\epsilon = y - \hat{y}$) ne contiennent pas d'informations sur les résidus passés ou sur la dynamique du système, il est probable que toutes les informations ont été extraites de l'ensemble d'apprentissage et que le modèle se rapproche le plus du système [74]. L'erreur du modèle ne doit donc pas être corrélée avec elle-même. Par conséquent, l'erreur a la structure d'un bruit blanc. La fonction d'auto-corrélation des résidus $\hat{r}_{\epsilon\epsilon}(\tau)$ peut nous renseigner si toutes les informations qui concernent la dynamique du système sont introduites dans le modèle. Cette fonction est égale à 1 pour un décalage nul ($\tau = 0$) et doit être proche de zéro pour les autres décalages ($\tau \neq 0$) :

$$\hat{r}_{\epsilon\epsilon}(\tau) = \frac{\sum_{t=1}^{N-\tau} (\epsilon_t - \bar{\epsilon})(\epsilon_{t-\tau} - \bar{\epsilon})}{\sum_{t=1}^N (\epsilon_t - \bar{\epsilon})^2} \quad (4.26)$$

où $\bar{\epsilon} = \frac{1}{N} \sum_{t=1}^N \epsilon_t$ est la moyenne du signal ϵ_t . La figure 4.17 montre la fonction d'auto-corrélation des résidus en fonction du décalage pour les flux du trafic q_1 et q_2 , respectivement. Pour des décalages non nuls, elle reste comprise dans l'intervalle de confiance, ce qui signifie que l'erreur de prédiction (résidu) pour les deux flux du trafic a une structure comme un bruit blanc. C'est une indication de la bonne performance des modèles proposés.

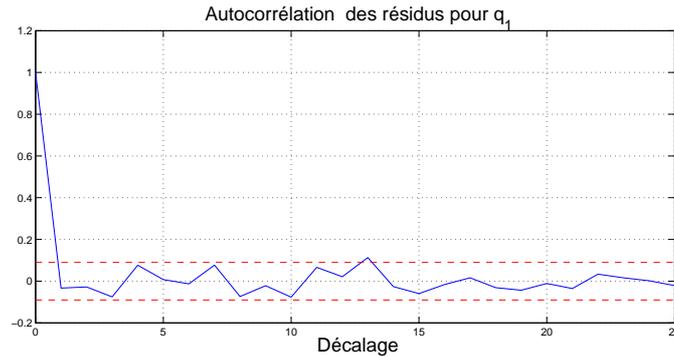
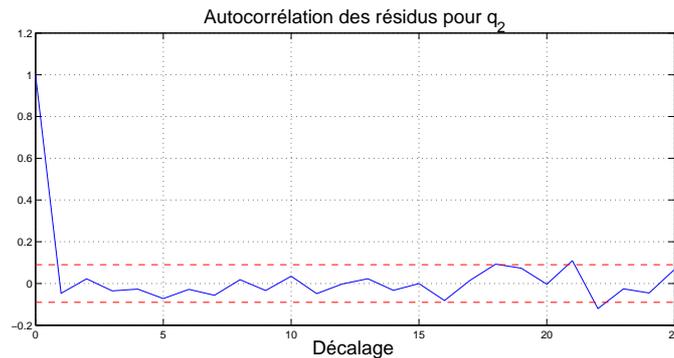
(a) Autocorrélation des résidus pour q_1 (b) Autocorrélation des résidus pour q_2

FIGURE 4.17 – Autocorrélation des résidus pour les flux du trafic d'entrée

Tests statistiques : Le tableau 4.5 résume les performances des modèles neuronaux. Les résultats de prévision par la méthode ARIMA sont également illustrés. Les mesures de la performance de prévision sont :

1. Erreur quadratique moyenne $MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$
2. Erreur quadratique moyenne normalisée $NMSE = \frac{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}{var(y_i)}$
3. Racine carrée de l'erreur quadratique moyenne $RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}}$
4. Erreur absolue moyenne $MAE = \frac{1}{n} \sum_{i=1}^n |(y_i - \hat{y}_i)|$
5. Erreur absolue maximale $MaxAE = \max_i |(y_i - \hat{y}_i)|$

$$6. \text{ Coefficient de corrélation } R = \frac{\sum_{i=1}^n (y_i - \frac{1}{n} \sum_{i=1}^n y_i)(\hat{y}_i - \frac{1}{n} \sum_{i=1}^n \hat{y}_i)}{\sqrt{\sum_{i=1}^n (y_i - \frac{1}{n} \sum_{i=1}^n y_i)^2} \sqrt{\sum_{i=1}^n (\hat{y}_i - \frac{1}{n} \sum_{i=1}^n \hat{y}_i)^2}}$$

où y_i et \hat{y}_i sont la valeur réellement observée et prédite pour le cycle i respectivement, n est le nombre d'exemples et $var(y_i)$ est la variance de y_i .

Flux d'entrée	Modèle	Indicateur d'erreur	Prévision				
			k+1	k+2	k+3	k+4	
q_1	Neuronal	MSE	0.591	1.517	4.255	8.259	
		NMSE	0.012	0.032	0.091	0.178	
		RMSE	0.769	1.321	2.062	2.873	
		MAE	0.567	0.845	1.326	1.863	
		MaxAE	4.597	6.196	15.443	15.177	
		R	0.993	0.983	0.953	0.911	
	ARIMA	RMSE	3.520				
		R	0.856				
	q_2	Neuronal	MSE	0.470	0.833	1.699	3.894
			NMSE	0.014	0.025	0.052	0.119
RMSE			0.686	0.912	1.303	1.973	
MAE			0.474	0.615	0.886	1.232	
MaxAE			3.998	8.241	6.212	16.510	
R			0.992	0.987	0.973	0.941	
ARIMA		RMSE	2.698				
		R	0.882				

TABLE 4.5 – Évaluation des performances des modèles

Le tableau indique que les précisions des réseaux neuronaux dans la prévision des 12 minutes futures (colonne $k + 4$) données par les indicateurs d'erreur RMSE et R sont meilleures que les précisions des premières valeurs future données par les modèles ARIMA. Cela peut être interprété par le fait que, contrairement aux modèles conventionnels statistiques ARIMA, les réseaux de neurones ont certaines propriétés inhérentes par lesquelles ils peuvent traiter la nature stochastique et la forte fluctuation des flux de trafic à l'intersection.

Sur ce, nous terminons notre aperçu général sur la prévision des flux de trafic par le modèle neuronal de type perceptron multi-couches pour passer au cas de la modélisation par des cartes auto-organisatrices.

4.7 Prévision des flux du trafic par les cartes auto-organisatrices

4.7.1 Introduction

La carte auto-organisatrice (SOM : Self-Organizing Map) ou carte de Kohonen est une technique de visualisation des données multidimensionnelles qui est inventée par le professeur Teuvo Kohonen [77], [78], [79]. Elle permet d'organiser automatiquement des données de grande dimension et de les visualiser par le biais d'une projection sur une carte 2D. Elle permet de décomposer un ensemble de données en sous-groupes similaires selon une fonction de similarité. De plus, elle permet de visualiser ces sous-groupes tout en conservant une information sur leur similarité. Ainsi, une carte auto-organisatrice permet simultanément de décomposer un ensemble de données, de l'organiser et de le visualiser.

La carte auto-organisatrice SOM est une architecture neuronale importante non supervisée qui, contrairement à celles supervisées, a été moins utilisée pour la prévision de séries temporelles y compris les séries temporelles de flux de trafic. Nous pensons que cela est dû au fait que la prévision de séries temporelles est généralement comprise comme un problème d'approximation de fonction, tandis que la SOM est généralement considérée comme un algorithme neuronal pour la quantification vectorielle de données, le regroupement (clustering) et la visualisation [80], [81]. Malgré cet handicap, l'utilisation de la SOM comme un prédicteur de séries temporelles est de plus en plus fréquent ces dernières années.

Notre but dans cette étude est de montrer que la SOM peut aussi être utilisée pour le problème de la prévision des flux d'entrée du trafic .

4.7.2 La carte auto-organisatrice (SOM)

La SOM est un algorithme d'apprentissage neuronal non supervisé bien connu. A partir d'exemples, la SOM apprend une projection entre un espace continu χ de grande dimension et un espace discret \mathcal{A} de dimension réduite (grille) de q neurones. Les neurones sont disposés physiquement aux nœuds d'un réseau dont les schémas de connexion sont les plus classiques : linéaire, rectangulaire, hexagonal (Figure 4.18). Chaque neurone est relié aux vecteurs d'entrée par des connexions (Figure 4.19). La fonction $i^*(\cdot) : \chi \rightarrow \mathcal{A}$, définie par la matrice de poids $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_q)$, $\mathbf{w}_i \in \mathbb{R}^p \subset \chi$, assigne au vecteur d'entrée $\mathbf{x}(t) \in \mathbb{R}^p$ un seul neurone telle que :

$$i^*(t) = \arg \min_{\forall_i} \{ \|\mathbf{x}(t) - \mathbf{w}_i(t)\| \} \quad (4.27)$$

où $\|\cdot\|$ désigne la distance euclidienne et t le pas de temps discret associé aux itérations de l'algorithme.

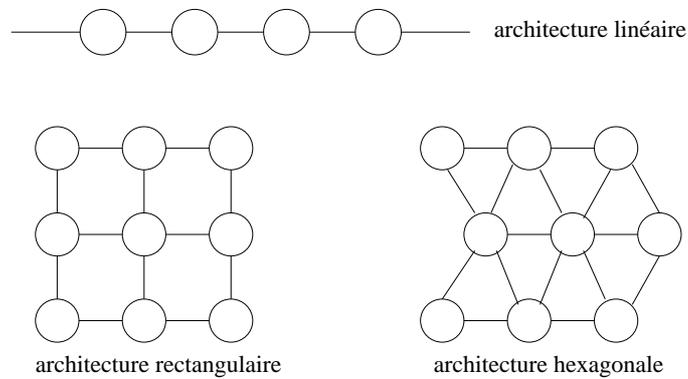


FIGURE 4.18 – Exemples d'architecture possibles

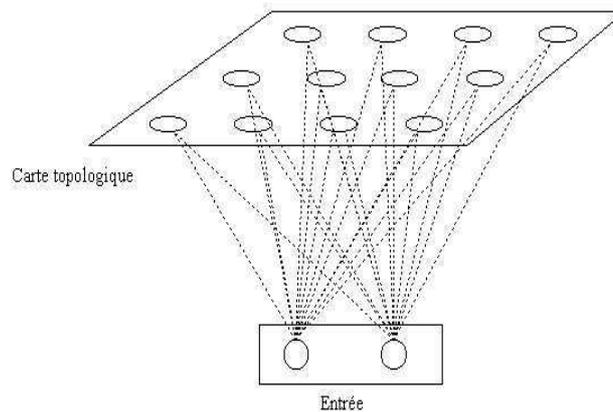


FIGURE 4.19 – La carte auto-organisatrice

Les vecteurs de poids, $\mathbf{w}_i(t)$, appelés aussi *vecteurs prototypes* ou encore *codebooks*, sont entraînés selon une règle d'apprentissage compétitif-coopératif dans laquelle les vecteurs de poids d'un neurone gagnant et ses voisins dans la grille sont mis à jour après la présentation de vecteur d'entrée :

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \alpha(t)h(i^*, i; t)[\mathbf{x}(t) - \mathbf{w}_i(t)] \quad (4.28)$$

où $0 < \alpha(t) < 1$ est le pas d'apprentissage et $h(i^*, i; t)$ est une fonction de voisinage qui pondère la variation des neurones de la carte en fonction de leur proximité du neurone gagnant. Un choix habituel pour $h(i^*, i; t)$ est donné par une fonction gaussienne autour du neurone gagnant et dont l'amplitude décroît au cours du temps. Cette fonction de voisinage permet une convergence rapide et une pondération plus fine des déplacements

des neurones suivant leur proximité du neurone gagnant. Elle s'écrit comme suit :

$$h(i^*, i; t) = \exp\left(-\frac{\|\mathbf{r}_i(t) - \mathbf{r}_{i^*}(t)\|^2}{2\sigma^2(t)}\right) \quad (4.29)$$

où $\mathbf{r}_i(t)$ et $\mathbf{r}_{i^*}(t)$ sont respectivement les coordonnées des neurones i et i^* dans la grille \mathcal{A} et $\sigma(t) > 0$ définit le rayon de la fonction de voisinage au temps t . En effet, plus un neurone est proche du neurone gagnant, plus son déplacement est important (voir figure 4.20).

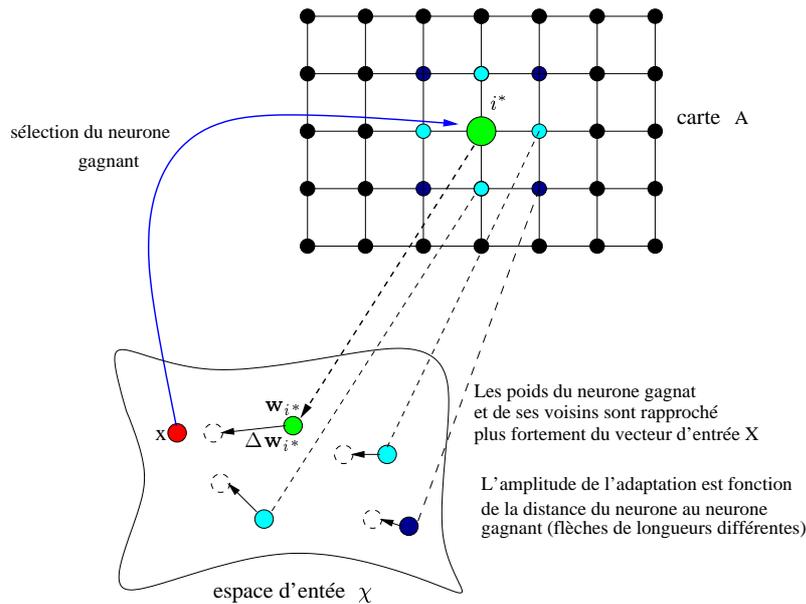


FIGURE 4.20 – Interprétation de l'algorithme SOM.

Les deux variables $\alpha(t)$ et $\sigma(t)$ doivent décroître en fonction du temps pour garantir la convergence des vecteurs de poids aux états stationnaires stables. Dans cette étude, nous adoptons pour les deux une décroissance exponentielle, donnée par :

$$\alpha(t) = \alpha_0 \left(\frac{\alpha_T}{\alpha_0}\right)^{(t/T)} \quad \text{et} \quad \sigma(t) = \sigma_0 \left(\frac{\sigma_T}{\sigma_0}\right)^{(t/T)} \quad (4.30)$$

où α_0, σ_0 (resp. α_T, σ_T) sont les valeurs initiales (resp. finales) des $\alpha(t)$ et $\sigma(t)$, et T désigne le nombre maximum d'itérations.

Cette étape de l'apprentissage est répétée jusqu'à ce que la variation des poids des neurones devient négligeable ou inférieure à un certain seuil ϵ . On utilise aussi comme critère de sécurité un nombre maximal d'itérations qui ne doit pas être dépassé. Ce nombre d'itérations doit être fixé à l'avance et choisi assez grand.

Malgré sa simplicité, l'algorithme SOM a été appliqué à une variété de problèmes complexes [82], [83], et est devenu l'une des architectures les plus importantes des réseaux de neurones.

4.7.2.1 La prédiction des séries temporelles par SOM

Considérons une série temporelle décrite par le modèle non linéaire auto-régressif d'ordre p :

$$x(t+1) = f[x(t), x(t-1), \dots, x(t-p+1)] \quad (4.31)$$

$$= f[\mathbf{x}^-] \quad (4.32)$$

où $\mathbf{x}^- \in \mathbb{R}^p$ est l'entrée ou le vecteur régresseur, $p \in \mathbb{Z}^+$ est l'ordre de la régression (ordre du modèle) et $f(\cdot)$ est une fonction non-linéaire de ses arguments. La structure de la fonction non-linéaire $f(\cdot)$ est inconnue. La seule chose connue est un ensemble d'observations : $x(1), x(2), \dots, x(N)$ où N est la longueur totale de la série temporelle. Notre tâche est de construire un modèle physique pour cette série temporelle. Pour atteindre cet objectif, nous utilisons une technique de modélisation neurale non supervisée, appelée le Vector-Quantized Temporal Associative Memory (VQTAM) [84], [85]. Basée sur cette technique, la carte auto-organisatrice (SOM) de Kohonen convient parfaitement pour rapprocher des relations d'entrée-sortie non-linéaires. La section suivante introduit la technique VQTAM et ses principales propriétés.

Le modèle VQTAM : Une façon tout à fait intuitive et directe d'utilisation de la SOM à des buts de prédiction de séries temporelles est d'effectuer une quantification vectorielle simultanément sur les vecteurs régresseurs $\mathbf{x}^-(t)$, $t = p, \dots, N-1$, et sur les observations d'un pas en avant associées $x(t+1)$. C'est exactement l'idée qui est derrière le Vector-Quantized Temporal Associative Memory (VQTAM) [84], [86]. Ce modèle peut être compris comme une généralisation au domaine temporel d'une technique de mémoire associative basée sur la SOM qui a été largement utilisée pour apprendre des relations d'entrée-sortie statiques (sans mémoire), particulièrement dans le domaine de la robotique [87].

Dans une formulation générale du modèle VQTAM, le vecteur d'entrée au pas de temps t , $\mathbf{x}(t)$, est composé de deux parties. La première partie, dénotée $\mathbf{x}^{in}(t) \in \mathbb{R}^p$ contient des données sur l'entrée de la relation dynamique à apprendre. La deuxième partie, dénotée $\mathbf{x}^{out}(t) \in \mathbb{R}^m$, contient des données concernant la sortie désirée de cette relation. La dimension du vecteur de poids de neurone i , $\mathbf{w}_i(t)$, augmente en conséquence. Ces changements sont formulés comme suit :

$$\mathbf{x}(t) = \begin{pmatrix} \mathbf{x}^{out}(t) \\ \mathbf{x}^{in}(t) \end{pmatrix} \quad \text{et} \quad \mathbf{w}_i(t) = \begin{pmatrix} \mathbf{w}_i^{out}(t) \\ \mathbf{w}_i^{in}(t) \end{pmatrix} \quad (4.33)$$

où $\mathbf{w}_i^{in}(t) \in \mathbb{R}^p$ et $\mathbf{w}_i^{out}(t) \in \mathbb{R}^m$ sont respectivement, les portions du vecteur de poids (prototype) qui stockent les informations sur les entrées et les sorties de la relation désirée.

Pour les tâches de la prévision des séries temporelles qui nous intéressent, les définitions suivantes s'appliquent :

$$\mathbf{x}^{out}(t) = x(t+1) \quad \text{et} \quad \mathbf{x}^{in}(t) = \mathbf{x}^-(t) \quad (4.34)$$

$$\mathbf{w}_i^{out}(t) = w_i^{out}(t) \quad \text{et} \quad \mathbf{w}_i^{in}(t) = \mathbf{w}_i^-(t) \quad (4.35)$$

où $w_i^{out}(t)$ est la valeur prédite et $\mathbf{w}_i^-(t) \in \mathbb{R}^p$ est la partie du vecteur de poids associée au vecteur régresseur $\mathbf{x}^-(t)$. Il est intéressant de noter que les vecteurs $\mathbf{x}^{out}(t)$ et $\mathbf{w}_i^{out}(t)$ dans l'équation (4.33) sont maintenant réduits respectivement aux quantités scalaires $x(t+1)$ et $w_i^{out}(t)$.

Pendant la phase d'apprentissage, le neurone gagnant (best-matching unit BMU, en anglais) au pas de temps t est déterminé uniquement sur $\mathbf{x}^{in}(t)$:

$$i^*(t) = \arg \min_{i \in A} \{ \|\mathbf{x}^{in}(t) - \mathbf{w}_i^{in}(t)\| \} = \arg \min_{i \in A} \{ \|\mathbf{x}^-(t) - \mathbf{w}_i^-(t)\| \} \quad (4.36)$$

Cependant, pour mettre à jour les poids, on utilise à la fois $\mathbf{x}^{in}(t)$ et $\mathbf{x}^{out}(t)$:

$$\mathbf{w}_i^{in}(t+1) = \mathbf{w}_i^{in}(t) + \alpha(t)h(i^*, i; t)[\mathbf{x}^{in}(t) - \mathbf{w}_i^{in}(t)] \quad (4.37)$$

$$w_i^{out}(t+1) = w_i^{out}(t) + \alpha(t)h(i^*, i; t)[\mathbf{x}^{out}(t) - w_i^{out}(t)] \quad (4.38)$$

Autrement dit, l'équation (4.37) effectue la préservation de la topologie de la quantification vectorielle sur l'espace d'entrée et l'équation (4.38) agit de manière similaire sur l'espace de sortie de la relation apprise. Vu la procédure d'apprentissage, la SOM apprend à associer les vecteurs prototypes d'entrée \mathbf{w}_i^{in} avec les vecteurs prototypes correspondants de la sortie w_i^{out} .

Pendant la phase de validation, le neurone gagnant $i^*(t)$ est trouvé pour chaque vecteur d'entrée $\mathbf{x}^{in}(t)$, et l'estimation à un pas en avant est alors donnée par :

$$\hat{x}(t+1) = w_{i^*}^{out} \quad (4.39)$$

Il est intéressant de souligner la différence entre la stratégie VQTAM et l'approche supervisée habituelle. En effet, dans les réseaux PMC, le vecteur $\mathbf{x}^{in}(t)$ est présenté à l'entrée du réseau, tandis que le $x^{out}(t)$ est utilisé à la sortie du réseau pour calculer explicitement un signal d'erreur qui guide l'apprentissage (figure 4.21a). Le schéma de VQTAM permet aux réseaux neuronaux non supervisés, tels que la SOM, d'apprendre à associer ou à corrélérer les entrées et sorties d'une relation sans calcul explicite d'un signal d'erreur (figure 4.21b).

Dans la section suivante, nous évaluons la performance de la SOM dans le problème de la prévision à un pas en avant des flux du trafic, ensuite nous comparons les résultats avec les résultats obtenus par les réseaux PMC et par les modèles ARIMA.

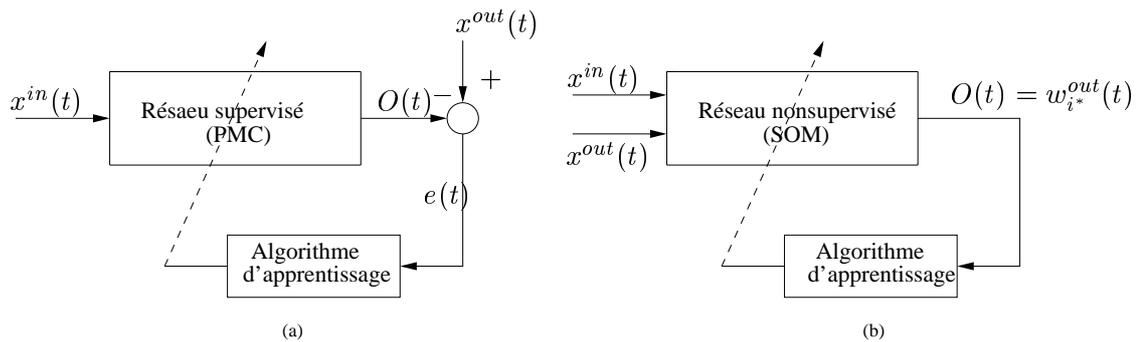


FIGURE 4.21 – Différences entre (a) l’approche supervisée et (b) l’approche non-supervisée

4.7.2.2 La performance du modèle SOM pour la prédiction des flux de trafic

Pour construire les modèles SOM, nous utilisons les mêmes données du trafic que nous avons utilisées pendant la modélisation supervisée (voir section 4.6.2). Les ordres des deux modèles sont choisis respectivement comme dans le tableau 4.1. Pour chaque modèle, une SOM à deux dimensions avec 400 neurones (20*20) a été entraînée pour 600 itérations. Nous justifions ce choix par le fait que si nous utilisons une très grande quantité de vecteurs prototypes, nous perdons la compacité qui est atteinte par le processus de quantification et par conséquent, le processus de prédiction devient plus lourd au niveau du calcul. Les poids sont initialisés aléatoirement entre 0 et 1. D’autres gammes d’initialisation aléatoire, par exemple entre -1 et 1, ont été testées sans aucune influence significative sur la configuration finale des poids.

La fonction de voisinage a été choisie comme la fonction gaussienne. Les paramètres d’apprentissage sont les suivants : $\alpha_0 = 1.0$, $\alpha_T = 10^{-5}$, $\sigma_0 = 200$ et $\sigma_T = 10^{-3}$. L’apprentissage des modèles neuronaux a été réalisé en utilisant la boîte à outils Matlab appelée SOM Toolbox [88].

Les performances de prédiction ont été mesurées en calculant la racine carrée de l’erreur quadratique moyenne $RMSE$ et le coefficient de corrélation R pour la prédiction à un pas en avant (3 minutes) de flux de trafic q_1 et q_2 . Les résultats des prévisions sont indiqués dans le tableau 4.6.

Flux d’entrée	Indicateur d’erreur	
	RMSE	R
q_1	2.144	0.949
q_2	1.463	0.967

TABLE 4.6 – Evaluation des performances des modèles SOM

En comparant ces résultats avec ceux du tableau 4.5, on constate que les résultats produits par la SOM sont moins précis que ceux produits par le réseau PMC, mais meilleurs que ceux produits par le modèle ARIMA.

Bien que les modèles PMC produisent des résultats intéressants, il n'en demeure pas moins que les modèles SOM sont des outils précieux pour la prévision des données du trafic car ils sont moins sensibles à l'initialisation de poids que les réseaux PMC et ne souffrent pas du sur-ajustement (overfitting).

4.8 Application de la commande (A, B) -invariance

Les détecteurs à boucle d'induction, qui sont les principales sources de mesure des flux des véhicules, fournissent des informations sur les flux d'entrées pour le cycle actuel ou au mieux pour le cycle suivant. Par conséquent, le contrôleur ne peut pas calculer la loi de commande en temps réel à cause de la limitation du temps de calcul. Pour ce faire, il doit être capable de prédire les valeurs futures de ces flux. D'où l'idée d'incorporer les réseaux de neurones dans le contrôleur pour réaliser cet objectif.

Ainsi, en intégrant l'approche neuronale dans notre système de commande, la procédure du calcul de la loi de commande pour chaque cycle peut être décrite comme suit :

1. Estimer plusieurs valeurs futures des flux d'entrées q_i par les réseaux de neurones.
2. Utiliser ces flux pour calculer la commande nominale u_N .
3. Trouver un gain G s'il existe, qui satisfait les contraintes sur la commande choisie.
4. Calculer la loi de commande $u(k)$ selon sa structure définie.

Notons que le modèle neuronal fournit les prévisions de q_i de 3 minutes à l'avance (c'est-à-dire le nombre de véhicules entrants pendant 2 cycles futurs). Ainsi, q_i est considéré constant pendant deux cycles.

Sur la figure 4.22 est représentée l'évolution des files d'attente des deux phases pour la période de 7h00 jusqu'à 20h00. Nous remarquons d'abord que l'ensemble de la période considérée inclut des moments de circulation à forte demande (les heures de pointes) et des phases de trafic fluide. En plus, quelque soit la période, lorsque la circulation n'est pas en excès, la stratégie de commande arrive à maintenir les longueurs des files d'attente inférieure à leurs valeurs critiques.

En outre, le tableau 4.7 présente les valeurs des flux d'entrée (q_1, q_2) ainsi que les variations du vert effectif g_{e_2} en fonction du cycle pendant la période de 7h04 jusqu'à 9h04 du fonctionnement de l'intersection.

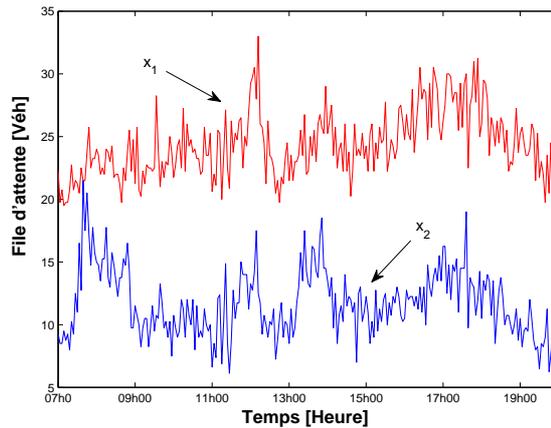


FIGURE 4.22 – Évolution des files d'attente.

Période de 7h04mn à 9h04mn										
cycles	1	2	3	4	5	6	7	8	9	10
q_1	0,15	0,15	0,15	0,15	0,17	0,17	0,15	0,15	0,15	0,15
q_2	0,20	0,20	0,20	0,20	0,17	0,17	0,20	0,20	0,20	0,20
g_{e_2}	44,66	44,67	44,68	44,69	37,93	37,94	44,71	44,72	44,72	44,73
cycles	11	12	13	14	15	16	17	18	19	20
q_1	0,16	0,16	0,15	0,15	0,15	0,15	0,15	0,15	0,18	0,18
q_2	0,17	0,17	0,20	0,20	0,20	0,20	0,20	0,20	0,14	0,14
g_{e_2}	39,75	39,75	44,74	44,75	44,75	44,76	44,76	44,77	33,17	33,17
cycles	21	22	23	24	25	26	27	28	29	30
q_1	0,2	0,2	0,22	0,22	0,23	0,23	0,2	0,2	0,3	0,3
q_2	0,3	0,3	0,33	0,33	0,27	0,27	0,33	0,33	0,2	0,2
g_{e_2}	54,78	54,79	53,08	53,09	48,07	48,08	54,81	54,82	41,62	41,62
cycles	31	32	33	34	35	36	37	38	39	40
q_1	0,2	0,2	0,21	0,21	0,24	0,24	0,21	0,21	0,15	0,15
q_2	0,33	0,33	0,32	0,32	0,25	0,25	0,23	0,23	0,3	0,3
g_{e_2}	44,82	44,82	44,83	44,83	41,54	41,55	33,22	33,22	34,49	34,50
cycles	41	42	43	44	45	46	47	48	49	50
q_1	0,17	0,17	0,17	0,17	0,16	0,16	0,18	0,18	0,15	0,15
q_2	0,16	0,16	0,16	0,16	0,17	0,17	0,15	0,15	0,20	0,20
g_{e_2}	36,47	36,47	36,48	36,48	39,88	39,88	34,85	34,86	44,88	44,88
cycles	51	52	53	54	55	56	57	58	59	60
q_1	0,15	0,15	0,17	0,17	0,15	0,15	0,17	0,17	0,15	0,15
q_2	0,19	0,19	0,16	0,16	0,20	0,20	0,16	0,16	0,20	0,20
g_{e_2}	43,20	43,20	36,52	36,52	44,90	44,90	36,53	36,54	44,91	44,91
cycles	61	62	63	64	65	66	67	68	69	70
q_1	0,15	0,15	0,15	0,15	0,15	0,15	0,16	0,16	0,16	0,16
q_2	0,19	0,19	0,20	0,20	0,20	0,20	0,18	0,18	0,18	0,18
g_{e_2}	43,23	43,23	44,91	44,92	44,92	44,92	41,61	41,61	41,61	41,61
cycles	71	72	73	74	75	76	77	78	79	80
q_1	0,17	0,17	0,18	0,18	0,15	0,15	0,15	0,15	0,17	0,17
q_2	0,17	0,17	0,14	0,14	0,20	0,20	0,20	0,20	0,17	0,17
g_{e_2}	38,24	38,24	33,28	33,28	44,93	44,93	44,93	44,94	38,25	38,25

TABLE 4.7 – Variation du feu vert effectif en fonction du cycle pendant la période de 7h04 jusqu'à 9h04.

Nous remarquons à partir de ce tableau que la variation de g_{e_2} est proportionnelle à la variation de q_2 . Ceci s'explique par l'adaptation de la commande par rapport à la situation réelle du trafic, à savoir le contrôleur privilégie d'allouer plus de temps vert à la ligne dont le flux d'entrée est le plus grand. Ce fait est illustré d'avantage dans la plage "cycle 20–cycle 30 qui correspond à l'heure de pointe.

En résumé, l'analyse de l'ensemble des résultats montre que l'incorporation des modèles neuronaux dans le système de commande constitue un excellent outil pour éviter la congestion. D'une part, la prévision des flux d'entrées du carrefour permet à la commande de s'adapter aux variations temporelles du trafic. D'autre part, la commande se prête mieux à une implémentation temps réel.

4.9 Conclusion

Ce dernier chapitre avait pour objectif d'enrichir les stratégies de commandes proposées dans les chapitres précédents par l'estimation des flux d'entrées des carrefours. Dans ce sens, la technique des réseaux de neurones a été utilisée. Deux modèles neuronaux ont été proposés. D'abord le perceptron multi-couches (PMC) combiné avec un vecteur de régression temporel NNAR (Neural Network Auto Regressive). Ensuite la carte auto-organisatrice SOM (Self-Organizing Map). Ces deux modèles ont été appliqués sur un carrefour réel et ont conduit à des résultats satisfaisants comparativement aux différentes approches traditionnelles de prévision telles que les modèles ARIMA. Cependant, les indicateurs d'erreurs ont montré que le modèle PMC est relativement précis par rapport au modèle SOM. Néanmoins, le modèle SOM a l'avantage d'être moins sensible à l'initialisation des poids que le réseau PMC et ne souffre pas du sur-ajustement (overfitting).

Enfin, l'utilisation des résultats de la prévision des flux d'entrée par la commande (A, B) -invariance a permis, d'une part, d'atteindre notre objectif concernant le contrôle a priori des carrefours et, d'autre part, de valider le fait que le stratégie de commande est tout à fait apte à répondre à la problématique du temps réel.

CONCLUSION GÉNÉRALE

Dans ce travail, nous avons présenté un ensemble de contributions portant sur le contrôle a priori de congestion des carrefours signalisés. Ces systèmes dynamiques sont sujets à des limitations d'ordre physique et d'autres dictées par des impératifs de sécurité. Ces limitations se traduisent dans la majorité des cas par des contraintes sur les variables d'état et/ou la commande. La prise en compte de telles contraintes dans la conception du plan de feux constitue l'une des orientations essentielles des travaux présentés dans ce mémoire.

La technique utilisée est basée sur l'approche par invariance positive. L'idée maîtresse de cette approche est de contraindre l'état du système à évoluer dans un domaine inclus dans le domaine défini par les contraintes. Cette technique nous a permis de traiter le problème de la commande de deux cas importants de carrefours. Celui pour lequel le cycle des feux est constant et celui où le cycle des feux est variable afin d'adapter la stratégie de commande de manière efficace aux variations permanentes du trafic.

Dans le cadre du problème de satisfaction des contraintes, deux outils ont été utilisés :

Le premier fait appel aux inégalités matricielles Linéaires (LMIs). Nous avons exploité cet outil pour arriver à des résultats théoriques dont la vérification permet de conclure sur la possibilité de confiner l'état du système à l'intérieur d'un domaine ellipsoïdal contenu dans l'ensemble des contraintes, au moyen d'une action de la commande adéquate. L'existence de ce domaine ellipsoïdal est caractérisée par la faisabilité de certaines inégalités matricielles linéaires dont les solutions s'avèrent relativement simples à mettre en œuvre et non coûteuses du point de vue temps de calcul. Cependant, malgré l'exis-

tence d'outils d'optimisation pour la recherche du plus grand ellipsoïde contenu dans un polyèdre, l'outil LMI ne permet pas d'étudier directement certains cas. En effet, les cas limites difficiles comme ceux des contraintes ayant des frontières qui passent par l'origine (comme c'est le cas des carrefours signalisés) nécessitent un changement de variable adéquat. En outre, dans le cas où les contraintes sont très dissymétriques, le domaine Ω sera très insignifiant autour de l'origine.

Pour remédier aux inconvénients de l'approche LMI, nous avons dû avoir recours à l'approche (A, B) -invariance pour la solution de notre problème de commande. Nous avons tiré profit de cet outil pour développer des résultats assurant l'invariance positive en boucle fermée de l'ensemble des contraintes sur l'état au lieu de construire un domaine ellipsoïdal contenu dans cet ensemble. Le test d'existence et le calcul d'une loi de commande par retour d'état peut alors se faire de façon simple par la résolution d'un problème de programmation linéaire. L'avantage de cet outil est d'inclure l'ensemble des trajectoires se trouvant dans le champ des contraintes. Un autre avantage se fait sentir notamment dans le calcul de la commande sans qu'elle ne soit conditionnée par l'existence d'un domaine ellipsoïdal. Cette méthode nous a permis en outre d'intégrer directement les contraintes sur la commande dans la synthèse du régulateur.

La prévision à court terme des paramètres de trafic, tels que le flux d'entrée, est un élément crucial pour les systèmes de commande du trafic en temps réel. Pour résoudre ce problème, nous avons proposé une approche basée sur la technique des réseaux de neurones pour estimer l'entrée d'un carrefour réel. Dans ce sens, deux modèles neuronaux ont été développés. D'abord le perceptron multi-couches. Les différentes étapes de la procédure de création d'un tel modèle neuronal ont été ensuite détaillées. Le deuxième modèle est basé sur l'utilisation de la carte auto-organisatrice SOM (Self-Organizing Map). Dans ce sens, en utilisant l'approche VQTAM (Vector-Quantized Temporal Associative Memory), nous avons montré que l'algorithme d'apprentissage neuronal non supervisé SOM convient parfaitement à la modélisation des séries temporelles. Les simulations ont montré que les deux modèles dégagent des performances intéressantes comparativement aux différentes approches traditionnelles de prévision telles que les modèles ARIMA. Ces simulations soulignent aussi un léger avantage pour les perceptrons multi-couches PMC par rapport aux cartes auto-organisatrices SOM. Nous avons montré également que l'incorporation des modèles neuronaux dans le système de commande constitue un excellent outil pour éviter la congestion. D'une part, la prévision des flux d'entrées du carrefour

permet à la commande de s'adapter aux variations temporelles du trafic. D'autre part, la commande se prête mieux à une implémentation temps réel.

Perspectives

Si les résultats de ce travail ont montré l'importance et l'utilité du concept de l'invariance positive pour résoudre le problème de la régulation du trafic au sein d'une intersection signalisée, il serait intéressant de voir les possibilités qu'offre cette approche pour étudier l'influence des intersections adjacentes. On se retrouve alors devant le problème de la régulation d'un réseau de transport global. Ce système, de part sa nature, est complexe et impose une synchronisation entre les sous-systèmes. En effet, chaque intersection doit être pilotée par sa propre loi de commande tout en tenant compte de l'interconnexion avec les autres intersections. Le problème ainsi défini revient au problème de la régulation des systèmes inter-connectés sous contraintes. L'utilisation de cette approche pour ce type de systèmes n'est pas systématique. Elle nécessite alors un réel travail de développement afin de l'adapter à notre objectif.

En ce qui concerne la prévision des débits d'entrée, nous pouvons citer les perspectives suivantes :

- Développer un modèle neuronal qui englobe non seulement les caractéristiques temporelles mais aussi les caractéristiques spatiales du flux de trafic. L'idée est d'utiliser la modélisation multi-échelles où les modèles seront développés en utilisant les données de l'emplacement d'intérêt ainsi que les emplacements en amont (flux de trafic à partir des détecteurs situés en amont du point de mesure). Nous estimons que cette influence spatiale peut contribuer à améliorer les performances prédictives des paramètres du système ainsi que la stratégie de commande.
- Mettre en œuvre d'autres techniques d'intelligence artificielle en combinaison avec les réseaux de neurones comme par exemple, les algorithmes génétiques et la logique floue. Cette combinaison n'est évidemment pas systématique. Elle mérite aussi un réel travail de développement et pourrait améliorer la fiabilité et la précision de la commande.

ANNEXE A

SUR LA SOLUTION POSITIVE DE L'ÉQUATION $BU = 0$

Considérons le problème de trouver un vecteur positif $u \neq 0$ tel que :

$$Bu = 0 \tag{A.1}$$

où $B \in \mathbb{R}^{n \times n}$. On sait dans le cours de l'algèbre linéaire que l'existence d'un tel vecteur u dépend fortement du rang de la matrice B . Si $\text{rang}(B) = n$, alors la matrice B est non-singulière. Par conséquent, aucun vecteur $u \neq 0$ solution de $Bu = 0$ ne peut exister. C'est la raison pour laquelle nous allons supposer, dans la suite, que $\text{rang}(B) < n$.

Notons que la solvabilité de (A.1) avec $u \in \mathbb{R}^n$ est bien connue dans le cours de l'algèbre linéaire par l'utilisation de la méthode de Cramer ou la décomposition en valeurs singulières. Alors, le sérieux problème que pose la solvabilité de (A.1) est de prouver l'existence d'un vecteur positif et, le cas échéant, donner une forme explicite de la solution. C'est l'objectif de ce qui va suivre.

Considérons maintenant le système (A.1) où $\text{rank}(B) = r$. On sait que le cofacteur d'un élément b_{ij} est le déterminant de la sous-matrice obtenue en éliminant la $i^{\text{ième}}$ ligne et la $j^{\text{ième}}$ colonne de B . La matrice des cofacteurs sera notée $\overset{\circ}{B}$. La matrice adjointe de la matrice B , notée $\text{adj}(B)$, est la matrice transposée de la matrice des cofacteurs : $\text{adj}(B) = \overset{\circ}{B}^t$. La proposition suivante donne une condition suffisante d'existence de solution positive du système (A.1).

Proposition A.1 *Une condition suffisante pour que le système (A.1) admette une solution positive est que la matrice $\text{adj}(B)$ soit positive ou négative.*

Preuve: Puisque B est singulière et $\text{adj}(B) \neq 0$, alors chaque déterminant B_{ij} d'ordre $n - 1$ obtenu de la matrice B en biffant la $i^{\text{ième}}$ ligne et la $j^{\text{ième}}$ colonne est différent de zéro. Ainsi, $\text{rang}(B) = n - 1$. Cela posé, il est bien connu que [47] pour une matrice dont $\text{rang} = n - 1$, la matrice $\text{adj}(B) = \alpha xy^t$, où $\alpha \neq 0$ est une constante, $x \neq 0$ et $y \neq 0$ tels que $Bx = 0$ et $B^t y = 0$. Ainsi, puisque $\text{adj}(B)$ est positive ou négative, alors deux cas sont possibles :

Cas 1 : $\text{adj}(B) > 0$: dans ce cas, soit $\alpha > 0$ ou $\alpha < 0$. Si $\alpha > 0$ alors $xy^t > 0$, ce qui entraîne $x > 0$ et $y > 0$ ou $x < 0$ et $y < 0$. Ainsi, il suffit de prendre $u = x$ dans le premier cas ou $u = -x$ dans le deuxième pour obtenir $u > 0$ avec $Bu = 0$. Si maintenant $\alpha < 0$ alors $xy^t < 0$ et le reste de la preuve est similaire au cas où $\alpha > 0$.

Cas 2 : $\text{adj}(B) < 0$: la preuve est identique au cas 1. ■

Il faut noter explicitement que le test de cette proposition ne dit rien sur la manière de construire les vecteurs positifs solutions du système (A.1). C'est la raison pour laquelle dans la suite on s'attache à développer des critères qui prouvent non seulement l'existence de tels vecteurs et dont l'application et l'implémentation dans un environnement logiciel restent abordables en temps de calcul, mais aussi de construire explicitement ces vecteurs.

Considérons le système (A.1) avec $\text{rang}(B) = n - 1$. Sans restreindre la généralité, supposons que les premières $n - 1$ colonnes de C sont linéairement indépendantes (il est toujours possible de permuter les colonnes de B pour satisfaire cette supposition). Soit maintenant :

$$\widehat{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1,n-1} \\ b_{21} & c_{22} & \cdots & b_{2,n-1} \\ \vdots & \vdots & \cdots & \vdots \\ b_{n,1} & b_{n,2} & \cdots & b_{n,n-1} \end{pmatrix},$$

la sous-matrice de B formée par ses premières $n - 1$ colonnes. Pour chaque $i = 1, 2, \dots, n$, notons \widehat{B}_i le déterminant d'ordre $n - 1$ obtenu à partir de \widehat{B} en biffant la $i^{\text{ième}}$ ligne. La proposition suivante donne une deuxième caractérisation de l'existence d'un vecteur positif solution de (A.1) et une manière explicite de sa construction.

Proposition A.2 *Une condition nécessaire et suffisante pour que le système (A.1) admette une solution positive est que :*

$$\widehat{B}_i \widehat{B}_{i+1} < 0, \quad \forall 1 \leq i \leq n - 1. \quad (\text{A.2})$$

Preuve: La condition est nécessaire. En effet, si le système (A.1) admet une solution positive alors il existe un vecteur $u > 0$ tel que $Bu = 0$. Ceci implique que :

$$\sum_{j=1}^n b_{ij}u_j = 0, \quad \forall 1 \leq i \leq n. \quad (\text{A.3})$$

Or, puisque $\text{rank}(B) = n - 1$, alors la solution de (A.3) est connue sous la forme [?]

$$u_i = (-1)^{i-1} \rho \widehat{B}_i, \quad \forall 1 \leq i \leq n, \quad (\text{A.4})$$

où $\rho \neq 0$ est une constante. Ainsi, puisque $u_i > 0$ pour tout $1 \leq i \leq n$, alors l'inégalité $u_i u_{i+1} > 0$ implique $-\rho^2 \widehat{B}_i \widehat{B}_{i+1} > 0$. Par conséquent (A.2) est satisfaite.

Pour montrer la suffisance, remarquons d'abord qu'à partir de (A.2) nous avons $\widehat{B}_i \neq 0$ pour tout $1 \leq i \leq n$. Soit maintenant :

$$y_i = |\widehat{B}_i|, \quad \forall 1 \leq i \leq n. \quad (\text{A.5})$$

Il est clair que $y_i > 0$. Cela posé, nous allons montrer à présent que les y_i définis par (A.5) sont solutions de (A.1). En effet, à partir de (A.2), \widehat{B}_i et \widehat{B}_{i+1} ont des signes opposés, ainsi deux cas se présentent :

Cas 1 : $\widehat{B}_i > 0$ et $\widehat{B}_{i+1} < 0$ pour tout $1 \leq i \leq n - 1$: dans ce cas, il est clair que $\widehat{B}_1 > 0$ et par conséquent $\widehat{B}_i > 0$ si i est impair et $\widehat{B}_i < 0$ si i est pair. Par ailleurs, à partir de (A.3) et (A.4) nous avons :

$$\sum_{j=1}^n (-1)^{i-1} \rho \widehat{B}_j b_{ij} = 0, \quad \forall 1 \leq i \leq n$$

ou d'une manière équivalente :

$$\widehat{B}_1 b_{i1} - \widehat{B}_2 b_{i2} + \widehat{B}_3 b_{i3} - \cdots \pm \widehat{B}_n b_{in} = 0, \quad \forall 1 \leq j \leq n, \quad (\text{A.6})$$

ce qui implique :

$$\sum_{j=1}^n b_{ij} |\widehat{B}_j| = 0, \quad \forall 1 \leq i \leq n, \quad (\text{A.7})$$

ce qui montre que (A.5) est la solution de (A.1). Par conséquent, il existe un vecteur $y > 0$, dont les composantes sont définies par (A.5), tel que $By = 0$.

Cas 2 : $\widehat{B}_i < 0$ et $\widehat{B}_{i+1} > 0$ pour tout $1 \leq i \leq n - 1$: dans ce cas, nous avons $\widehat{B}_i < 0$ si i est impair et $\widehat{B}_i > 0$ si i est pair. Le reste de la preuve est similaire au cas 1 excepté qu'il faut multiplier l'équation (A.6) par -1 pour arriver à l'équation (A.7). ■

Remarque A.1 Il est important de noter que cette proposition donne une manière explicite de construire des vecteurs positifs solution du système (A.1) (voir équation (A.5)) et dont l'application et l'implémentation dans un environnement logiciel restent abordables en temps de calcul. ►

ANNEXE B

LA RÉTRO-PROPAGATION DU GRADIENT

L'algorithme de rétro-propagation du gradient est une technique très populaire pour ajuster les poids dans un réseau de neurones multicouches. Elle a été développée par David Rumelhart, Geoff Hinton et R. J. Williams en 1986 [89]. Le terme rétro-propagation du gradient provient du fait que l'erreur calculée en sortie est transmise en sens inverse vers l'entrée. L'algorithme de rétro-propagation du gradient commence, nécessairement avec le calcul de la couche de sortie, qui est la seule couche où les sorties désirées sont disponibles. L'algorithme suit les étapes suivantes : (voir Figure B.1)

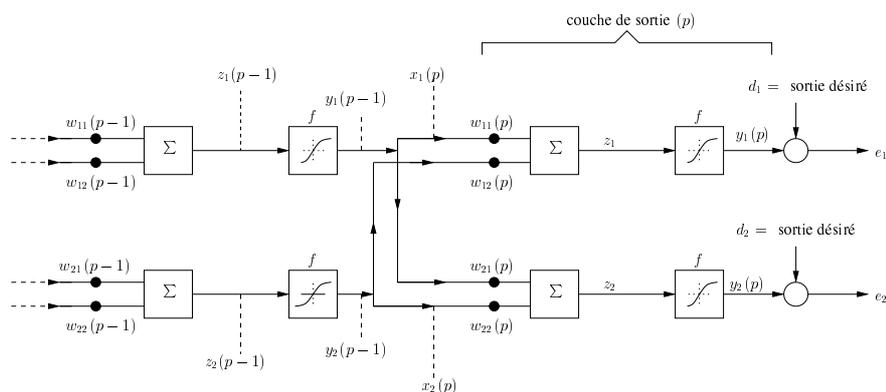


FIGURE B.1 – Un perceptron multi-couches

Mise à jour des poids de la couche de sortie : Considérons un neurone k de la couche de sortie. Lorsqu'un motif lui est présenté, il produit la sortie y_k alors que c'est la valeur d_k qui est attendue. Notons :

$$e_k = d_k - y_k \quad (\text{B.1})$$

l'erreur associée à ce neurone. L'erreur globale du réseau est donnée par :

$$E(w) = \frac{1}{2} \sum_{k=1}^N e_k^2 = \frac{1}{2} \sum_{k=1}^N (d_k - y_k)^2 \quad (\text{B.2})$$

N étant le nombre de neurones dans la couche de sortie. Ainsi, le gradient de E est :

$$\nabla E_k = \frac{\partial E}{\partial w_{kj}} \quad (\text{B.3})$$

A partir de l'algorithme du gradient, nous avons :

$$w_{kj}^{(n+1)} = w_{kj}^{(n)} + \Delta w_{kj}^{(n)} \quad (\text{B.4})$$

où j est la j ème entrée au k ème neurone de la couche de sortie, et :

$$\Delta w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}} \quad (\text{B.5})$$

Le signe moins (-) dans (B.5) indique une direction de descente vers un minimum. Soit :

$$z_k = \sum_j w_{kj} x_j \quad (\text{B.6})$$

l'activité interne (potentiel) du neurone de sortie k . x_j est la j ème entrée de ce neurone. Sa sortie s'écrit alors sous la forme :

$$y_k = f(z_k) \quad (\text{B.7})$$

f est la fonction d'activation. Pour calculer la dérivée partielle de E , il faudra faire appel à la règle de chaînage des dérivées.

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial z_k} \frac{\partial z_k}{\partial w_{kj}} \quad (\text{B.8})$$

D'après (B.6) on peut écrire :

$$\frac{\partial z_k}{\partial w_{kj}} = x_j(p) = y_j(p-1) \quad (\text{B.9})$$

p dénote la couche de sortie. Ainsi, (B.8) devient :

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial z_k} x_j(p) = \frac{\partial E}{\partial z_k} y_j(p-1) \quad (\text{B.10})$$

Définissons :

$$\delta_k(p) = -\frac{\partial E}{\partial z_k} \quad (\text{B.11})$$

Alors (B.10) donne :

$$\frac{\partial E}{\partial w_{kj}} = -\delta_k(p)x_j(p) = -\delta_k(p)y_j(p-1) \quad (\text{B.12})$$

Il s'ensuit à partir de (B.5) et (B.12) que :

$$\Delta w_{kj} = \eta \delta_k(p)x_j(p) = \eta \delta_k(p)y_j(p-1) \quad (\text{B.13})$$

où j est la $j^{\text{ème}}$ entrée au neurone k de la couche de sortie (p). Par ailleurs, (B.11) peut s'écrire sous la forme :

$$\delta_k = -\frac{\partial E}{\partial z_k} = -\frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial z_k} \quad (\text{B.14})$$

Mais, par (B.2) :

$$\frac{\partial E}{\partial y_k} = -(d_k - y_k) = y_k - d_k \quad (\text{B.15})$$

Ainsi, pour une fonction sigmoïde :

$$y_k = f(z_k) = \frac{1}{1 + \exp(-z_k)} \quad (\text{B.16})$$

nous avons :

$$\frac{\partial y_k}{\partial z_k} = y_k(1 - y_k) \quad (\text{B.17})$$

Par conséquent, d'après (B.14), (B.15) et (B.17) :

$$\delta_k = y_k(1 - y_k)(d_k - y_k) \quad (\text{B.18})$$

La règle de mise à jour des poids de la couche de sortie est donc finalement :

$$\Delta w_{kj} = \eta \delta_k(p)y_j(p-1) \quad (\text{B.19})$$

où δ_k est définie par (B.18).

Mise à jour des poids des couches cachées : En ce qui concerne les autres couches, on ne peut pas utiliser (B.19) pour la modification des poids des neurones cachés, voyons alors quelle va être la nouvelle équation de mise à jour des poids. Pour cela, considérons un neurone j sur la couche cachée $r = (p-1)$. Nous avons toujours, comme dans (B.5) :

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} \quad (\text{B.20})$$

pour la $i^{\text{ème}}$ branche au $j^{\text{ème}}$ neurone. Par conséquent, d'après (B.8) :

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial z_j} \frac{\partial z_j}{\partial w_{ji}} \quad (\text{B.21})$$

Il s'ensuit à partir de (B.9) et (B.11) que :

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial z_j} y_i(r-1) = \eta \delta_j(r) y_i(r-1) \quad (\text{B.22})$$

de plus avec (B.14) :

$$\Delta w_{ji} = -\eta \left[\frac{\partial E}{\partial y_j(r)} \frac{\partial y_j(r)}{\partial z_j} \right] y_i(r-1) \quad (\text{B.23})$$

Par la règle de chaînage des dérivées :

$$\frac{\partial E}{\partial y_j(r)} = \sum_k \frac{\partial E}{\partial z_k(r+1)} \left[\frac{\partial z_k(r+1)}{\partial y_j(r)} \right] = \sum_k \frac{\partial E}{\partial z_k} \left[\frac{\partial}{\partial y_j(r)} \sum_m w_{km}(r+1) y_m(r) \right] \quad (\text{B.24})$$

où la somme sur k est réalisée sur les neurones de la couche suivante ($r+1$) qui se connectent à $y_j(r)$, tandis que la somme sur m est sur toutes les entrées à chaque neurone k de la couche ($r+1$).

Par conséquent, et d'après la définition de δ , (B.24) donne :

$$\frac{\partial E}{\partial y_j(r)} = \sum_k \frac{\partial E}{\partial z_k(r+1)} w_{kj} = - \sum_k \delta_k(r+1) w_{kj}(r+1) \quad (\text{B.25})$$

puisque $w_{kj}(r+1)$ est relié uniquement à $y_j(r)$. Par conséquence, d'après (B.14), (B.17) et (B.25) nous avons :

$$\delta_j(r) = \frac{\partial y_j}{\partial z_j(r)} \sum_k \delta_k(r+1) w_{kj}(r+1) \quad (\text{B.26})$$

$$= y_j(r) [1 - y_j(r)] \sum_k \delta_k(r+1) w_{kj}(r+1) \quad (\text{B.27})$$

et, via (B.19) :

$$\Delta w_{ji}(r) = \eta \delta_j(r) y_i(r-1) \quad (\text{B.28})$$

Ainsi, l'erreur de sortie est rétro-propagée vers l'entrée (c'est-à-dire dans le sens inverse de celui utilisé pour propager un signal au travers d'un réseau) afin de corriger de couche en couche les valeurs des poids. Son calcul peut donc être résumé comme suit :

Présenter le premier stimulus. Par la suite, calculer $\Delta w_{kj}(p)$ pour la couche de sortie (la couche p) à partir de (B.18) et (B.19), et ensuite passer au calcul de $\Delta w_{ji}(p)$ à partir de (B.28) pour $r = p-1, p-2, \dots, 2, 1$; en utilisant (B.26) pour la mise à jour de $\delta_j(r)$ sur la base de $\delta_j(r+1)$ amont (rétropropagation de la couche $r+1$ à la couche r), etc. Ensuite, mettre à jour $w^{(n+1)}$ à partir de $w^{(n)}$ et $\Delta w^{(n)}$ pour l'itération $(n+1)$ par (B.4). Répéter tout le processus lors de la présentation de l'exemple d'apprentissage suivant jusqu'à ce que tous les exemples d'apprentissage aient été présentés, puis répéter tout le processus pour les itérations suivantes $(n+2), (n+3), \dots$ jusqu'à ce qu'une convergence adéquate soit atteinte.

BIBLIOGRAPHIE

- [1] F. Webster and B. Cobbe, "Traffic signals," Great Britain Road Research Laboratory, London, Road Research Technical Paper 56, 1966.
- [2] A. Benesch, "Regulating street traffic in cleveland," *The American City*, pp. 182–184, 1915.
- [3] B. Marsh, "Traffic control," *Annals of the American Academy of Political and Social Science*, vol. 133, pp. 90–113, 1927.
- [4] M. Bell, "Future directions in traffic signal control," *Transportation Research. Part A*, vol. 26A, pp. 303–313, 1992.
- [5] J. Pline, *Traffic Engineering Handbook*. Institute of Transportation Engineers, 2000.
- [6] R. Hall, *Handbook of transportation science*. Kluwer Academic Publishers, 1999.
- [7] Anonyme, "Régulation du trafic routier, feux de circulation, terminologie," Association Française de normalisation (AFNOR), Tech. Rep., 1991.
- [8] W. Brilon and T. Miltner, "Capacity at intersections without traffic signals," in *Transportation Research Record : Journal of the Transportation Research Board*, no. 1920, Washington, D.C, 2005, pp. 32–40.
- [9] J. Kamata and T. Oda, "*Detectors for road traffic*" *Concise encyclopædia of traffic and transportation systems*. Pergamon Press, 1991.
- [10] A. Klein, k. Mills, and P. Gibson, *Traffic Detector Handbook*. U.S.Department of Transportation Federal Highway Administration, 2006.
- [11] S. Avner, R. Glachet, M. Pintado, and F. Boillot, "L'image au service de la gestion du trafic urbain," *T.E.C*, vol. 160, pp. 9–13, 2000.

- [12] C. Chaudy, J. F. Delaigle, and M. Macq, "Analyse du trafic routier par caméra intelligente : Mvtraffic," UCL, Tech. Rep., 2004.
- [13] D. Robertson, "Transyt : Traffic network study tool," in *Fourth International Symposium on the Theory of Traffic Flow*. Karlsruhe, Germany, 1968.
- [14] C. W. et al, *TRANSYT-7F User's Manual*. Gainesville, Florida : Transportation Research Center, University of Florida, 1988.
- [15] D. Hale, *Traffic Network Study Tool- TRANSYT-7F, United States Version*. Gainesville, Florida : McTrans Center in the University of Florida, 2006.
- [16] P. Hunt, D. Robertson, R. Bretherton, and R. Winto, "Scoot- a traffic responsive method of coordinating signals," Transport and Road Research Laboratory, Crowthorne, Berkshire, U.K, Laboratory Report T1014, 1981.
- [17] R. Bretherton, G. Bowe, P. Burton, and K. Wood, "The use of scoot for traffic management," in *IEEE Second International Conference on Road Traffic Control*. London, U.K, 1986.
- [18] B. Bing and A. Carter, *SCOOT : The World's Foremost Adaptive Traffic Control System*. Toronto, Canada : Traffic Technology International, 2006.
- [19] R. Bretherton, "Current developments in scoot : Version3," *Transportation Research Record*, vol. 1554, pp. 48–52, 1996.
- [20] P. Lowrie, "Scats : The sydney co-ordinated adaptive traffic system - principles, methodology, algorithms," in *Proceedings of the IEE International Conference on Road Traffic Signaling*, London, U.K, 1982, pp. 67–70.
- [21] I. Porche and S. Lafortune, "Adaptive look-ahead optimization of traffic signals," *Intelligent Transportation Systems Journal*, vol. 4, pp. 209–254, 1999.
- [22] M. Bielli, G. Ambrosino, and M. Boero, *Artificial Intelligence Applications to Traffic Engineering*. Utrecht, Netherlands : VSP, 1994.
- [23] C. D. Taranto and V. Mauro, "Utopia," in *Proceedings of the 6th IFAC/IPIF/IFORS Symposium on Control, Computer and Communication in Transportation*. Paris, France, 1989, pp. 245–252.
- [24] J. Henry, J. Farges, and J. Tufal, "The prodyn real time traffic algorithm," in *Proceedings of the IFAC Symposium on Control in Transportation Systems*. Baden-Baden, 1983, pp. 305–310.
- [25] M. Vasudevan, "Robust optimization model for bus priority under arterial progression," Ph.D. dissertation, University of Maryland, 2005.

- [26] N. Gartner, "Opac : A demand-responsive strategy for traffic signal control," *Transportation Research Record*, vol. 906, pp. 75–81, 1983.
- [27] N. Gartner, P. Tarnoff, and C. Andrews, "Evaluation of optimized policies for adaptive control strategy," *Transportation Research Record*, vol. 1324, pp. 105–114, 1991.
- [28] C. Stamatiadis and N. Gartner, "Accelerated progression optimization in large urban grid networks," in *Transportation Research Board 79th Annual Meeting*. Washington D.C., USA, 2000.
- [29] N. Gartner, F. Poorab, and C. Andrews, "Implementation of the opac adaptive control strategy in a traffic signal network," in *IEEE Intelligent Transportation Systems Proceedings*. Oakland, California, USA, 2001.
- [30] F. Boillot, S. Midenet, and J.-C. Pierrelée, "The real-time urban traffic control system cronos : Algorithm and experiments," *Transportation Research Part C : Emerging Technologies*, vol. 14, no. 1, pp. 18–38, 2006.
- [31] J. Kuester and J. Mize, *Optimization Techniques with Fortran*. M.c.Graw-Hill, 1973.
- [32] S. Chen, "Real-time traffic signal control for over-saturated networks," Ph.D. dissertation, Texas Tech University, 2007.
- [33] T. Chang and J. Lin, "Optimal signal timing for an oversaturated intersection," *Transportation Research Part B*, vol. 34, no. 6, pp. 471–491, 2000.
- [34] B. D. Schutter, "Optimizing acyclic traffic signal switching sequences through an extended linear complementarity problem formulation," *European Journal of Operational Research*, vol. 139, no. 2, pp. 400–415, 2002.
- [35] E. F. Stephen Boyd, Laurent El Ghaoui and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia : Society for Industrial and Applied Mathematics, 1994.
- [36] P. O. Gutman and P. Hagander, "A new design of constrained controllers for linear systems," *IEEE Trans on Automatic Control*, vol. 30, no. 1, pp. 22–33, 1985.
- [37] G. Bitsoris, "Positively invariant sets of discrete-time linear systems," *Int. J. Control*, vol. 45, pp. 1713–1726, 1988.
- [38] J. chegancas, *Sur le concept d'invariance positive appliqué à l'étude de la commande contrainte des systèmes dynamiques*. These de Doctorat, Universite Paul Sabatier, Toulouse, 1985.

- [39] A. Benzaouia, *Le Concept d'invariance positive appliqué à la commande des systèmes linéaires discrets avec contraintes*. Toulouse : Thèse : Université Paul Sabatier, 1988.
- [40] F. M. M. Ait Rami, H. Ayad, "Enlarging ellipsoidal invariant sets for constrained linear systems," *International Journal of Innovative Computing, Information and Control*, vol. 3, no. 5, pp. 1097–1108, 2007.
- [41] F. John, *chapter : Extremum problems with inequalities as subsidiary conditions*. New York : Interscience Publishers, 1948.
- [42] E. W. C. Durieu, B. T. Polyak, "Trace versus determinant in ellipsoidal outer-bounding with application to state estimation," San Francisco, Californie, 1996.
- [43] J. Sturm, "Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11, no. 1, pp. 625–653, 1999.
- [44] J. Lofberg, "Yalmip : A toolbox for modeling and optimization in matlab," in *Computer Aided Control Systems Design, CACSD*, Taipei, Taiwan, 2004, pp. 284–289.
- [45] R. W. Brockett, *Finite dimension linear systems*. New York : Wiley-Interscience, 1970.
- [46] F. Robert, *Matrices nonnégatives et normes vectorielles*. Grenoble : Institut National polytechnique de Grenoble, Cours de DEA, 1973.
- [47] M. Fiedler and V. Ptak, "On matrices with nonpositive off-diagonal elements and positive principal minors," *Czech. Math. J.*, vol. 12, no. 3, pp. 382–400, 196257.
- [48] L. Farina and S. Rinaldi, *Positive Linear Systems : Theory and Applications*. New York : Wiley-Interscience, 2000.
- [49] J. Hennet, "Une extension du lemme de farkas et son application au problème de régulation linéaire sous contraintes," *Compte rendus à l'Academie des Sciences*, vol. 308, pp. 415–419, 1989.
- [50] R. S. Varga, *Matrix iterative analysis*. London : Springer, 2000.
- [51] F. T. M. A. Rami and A. Benzaouia, "Control of constrained positive discrete systems," *Proceedings of the 2007 American Control Conference*, pp. 5851–5856, 2007.
- [52] E. César and B. Richard, *Les Séries Temporelles*. Université de Versailles Saint-Quentin-en-Yvelines, Mars 2006.
- [53] D. Eilstein and al, "Séries temporelles et modèles de régression," Institut de veille sanitaire, Département santé environnement," Rapport, 2005.
- [54] G. Box and G. Jenkins, *Time Series Analysis : Forecasting and Control*. San Francisco : CA : Holden-Day, 1976.

- [55] W. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943.
- [56] J. H. P. Borne, M. Benrejeb, *Les réseaux de neurones - Présentation et applications*. Paris : Technip, Broché, 2008.
- [57] A. Assoum, "Etude de la tolérance aux aléas logiques des réseaux de neurones artificiels," Ph.D. dissertation, Institut National Polytechnique de Grenoble, 1997.
- [58] F. Rosenblatt, *Principles of Neurodynamics : perceptrons and the theory of brain mechanisms*. Washington, D.C : Spartan Books, 1962.
- [59] J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. of the National Academy of Sciences of the USA*, pp. 2554–2558, 1982.
- [60] D. Ackley, G. Hinton, and T. Sejnowski, "A learning algorithm for boltzmann machines," *Proc. of the National Academy of Sciences of the USA*, vol. 9, pp. 147–169, 1985.
- [61] M. Parizeau, *Réseaux de neurones*. Université Laval, 2004.
- [62] T. Kohonen, "Unsupervised learning algorithms," in *Neural Networks : Biological Computers or Electronic Brains*. Springer Verlag, 1990, pp. 29–36.
- [63] M. Stricker, "Réseaux de neurones pour le traitement automatique du langage : conception et réalisation de filtres d'informations," Ph.D. dissertation, Université Pierre et Marie Curie-Paris VI, Décembre 2000.
- [64] C. G. Broyden, "The convergence of a class of double-rank minimization algorithms 2 : the new algorithm," *Journal of the Institute of Mathematics and Its Applications*, vol. 6, pp. 222–231, 1970.
- [65] R. Fletcher, "A new approach to variable metric algorithms," *Computer Journal*, vol. 13, no. 3, pp. 317–322, 1970.
- [66] D. Goldfarb, "Family of variable metric updates derived by variational means," *Mathematics of Computation*, vol. 24, pp. 23–26, 1970.
- [67] D. F. Shanno, "Conditioning of quasi-newton methods for function minimization," *Mathematics of Computation*, vol. 24, pp. 647–656, 1970.
- [68] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly Journal of Applied Mathematics* 2, pp. 164–168, 1944.
- [69] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

- [70] B. Hassibi and D. Stork, "Second order derivatives for network pruning : Optimal brain surgeon," in *Advances in Neural Information Processing Systems 5*. Morgan Kaufmann, 1993, pp. 164–171.
- [71] M. Norgaard, *Neural Networks Based System Identification Toolbox Ver 2 Technical Report 00-E-891*. Departement of Automation, Technical University of Denmark, 2000.
- [72] O. Nelles, *Nonlinear System Identification*. Springer, 2000.
- [73] Y. L. Cun, I. Kanter, and S. A. Solla, "Eigenvalues of covariance matrices : Application to neural network learning," *Physical Review Letters*, vol. 66, no. 18, pp. 2396–2399, 1991.
- [74] M. Norgaard, O. Ravn, N. Poulsen, and L. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems*. Springer, 2001.
- [75] M. S. M. G. F. B. r. S. e. L. H. r. G., J. Marti nez, *Réseaux de neurones - Méthodologie et application*. Paris : Eyrolles, 2002.
- [76] M. Norgaard, "System identification and control with neural networks," Ph.D. dissertation, Institute of Automation, Technical university of Denmark, 1996.
- [77] T. Kohonen, "The self-organizing map," *Neurocomputing*, vol. 21, pp. 1–6, 1998.
- [78] ———, *Self-Organizing Maps. 2nd extended edn*. Berlin, Heidelberg : Springer-Verlag, 1997.
- [79] ———, "The self-organizing map," in *Proceedings of the IEEE*, vol. 78, no. 8, 1990, pp. 1464–1480.
- [80] R. Xu and D. Wunsch, "Survey of clustering algorithms," in *IEEE Transactions on Neural Networks*, vol. 16, no. 3, 2005, pp. 645–678.
- [81] A. Flexer, "On the use of self-organizing maps for clustering and visualization," *Neurocomputing*, vol. 5, no. 5, pp. 373–384, 2001.
- [82] T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas, "Engineering applications of the self-organizing map," in *Proceedings of the IEEE*, vol. 84, no. 10, 1996, pp. 1358–1384.
- [83] M. Oja, T. Kohonen, and S. Kaski, "Bibliography of self-organizing map (som) papers : 1998-2001 addendum," *Neural Computing Surveys*, vol. 3, pp. 1–156, 2003.
- [84] G. Barreto and A. Araujo, "Identification and control of dynamical systems using the self-organizing map," *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1244–1259, 2004.

-
- [85] G. Barreto, "Time series prediction with the self-organizing map : A review," in *Perspectives of Neural-Symbolic Integration*, ser. Studies in Computational Intelligence. Springer Berlin / Heidelberg, 2007, vol. 77, pp. 135–158.
- [86] G. Barreto and A. Araujo, "A self-organizing narx network and its application to prediction of chaotic time series," in *Proceedings of the IEEE-INNS Int. Joint Conf. on Neural Networks, (IJCNN01)*, vol. 3, no. 10, Washington, D.C, 2001, pp. 2144–2149.
- [87] G. Barreto, A. Araujo, and H. Ritter, "Self-organizing feature maps for modeling and control of robotic manipulators," *Journal of Intelligent and Robotic Systems*, vol. 36, no. 4, pp. 407–450, 2004.
- [88] J. vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas, "Som toolbox for matlab 5," Helsinki University of Technology, Finland, Technical report, 2000.
- [89] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.