Communication interprocessus entre Dynasim et Scilab pour l'évaluation d'une stratégie de régulation du trafic urbain

Sofiene Kachroudi, Neila Bhouri

▶ **To cite this version:**

Sofiene Kachroudi, Neila Bhouri. Communication interprocessus entre Dynasim et Scilab pour l'évaluation d'une stratégie de régulation du trafic urbain. Congrès de l'ATEC, Jan 2007, Issy les Moulineaux, France. 12p. hal-01473684

HAL Id: hal-01473684
https://hal.science/hal-01473684

Submitted on 22 Feb 2017

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

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

Communication interprocessus entre Dynasim et Scilab pour l'évaluation d'une stratégie de régulation du trafic urbain

Sofiene KACHROUDI **&** **Neila BHOURI**

Sofiane.kachroudi@inrets.fr bhouri@inrets.fr

INRETS/GRETIA– Arcueil, 2 av. Général Malleret-Joinville
94114 Arcueil

RÉSUMÉ : *L'objectif de cet article est de décrire la façon avec laquelle nous avons établi la communication entre un logiciel de simulation microscopique du trafic Dynasim et le logiciel de calcul Scilab pour évaluer notre stratégie de régulation du trafic multimodale NeTPrior. Après une présentation de Dynasim, l'article explique la mise en œuvre de la communication entre ce dernier et Scilab. Les fondements théoriques de NeTPrior sont ensuite donnés ainsi que le réseau de test. Les résultats de simulation sont présentés et commentés.*

MOTS-CLÉS : *stratégie NeTPrior, Communication interprocessus, voitures particulières (VP), Transport en commun (TC), modélisation, multimodale.*

1. Introduction

Le problème de la régulation du trafic routier occupe une place centrale dans les recherches sur les réseaux de transport. En milieu urbain, le développement de stratégies de régulation est devenu même une nécessité ; d'une part à cause de la croissance rapide de l'usage des véhicules particuliers (VP), d'autre part pour favoriser les transports en commun de surface (TC) qui peuvent être une alternative aux VPs.

Plusieurs stratégies ont été développées et certaines sont appliquées sur des réseaux réels. Cependant pour le mode TC ces stratégies permettent de donner la priorité aux feux au niveau local c'est-à-dire quand le véhicule est à proximité des feux. Ce ci améliore le temps de parcours des TCs quand le trafic est fluide. Cependant dans le cas d'une forte perturbation du réseau, donner la priorité au niveau local d'un carrefour peut permettre d'alimenter des zones déjà congestionnées. D'où la nécessité d'une stratégie qui agit au niveau global sur tous les carrefours du réseau.

Après la phase de conception d'une stratégie, une phase de test est nécessaire. Les tests sur le terrain sont coûteux, requièrent beaucoup de temps et peuvent être dangereux pour la sécurité des personnes. C'est pourquoi il est plus judicieux de procéder à des tests en simulation à condition que le modèle de simulation utilisé soit suffisamment fidèle à la réalité.

Cet article a un double objectif. Le premier est de présenter un processus de communication entre un logiciel de simulation microscopique du trafic et une application extérieure sous laquelle est implémentée une stratégie de régulation. Ce processus est nécessaire pour évaluer une stratégie avec un logiciel de simulation avant de la tester sur un réseau réel. Le second objectif est de présenter la stratégie de régulation multimodale (VP et TC) NeTPrior qu'a été développée sous Scilab et que nous évaluons à l'aide de Dynasim.

L'article est organisé de la manière suivante. Nous commençons par poser la problématique de notre travail. Nous introduisons ensuite le logiciel de simulation microscopique du trafic Dynasim ainsi que le principe de la communication interprocessus entre lui et une application extérieure. Puis nous présentons la stratégie de régulation NeTPrior, le réseau de test et la mise en œuvre de la communication interprocessus. Nous donnons, par la suite, des résultats des simulations illustrant l'apport de la stratégie sur deux arcs du réseau. Sur le premier arc circulent les véhicules TC et VP et sur le second ne circulent que les VPs. Enfin nous présentons nos conclusions sur la communication interprocessus et sur la stratégie.

2. Position du problème

Notre objectif est d'évaluer la stratégie de régulation du trafic urbain multimodal NeTPrior. Pour des raisons évidentes de sécurité et de coût, cette stratégie de

régulation du trafic routier ne peut pas être évaluée directement sur un réseau réel. Pour ceci nous utilisons un modèle de simulation. Nous avons choisi un logiciel de simulation microscopique qui s'appelle Dynasim. De notre point de vue, ce logiciel est bien adapté pour des simulations en milieu urbain.
Notre stratégie est fondée sur les méthodes de contrôle automatique du trafic routier. L'architecture générale de tels systèmes de régulation [5] est présentée sur la figure 1.

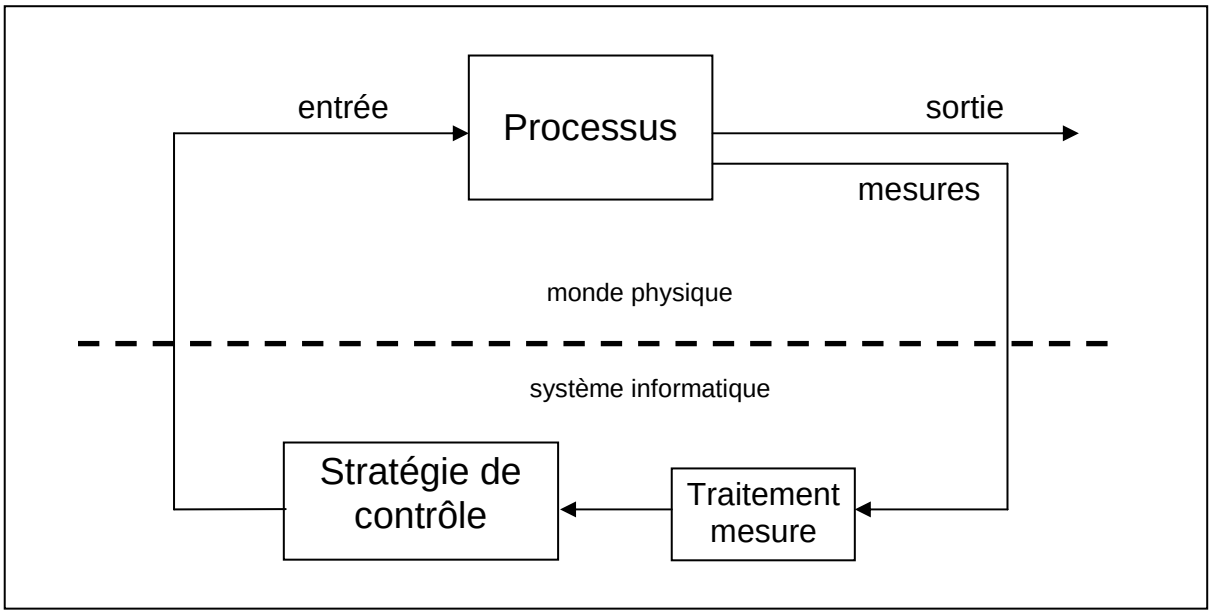


figure 1 : architecture générale d'un système de contrôle automatique du trafic routier

Le processus qui est normalement le réseau routier est dans ce contexte le modèle microscopique de Dynasim. Cette architecture met en évidence un lien bidirectionnel entre le modèle de simulation et l'application sous laquelle est implémentée la stratégie. D'où la nécessité d'une communication interprocessus entre ces deux applications.

3. Communication interprocessus : Dynasim et Scilab

3.1. Dynasim : logiciel de simulation microscopique

Dynasim est un logiciel de simulation dynamique microscopique de la circulation routière. Les véhicules sont pris en compte individuellement et leurs propriétés comportementales et cinématiques sont issues d'une distribution paramétrable par l'utilisateur.

Par rapport aux modèles macroscopiques, le modèle microscopique de Dynasim offre une représentation plus fine du trafic. Il permet, en outre, de tenir compte de tous les modes de transports : bus, tramway, …etc

Dynasim possède une interface graphique facile à utiliser. Cependant, l'édition d'un réseau routier en milieu urbain constitué de carrefours à feux nécessite un réglage très fin des paramètres pour avoir un aménagement qui fonctionne.

Les grandes étapes pour l'édition et la simulation d'un aménagement urbain sous Dynasim sont les suivantes :

- Modélisation géométrique de l'aménagement : routes, arrêts bus, feux tricolores, …
- Définition d'un scénario de flux de véhicules suivant « Origine - Destination ».
- Définition d'un scénario des feux tricolores.
- Définition d'un scénario de simulation.

Pour exploiter les résultats de simulation, le logiciel permet une analyse qualitative grâce à une visualisation 2D et 3D de la circulation et une analyse quantitative par le système de réplications qui génère des résultats statistiques suivant divers critères : temps de parcours, vitesse moyenne, etc.

Pour réaliser des analyses quantitatives, Dynasim possède deux types d'objets : les points de mesure et les capteurs. Les valeurs renvoyées par les points de mesure ne sont accessibles qu'à la fin d'une simulation. Par contre les valeurs renvoyées par les capteurs sont accessibles à tout instant durant la simulation. Ce sont ces derniers que nous utiliserons pour renvoyer des mesures de Dynasim vers une application extérieure.

3.2. Mise en oeuvre de la communication interprocessus

L'objectif est de permettre à Dynasim de recevoir ou d'envoyer des informations depuis ou vers une application extérieure. Le schéma de la figure 2 permet d'expliquer cette communication.

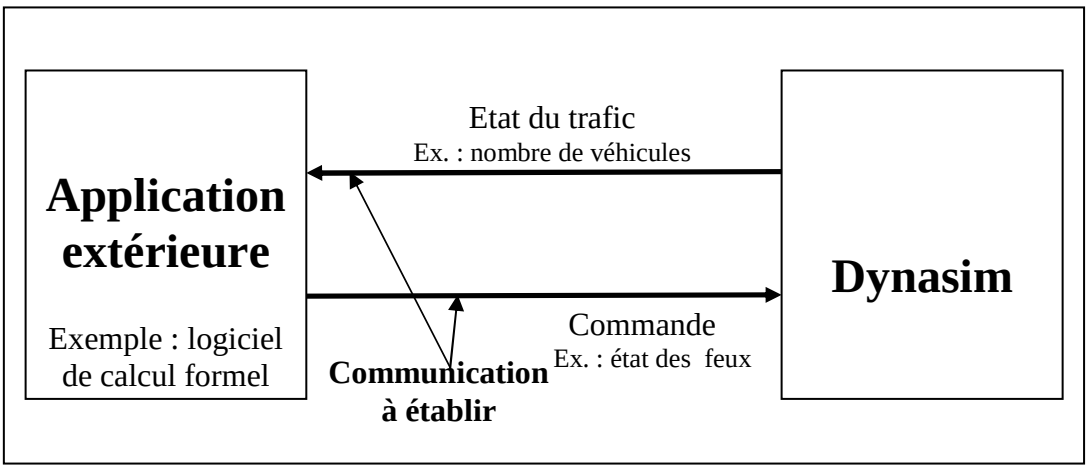


figure 2 : Communication interprocessus
entre Dynasim et une application extérieure

Sous sa version de base, le logiciel Dynasim ne permet pas de recevoir et/ou d'envoyer des informations depuis ou vers une application extérieure durant la simulation de l'aménagement routier. Pour permettre cette communication, nécessaire à notre application, la société Dynasim a développé une interface de programmation ou API (*Application Programming Interface)* écrite en langage C. Cette API permet théoriquement l'interface avec n'importe quel langage supportant les DLL (*Dynamic Link Library*). Pour l'instant cette API ne permet que la commande des feux à partir de l'état des capteurs.

Si la plupart des logiciels de calcul formel supportent les DLLs, le logiciel Scilab sous lequel notre stratégie est implémentée ne le permet pas. Nous avons donc opté à la solution de lancer Scilab à partir d'un programme C. L'architecture finale de la communication interprocessus utilisée est donnée sur la figure 3.

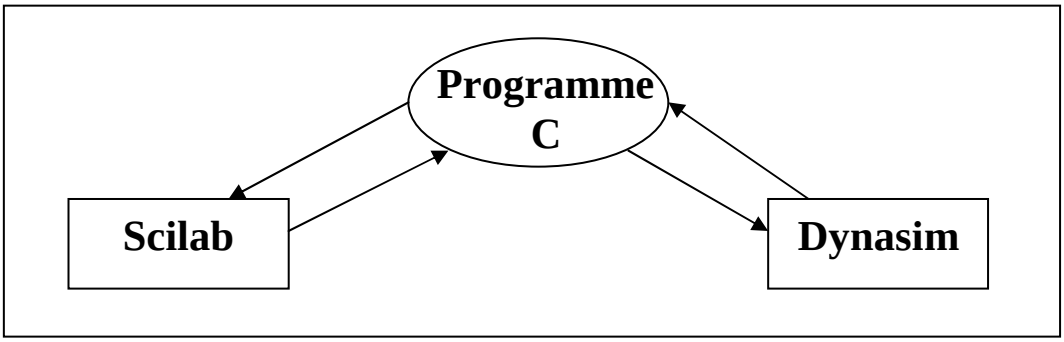


figure 3: Architecture de la communication interprocessus entre Dynasim et Scilab

4. Evaluation d'une stratégie de régulation du trafic

4.1. La stratégie de régulation

Le but de la stratégie NeTPrior est d'agir sur les feux de circulation du réseau afin de minimiser le nombre de véhicules sur les arcs en particulier sur les arcs où les lignes de bus passent et à l'instant où ils passent.
Partant d'un modèle multimodal du trafic urbain, on utilise la méthode de commande linéaire quadratique (LQ) pour déterminer les durées optimales des feux de tous les carrefours [1].
NeTPrior suppose un cycle de feu fixe, un décalage et un plan de phase connus. La loi de commande est déterminée compte tenu d'un modèle linéaire et d'un critère quadratique.

Le modèle utilisé consiste en deux équations. L'une modélisant la circulation des véhicules particuliers, l'autre la progression des transports en commun dans un réseau urbain.
L'équation modélisant les VPs, consiste en l'équation de conservation du nombre de véhicules sur un arc a.

$$x_a^v(k+1) = x_a^v(k) + q(k) - u(k)$$

Où

$x_a^v(k)$: Nombre de véhicules sur l'arc « a » à l'instant k.

$q(k)$: Débit de véhicule entrant dans l'arc « a ».

$u(k)$: Débit sortant de l'arc « a ».

Les débits u(k) et q(k) sont écrits ensuite en fonction des durées des feux verts G(k) de l'arc « a » et des arcs qui mènent à « a ».

Connaissant l'itinéraire d'une ligne de bus bi (d'un arc a' à un arc a par exemple), la progression de ce mode de transport est modélisée par l'équation suivante :

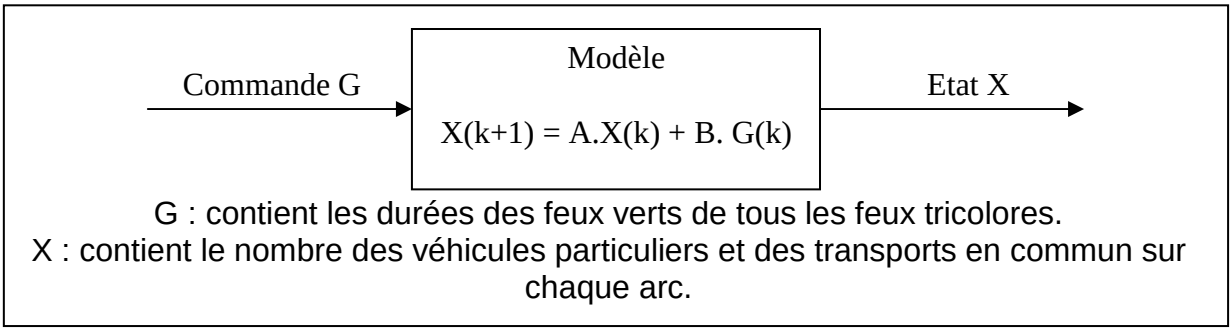
$$x_a^{bi}(k) = x_{a'}^{bi}(k - t_i) \text{ Où}$$

Où

$x_a^{bi}(k)$: Nombre de bus sur l'arc « a » à l'instant k.

t_i : Temps de parcours du bus entre l'arc « a' » et l'arc « a »

Sur la figure 4, nous mettons le modèle sous forme d'un système avec un vecteur d'entrée G et un vecteur de sortie X. k désigne l'instant k.T où T est la période d'échantillonnage (T = C, C étant le cycle des feux des carrefours).



G : contient les durées des feux verts de tous les feux tricolores.
X : contient le nombre des véhicules particuliers et des transports en commun sur chaque arc.

figure 4 : Modèle de commande d'un système de trafic urbain multimodal

A partir des mesures sur le vecteur X, la commande LQ calcule le vecteur d'entrée G optimal qui minimise un critère quadratique. Ce critère consiste en deux composants. Le premier a pour objectif de minimiser le nombre de véhicules sur la totalité du réseau. Le second permet de minimiser le nombre de véhicules sur les arcs où les TCs sont présents. Le schéma de principe de la commande est le suivant :

La stratégie est implémentée sous Scilab. Elle a été testée sur un modèle théorique du trafic [2]. Notre objectif est de l'évaluer sur un réseau urbain réel modélisé sous Dynasim.

4.2. La modélisation sous Dynasim

Le réseau de test considéré est situé dans le sud de la ville de Paris, une région où le trafic est dense. Le réseau comprend cinq carrefours qui sont gérés par des feux tricolores. Il est traversé par une ligne de bus avec une fréquence de 1 bus tous les 5 minutes.
Le modèle du réseau sous Dynasim est sur la figure 5.

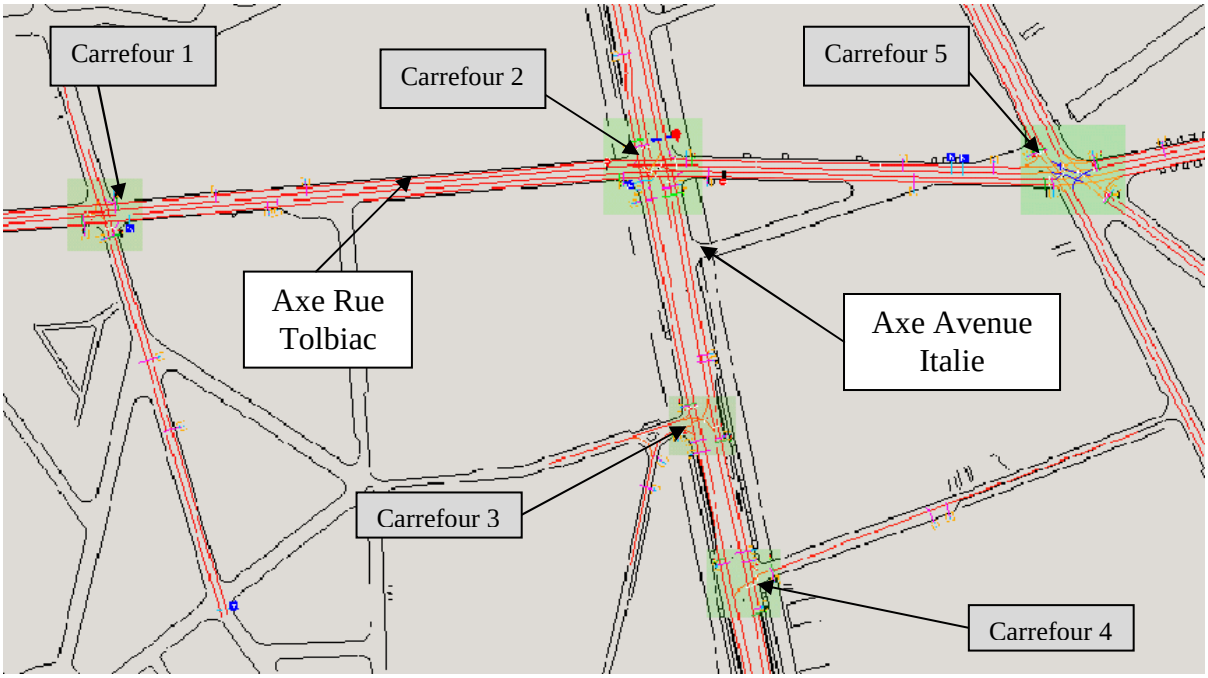


figure 5 : Réseau de test modélisé sous Dynasim

NeTPrior récupère le nombre de véhicules sur chaque arc du réseau et calcule la durée optimale du feu vert qui permet de minimiser ce nombre.

Pour récupérer le nombre de véhicules sur chaque arc, on utilise des capteurs qu'on place sur les arcs. Une règle stricte de dénomination des feux et des capteurs doit être respectée afin de pouvoir récupérer les valeurs des capteurs et de commander les feux.

4.3. La communication interprocessus

La mise en œuvre de la communication interprocessus nécessite l'existence de trois modules :

- Module Dynasim : modélisant le réseau urbain.
- Module Scilab : calculant les durées optimales des feux tricolores.
- Module C : module maître qui fait l'interface entre Dynasim et Scilab.

En ce qui concerne le module Dynasim, il faut définir quatre fichiers de configuration. Ces fichiers sont nécessaires pour l'établissement de la communication interprocessus.

Le module Scilab prend comme argument le vecteur constitué par les mesures qu'on recueille de Dynasim et renvoie les valeurs optimales des durées des feux verts.

Le module maître est écrit en langage C. C'est ce dernier qui fait l'interface entre Dynasim et Scilab. Sur la figure 6, nous présentons le chronogramme de ce module. N désigne le nombre de cycle simulé et i un compteur de cycle.

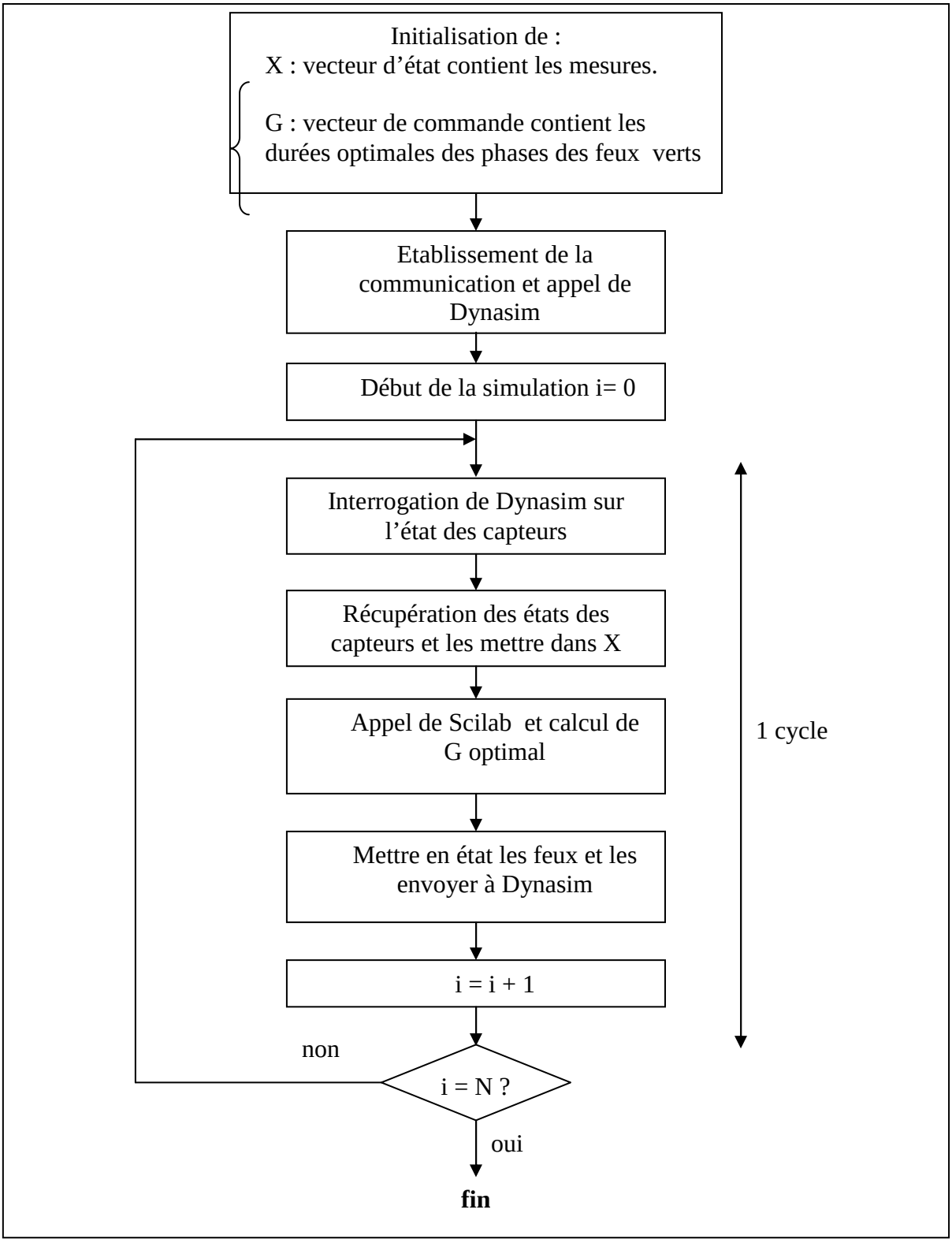


figure 6 : Chronogramme du module maître

Bien que l'appel de Scilab par le module maître soit effectué en écrivant et en lisant à travers des fichiers, le temps de calcul est très satisfaisant et permet d'appliquer cette stratégie en temps réel.

4.4. Les résultats de test

Les tests ont été effectués sur deux heures et quart avec un flux de véhicules correspondant à une heure de pointe. Sur le tableau suivant nous présentons les flux et les perturbations que nous avons introduits.

Heure de simulation	Perturbations du flux
7h45 – 8h00	Pas de perturbation
8h00 – 8h30	+ 25% sur l'axe Avenue d'Italie
8h30 – 9h00	+25% sur l'axe Rue de Tolbiac
9h00 – 9h30	+ 25% sur l'axe Avenue d'Italie
9h30 – 10h00	+25% sur l'axe Rue de Tolbiac

Le bus circule sur l'axe Avenue d'Italie du sud au nord avec une fréquence d'un bus tous les cinq minutes. Il traverse, dans l'ordre, les carrefours 4, 3 et 2.

Sur la figure 7, on présente les temps de parcours des bus entre le sud et le nord.

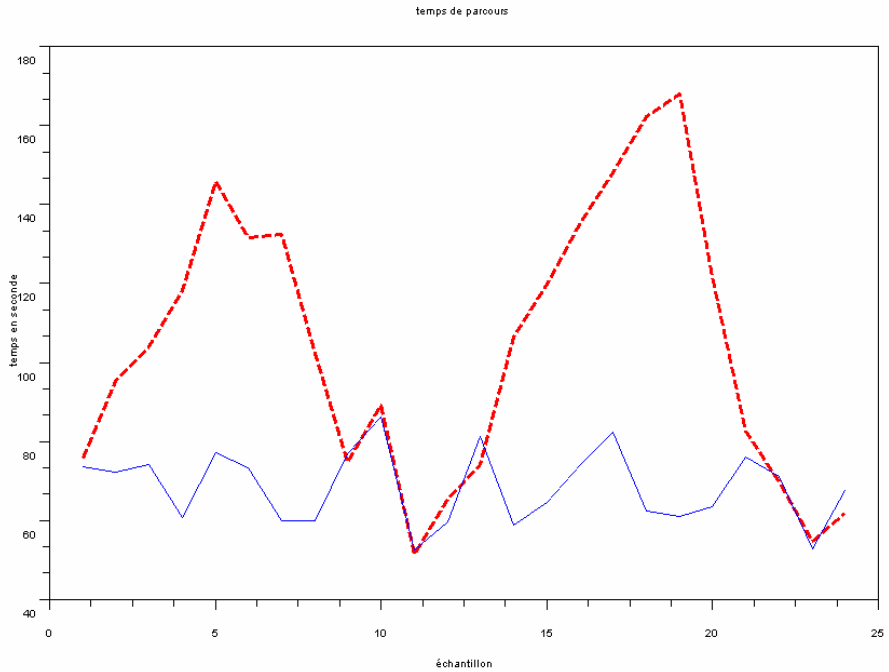


figure 7 : Temps de parcours d'une ligne de bus
' **- -** ' : sans stratégie de régulation
'—' : avec stratégie de régulation

Sur cet axe Sud Nord, nous avons introduit deux perturbations de +25% : la première entre 8h00 et 8h30 et la deuxième entre 9h00 et 9h30. Sans la stratégie et durant les perturbations (surcharge), le temps de parcours des bus s'allonge considérablement et atteint même les 170 secondes sur la distance de test. Par contre avec la stratégie, le temps de parcours n'est pas affecté par la surcharge. Le temps de parcours reste autour des 70 secondes.
On s'intéresse maintenant au carrefour 4 le premier traversé par la ligne de bus. Sur la figure 8, nous pouvons voir en détail ce carrefour.

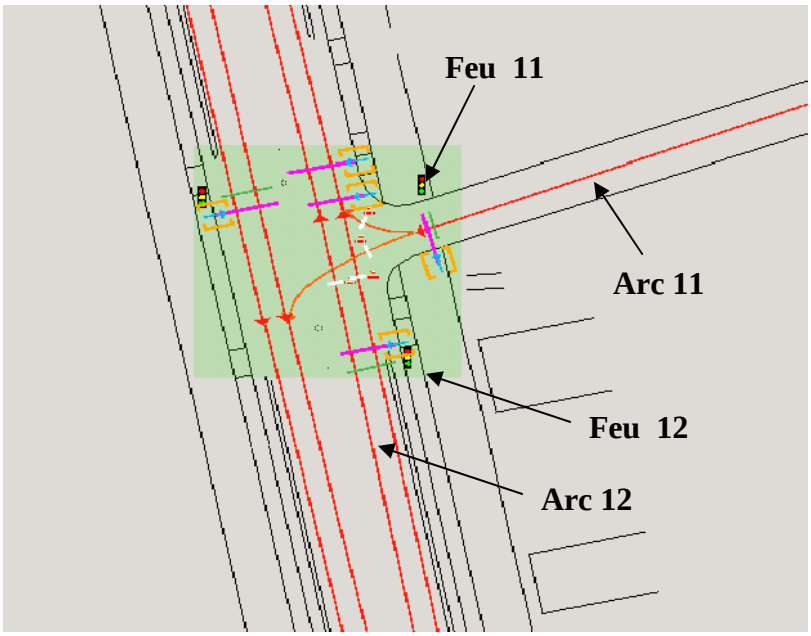
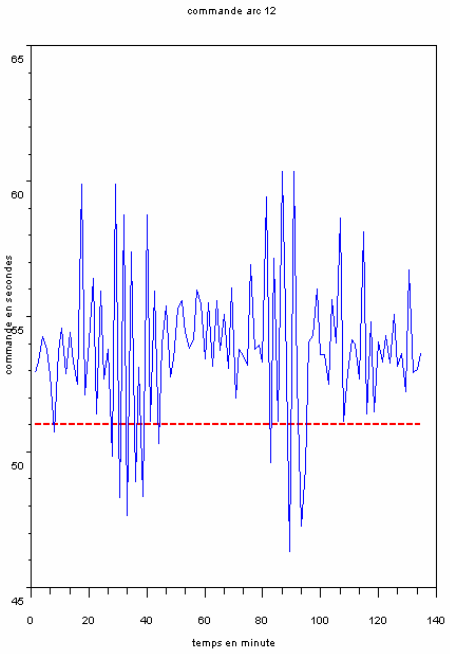


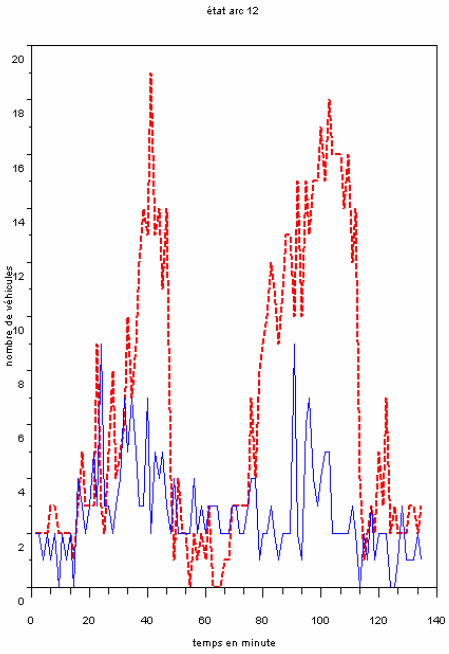
figure 8 : carrefour 4

La stratégie agit pour atteindre deux objectifs. D'une part réguler le trafic sur tout le réseau et d'autre part favoriser les TCs en régulant en particulier les arcs où des bus passent et à l'instant où ils passent. Nous mettons en évidence l'apport de la stratégie sur ces deux aspects à travers deux arcs du réseau. Le premier est l'arc 12 qui supporte une ligne de bus. Le deuxième est l'arc 11 qui est utilisé seulement par les VPs.

La figure 9.donne les durées du feu vert accordées à l'arc 12 pour chaque cycle de feux et son état qui est le nombre total de véhicules. Sur la figure 10, nous traçons les durées du feu vert et le nombre total de véhicules sur l'arc 11. Pour chaque arc, nous avons tracé la commande et l'état de notre stratégie en comparaison avec un réseau géré sans stratégie de régulation (durées constantes du vert).



(9.a)

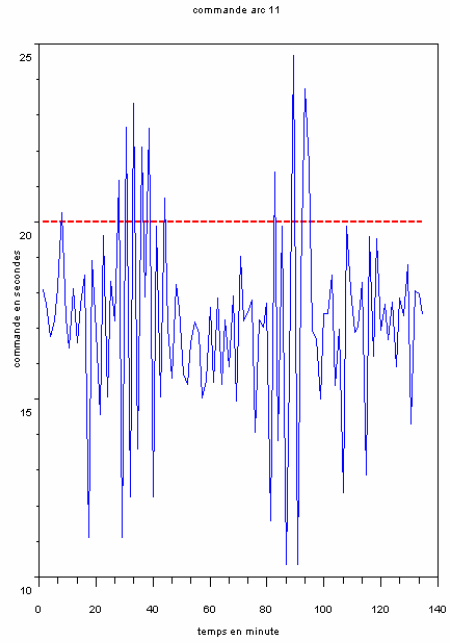


(9.b)

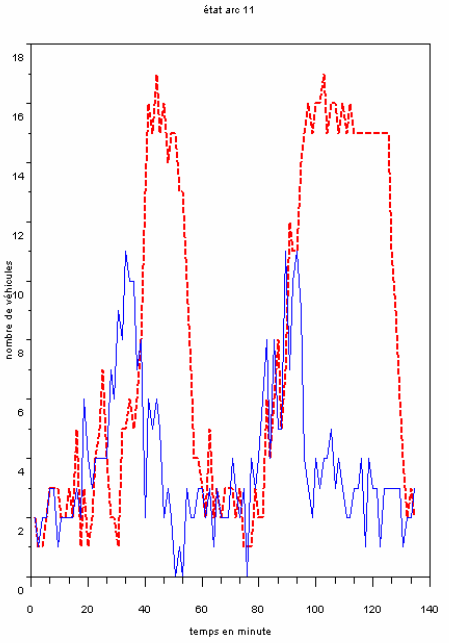
figure 9 : Durées des feux verts et Nombre de véhicules

- - : sans stratégie de régulation

—— : avec stratégie de régulation



(10.a)



(10.b)

figure 10 : Durées des feux verts et Nombre de véhicules

- - : sans stratégie de régulation

—— : avec stratégie de régulation

Nous pouvons observer sur les figures 9.a et 10.a que les durées des feux verts s'adaptent en fonction de la densité du trafic. Ceci permet de réduire le nombre de véhicule sur les arcs (voir figures 9.b et 10.b) et d'éliminer la congestion.

A travers ces tests, nous avons mis en évidence que notre stratégie fonctionne parfaitement pour un réseau de complexité moyenne et pour des perturbations qui affectent un seul axe à la fois. Cependant pour des perturbations généralisées et avec plusieurs lignes de bus circulant sur le réseau, les résultats ne sont pas très satisfaisants. D'autres pistes seront explorées.

5. Conclusion

Nous avons présenté dans cet article comment établir une communication interprocessus entre le logiciel de simulation microscopique du trafic Dynasim et le logiciel Scilab. Cette communication était nécessaire pour évaluer notre stratégie de régulation du trafic urbain NeTPrior. Cette stratégie a pour objectif de favoriser les transports en commun sur les arcs où circulent et à l'instant où ils y sont sans pénaliser les véhicules particuliers. Bien que la dimension du réseau de test ne soit pas très importante, l'évaluation a permis de tirer deux conclusions. La première est que notre stratégie est parfaitement applicable en temps réel puisque les temps de calcul sont largement inférieurs à la durée du cycle C (C = 80 s en général). La seconde conclusion est que pour des perturbations sur un seul axe du réseau urbain, NeTPrior parvient parfaitement à réguler le trafic. Pour des perturbations généralisées sur tout le réseau, les tests montrent jusqu'à présent que la stratégie améliore légèrement l'état de la circulation mais ne parvient pas à la réguler complètement. D'autres approches seront explorées dans le futur.

Références

[1]

[2] Bhouri, N. & Lotito, P. (2006). Régulation du trafic urbain multimodal avec priorité pour les transports en commun, 6^{e} Conférence Francophone de Modélisation et Simulation, Rabat-Maroc.

[3] Culioli, J.-C. (1994), Introduction à l'optimisation, Edition Ellipses.

[4] Alazard, D. (2002), La régulation LQ/LQG, notes de cours SUPAERO.

[5] Papageorgiou, M. (2005), Méthodes de contrôle automatique dans les transports et la gestion du trafic, Magazine Génie Logiciel.

[6] Dynalogic, Manuel de référence de Dynasim, www.dynasim.fr.