



**HAL**  
open science

# Leader-follower cooperative control paradigm with applications to urban traffic coordination control

René Boel, Nicolae Marinica, Alain Sarlette

► **To cite this version:**

René Boel, Nicolae Marinica, Alain Sarlette. Leader-follower cooperative control paradigm with applications to urban traffic coordination control. European Control Conference, Jul 2015, Linz, Austria. hal-01248796

**HAL Id: hal-01248796**

**<https://inria.hal.science/hal-01248796>**

Submitted on 29 Dec 2015

**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.

# Leader-follower cooperative control paradigm, with applications to urban traffic coordination control\*

René K. Boel, Nicolae E. Marinică and Alain Sarlette<sup>‡</sup>

## Abstract

*This paper considers control of networks modeled by a directional graph. We propose a leader-follower paradigm for coordinating local feedback controllers in a large network of interacting components, where most components - the followers - can handle local load without problems, but where a few heavily loaded components - the leaders - are close to saturation. The leader-follower paradigm partitions the network in subsystems, with one leader in each subsystem. Global performance can be improved if followers control their output variables so that the resulting input variables for their leader improve the performance at their leader. The leader-follower approach is explained using as a case study the coordination of the switching times of traffic lights in an urban traffic network. The local feedback controller at a leader uses local traffic data only, selfishly minimizing local queues by selecting the switching times of its traffic lights. Local controllers at follower intersections minimize a local delay, but also take into account specifications generated by their leader. Simulations show that this approach can improve system performance. The flexible feedback nature of this leader-follower control paradigm in one subsystem of the overall network makes it a good candidate as a building block for a scalable hierarchical controller for the complete network.*

## 1. Introduction

Given the heavy social impact of urban traffic, both in terms of delays and in terms of pollution, and given the limitations on physical expansion of the infrastructure, it is no surprise that there exists a huge literature on feedback control of traffic lights in order to

improve the performance of urban traffic [1],[10],[11]. The difficulty of predicting behavior of vehicles, the complicated, hard to measure patterns of urban traffic, and the size of the network, make the design of good feedback controllers for traffic lights a very challenging problem. Scalable designs require a hierarchical control architecture, using cooperating local control agents at various levels of the control hierarchy. Some proprietary commercial tools for traffic management already implement such adaptive, hierarchical traffic controllers: OPAC [6], SCOOT[8], a component of the UTMC [11] traffic manager, and SCATS[9], and the platoon-based hierarchically optimizing RHODES[7], which is closest in approach to this paper. However all these tools use cycle time, red-green split and offset as controlled variables, while this paper proposes the actual switching times of the traffic lights as controlled variables. The proposed leader-follower approach, combined with novel sensor technologies and ICT tools, provides enough freedom so that switching times can adapt on-line to local circumstances while still coordinating intersections.

The leader-follower paradigm for coordinated control, proposed in [5], is applicable to many large networks of interacting components, such as logistics networks, water supply or irrigation networks, and energy networks where the direction of energy flow is fixed. Consider a directed graph model where both nodes and arcs are represented by an input-output dynamical model. The control of node dynamics can be used to improve network behavior (arc dynamics are not controllable in our approach). The leader-follower approach is useful when most nodes can handle their offered load without problems - these nodes are **followers** - except for a few nodes - the **leaders** - that are so heavily loaded that small perturbations there might lead to global problems. Assume that a higher layer controller in the hierarchy has partitioned the network in subsets, with one leader per subset. The leader-follower paradigm for designing distributed feedback controllers then takes into account the interaction between one leader and its followers within one subset.

\*This work was partially supported by the the European Union FP7 project C4C, and by the IAP network DYSCO initiated by the Belgian Science Policy Office.

<sup>‡</sup>The authors are with SYSTeMS, Ghent University, Technologiepark Zwijnaarde 914, 9052 Gent, Belgium. Corresponding author: rene.boel@ugent.be

The followers can handle additional control specifications ensuring that their output reduces the load for their downstream leader, helping the leader to achieve acceptable performance. To "reduce the load" does not mean reducing the average load (it is not possible to reduce the average arrival rate of vehicles at an intersection anyway), but rather modifying the burstiness of the input, or some other second order statistical effect. Of course the load generated at the sources must be executed eventually. In the urban traffic case study this takes the form of selecting departure times of vehicles at a follower so that their arrival occurs during a green period at the leader, so that they can pass the intersection without extra delay and without causing extra delay for other vehicles. This reshaping of the input variables at a leader does not change the long term average value, only the correlation structure.

In our urban traffic case study intersections are controlled by the switching times of the traffic lights. The model describes the evolution of the queue sizes, and the extra delays they cause. The external inputs of the intersection model are the arrival streams of platoons of vehicles generated by upstream intersections. The output represents the flow of vehicles towards downstream intersections. The arc model describes the delay (and the random change in size) of platoons traveling along the arc. This directed graph model includes one arc for each direction of traffic along a road (two-way roads are represented by 2 separate directed arcs).

Performance of an urban traffic network can be improved by minimizing the waste of "green time" capacity, by trying to ensure that vehicles arrive approximately at the time when a traffic light turns green. Green waves try to implement this via open-loop policies. However these open-loop policies cannot adapt to local perturbations of the arrival streams, thus leading to inefficient operation. Moreover it is known that for queueing systems close to saturation [1], as is the case for leader nodes, the delays are extremely sensitive to small perturbations. The long queues resulting from these perturbations (e.g. unexpected delays in arrival times of a platoon of vehicles) may cause blocking of upstream follower intersections: a local problem at a leader node may spread "like a bushfire" and cause drastic deterioration of the global system performance.

The leader-follower approach uses the control freedom at the followers in order to ensure that platoons of vehicles arrive at a leader so as to avoid waste of green capacity at the leader. Of course due to interference between opposing traffic directions, randomness in travel time, and the random change of size of the platoons along a road it is impossible to exactly achieve this goal. Simulation experiments reported (see subsec-

tion 3.3 and [2]) do show that cooperation between a leader and its followers improves overall system performance: the coordination controllers approximately synchronize intersections according to a green wave, but in a flexible way so that the traffic lights can adapt to local perturbations. Note that the green wave signal timing is not imposed by the leader-follower control design. In fact the leader-follower method is not an example of synchronization of systems. Synchronization of traffic lights is not a goal of the controller in itself, and in fact the switching times need to be sufficiently flexible since at the boundary between 2 neighboring subsystems traffic streams synchronized to different leaders will interact. If the synchronization were too strict this could lead to significant performance deterioration.

Section 2 explains how the leader-follower paradigm partitions the graph into connected subsets, each subset having one leader and several followers. How the controllers of each of the components of one subsets can cooperate in order to guarantee good performance at the leader is explained for the urban traffic case in section 3. Subsection 3.3 shows, for a simple subsystem of a Manhattan grid network, with one central leader intersection and 4 followers, that the leader-follower control approach does indeed reduce the delays, not only at the leader, but also in many cases at the followers. Some of the issues of tuning the subsystem controllers as part of a hierarchical control problem and some future extensions of the current work are discussed in the concluding section.

## 2. Leader-follower paradigm as part of a hierarchical control system

### 2.1. Modeling

Consider a dynamical system represented by a directed graph  $(\mathcal{N}, \mathcal{A})$  with  $N = \#\mathcal{N}$  nodes  $Comp_n, n = 1, \dots, N$ , interconnected by  $A = \#\mathcal{A}$  directed arcs  $(n, m)$  connecting  $Comp_n \in \mathcal{N}$  to  $Comp_m \in \mathcal{N}$ . The set  $*Comp_n = \{\ell : (\ell, n) \in \mathcal{A}\}$  represents all the input arcs connecting  $Comp_n$ , to its upstream nodes  $Comp_\ell$ . Similarly  $Comp_n^* = \{\ell : (n, \ell) \in \mathcal{A}\}$  represents all the output arcs connecting node  $Comp_n$ , to its downstream nodes. Components  $Comp_n \in \mathcal{N}$  are modeled as input-output dynamical systems, with controllable input variables  $u_n$ , and external input variables  $v'_{\ell, n}, \ell \in *Comp_n$ . The dynamical system model  $Comp_n$  (note that we abuse notation and denote both the nodes and their dynamical system model by  $Comp_n$ ) generates outputs  $v_{n, \ell}, \ell \in Comp_n^*$  that influence the downstream neighbors of  $Comp_n$ , defining the interaction between components. Each arc  $(n, m)$  is represented as a dynamical

cal system, transforming the output  $v_{n,m}$  of its upstream component  $Comp_n$  into the, usually delayed, external input  $v'_{n,m}$  of its downstream component  $Comp_m$ . Some uncontrollable source nodes  $Comp_s$  at the edge of the network, with  $*Comp_s = \emptyset$ , generate according to a given random model the output  $v_{s,\ell}, Comp^* = \{\ell\}$ .

The dynamic models for nodes can be heterogeneous, with a different type of model in different nodes. In the urban traffic case for example some nodes may use a detailed hybrid model that keeps track of individual arrival times of vehicles, or a more abstract model keeping track of arrival times and sizes of platoons of vehicles [4], or an even more abstracted fluid flow model keeping track only of flow rates of vehicles averaged over several cycles of the traffic signals [12]. All that is needed is for the arc models to properly translate output variables  $v_{\ell,n}$  for one model into input variables into  $v'_{\ell,n}$  for another model.

## 2.2. Supervisory control and partitioning

We assume in this paper that each component  $Comp_n$  can locally observe some time averaged values (like queue sizes, arrival rates) that can be sent to the supervisor, and that provide an indication of how heavily loaded a node is. In any case application of the leader-follower paradigm requires a simple criterion, involving infrequent low-rate communication between the components and the central supervisor, that allows the supervisor to decide whether a component can easily handle its specifications - these components are called followers - or whether a component has difficulties in achieving its specifications, and should be treated as a leader. The supervisor thus selects a subset  $\mathcal{M}_{\text{leader}} \subset \mathcal{N}$  of leader nodes  $Comp_i, i = 1, \dots, \#\mathcal{M}_{\text{leader}}$ . The supervisor then partitions the network  $\mathcal{N}$  into  $\#\mathcal{M}_{\text{leader}}$  connected subgraphs  $\mathcal{S}_i \subset \mathcal{N}$  such that  $\mathcal{S}_i$  contains the leader node  $Comp_i$ , and some upstream follower nodes. We assume that a leader  $Comp_i$  is never at the edge of a subsystem  $\mathcal{S}_i$ , i.e. a leader never is a neighbor of a component in another subsystem. Some nodes in  $\mathcal{S}_i$  may be both upstream and downstream neighbors (two-way roads connecting 2 intersections in the urban traffic case study), but the allocation of a follower to a leader is based only on upstream connections.

The long term averaged information on load flows in the network, used to partition the network can also be used to tune parameters of the cost functions that are used, as explained in the next subsection, by the local controllers in each component  $Comp_i, i = 1, \dots, N$ . These tuning parameters must be communicated by the supervisor to each of the components. Note that the partitioning and these parameters may be adapted by the

supervisor whenever it receives load updates.

## 2.3. Partitioning an urban traffic network

Consider, as an illustrative example, the case where the network  $\mathcal{N}$  is an urban traffic network with  $N$  signalized intersections  $Comp_n$ . By properly selecting the switching times of the traffic lights at these  $N$  intersections the extra delay caused by vehicles waiting in queues at the intersections should be minimized. The vehicles passing through the network are generated by source nodes  $Comp_s$  (such that  $*Comp_s = \emptyset$ ). Source nodes  $Comp_s$  generate vehicles at a rate of  $\lambda_s(t)$  vehicles per sec. Assuming that at each intersection  $Comp_n$  all turning ratios are known simple linear calculations allow the designer to calculate the flow rates  $\lambda_\ell(t)$  veh/sec along any road  $\mathcal{A}_\ell = (m, n)$  in the network.

To simplify the presentation we consider in this paper intersections  $Comp_n$  where the traffic light only has two phases, i.e.  $*Comp_n = R_{1,n} \uplus R_{2,n}$ , (to simplify the notation we further drop the index  $n$  whenever it is clear from the context) where the traffic from inflow directions  $R_{1,n}$ , resp.  $R_{2,n}$  alternately get green light for durations  $\tau_{1,k}$ , resp.  $\tau_{2,k}, k = 1, \dots$ . This analysis ignores the yellow phase, and the limitations on acceleration/deceleration of vehicles. Traffic lights switch at times  $\theta_1 = \tau_{1,1}, \theta_2 = \tau_{1,1} + \tau_{2,1}$ , etc., in general  $\theta_0 = 0, \theta_{2,k} = \sum_{\ell=1, \dots, k} (\tau_{1,\ell} + \tau_{2,\ell}), \theta_{2,k+1} = \theta_{2,k} + \tau_{1,k+1}$ . If at time  $t \in [\theta_{2,k}, \theta_{2,k+1}]$  traffic from inflow direction  $j \in R_1, j = (n, m) \in \mathcal{A}$  gets green light then traffic from  $Comp_n$  passes intersection  $Comp_m$  at the flow rate  $\mu_j(t) = \mu_j \cdot 1_{Q_j(t) > 0}$  veh/sec, while the queue is not empty; for inflow direction  $j \in R_2$  the flow rate is during that interval  $\mu_j(t) = 0$ . During intervals  $[\theta_{2,k+1}, \theta_{2(k+1)})$  the flow rates for  $R_1$  is 0, and  $R_2$  is served.

Using the average arrival and departure rates along inflow direction  $\ell \in *Comp_n$ , the average number of arriving vehicles during the interval  $[\theta_{2,k}, \theta_{2(k+1)})$  is  $\lambda_\ell \cdot (\theta_{2(k+1)} - \theta_{2k})$  while the average number of departing vehicles is at most  $\tau_{m,k} \cdot \mu_\ell$  (less if the queue empties before the end of the green interval). Queues can remain finite only if the average number of arriving vehicles during a long interval of time is smaller than the number of vehicles that can leave the intersection. It is thus reasonable to define the load  $\rho_{\ell,n}$  of the traffic along inflow direction  $\ell$  during the interval  $[\theta_{2,k}, \theta_{2(k+1)})$  as  $\lambda_\ell \cdot (\theta_{2(k+1)} - \theta_{2(k)}) / \tau_{\ell,k} \cdot \mu_\ell$ . Since all inflow directions  $j \in R_m (m = 1 \text{ or } 2)$  get the same green duration, the load of the intersection  $Comp_n$ , for given sequence  $\tau$  of values of  $\tau_{m,k}$  can be defined as  $\bar{\rho}_{\tau,n} = \max_{j \in *Comp_n} \rho_{j,n}$ .

Note that  $\tau_{1,k}, \tau_{2,k}$ , and thus also the loads  $\bar{\rho}_{\tau,n}$  depend on the on-line controller. For reasons of fairness and in order to ensure robustness against sensor

failures the controller designer must select  $\tau_{1,k}, \tau_{2,k}$  to always satisfy upper and lower bounds for the green periods:  $\tau_{min} \leq \tau_{j,n} \leq \tau_{max}$ . The allocation of green times should thus take into account  $\rho_{j,n}$  taking upper and lower bounds on  $\tau_n$  into account. Hence the supervisor should declare an intersection  $Comp_n$  heavily loaded, and make it a leader, if even for the best possible choice of the red and green intervals  $\tau_{1,k}, \tau_{2,k}$  the load  $\bar{\rho}_{\tau,n}$  gets too close to 1 for at least one of its inflow directions:  $Comp_n$  is selected as leader when  $\min_{\tau_{min} \leq \tau_{j,n} \leq \tau_{max}} \bar{\rho}_{\tau,n} \geq \rho_{threshold}$  for a proper choice of  $\rho_{threshold} < 1$ . This is justified by the classical queueing formula (see e.g. [1]) stating that the average delay for a simple queueing system behaves like  $1/(1 - \rho)$  for  $\rho \approx 1$ , making the delay for that direction extremely sensitive to small perturbations in the traffic flow.

## 2.4. Coordination layer control design

The feedback controller for a leader  $Comp_n, n = 1, \dots, \#\mathcal{N}_{leader}$  generates at time  $t$  the control input signal  $u_n(t)$  as a predictive feedback controller, selecting  $\tau_{1,k}, \tau_{2,k}$  so as to selfishly minimize a local cost function  $c_{i,t}(u_i)$  over a given prediction horizon  $Hor$  while satisfying certain local constraints  $F_{i,t}(u_i) \leq 0$ . The expressions for  $c_n$  and for  $F_n$  depend on the predicted evolution of internal variables of the model  $Comp_n$  (e.g. internal state variables of  $Comp_n$ ) as a function of the selected control values  $u_\sigma, \sigma \in [t, t + Hor]$ , and on input variables  $v'_{\ell,n}, \ell \in^* Comp_n$ , and possibly on parameters imposed by the supervisor. The current value of the internal state variables like queue lengths, and the inflow variables  $v'_{\ell,n}, \ell \in^* Comp_n$ , needed for the prediction, can be predicted (there usually is a delay along an arc) using signals from local sensors and from sensors along the input arcs  $\mathcal{A}_\ell, \ell \in^* Comp_n$ , (possibly after filtering, see [3]). For inflow values farther into the future, beyond the current time  $t$  plus the traffic delay along road  $(\ell, n)$ , the inflow depends on decisions - selection of switching times at  $Comp_\ell$  - still to be made by the follower  $Comp_\ell$ . The leader controller can determine desirable input trajectories for this future interval, that improve its future performance, and send a message  $M_{n,\ell}(t)$  to its upstream follower  $Comp_\ell$  requesting that  $Comp_\ell$  approximately generates outflow satisfying this request encoded by  $M_{n,\ell}(t)$ .

The controller of a follower  $Comp_j \in \mathcal{S}_i$  of  $Comp_i$  similarly selects the control signal  $u_j(t)$  at time  $t$ , minimizing a cost  $c_{j,t}(u_j, M_{n,j}, \omega_{n,j})$  subject to specifications  $F_{j,t}(u_j, M_{n,j}, \omega_{n,j}) \leq 0$ . Cost and specification now depend not only on the evolution of local variables of  $Comp_j$  (obtained from local sensors similar to the leader) but also on the signal  $M_{n,j}(t)$  sent by the leader

$Comp_i$ . If  $Comp_j$  is not a direct neighbor of  $Comp_n$  then this signal can be sent directly from the leader  $Comp_n$  to the follower  $Comp_j$  or be transmitted indirectly via intermediate components connecting  $Comp_j$  to  $Comp_n$ . How strongly this signal  $M_{n,j}$  influences the cost to be minimized and the specifications depends on a tuning parameter  $\omega_{n,j}$ , assigned by the supervisor. Small values of  $\omega_{n,j}$  allow the follower a lot of freedom to optimize its own performance, without helping its leader. Very large values of  $\omega_{n,j}$  on the other hand indicate that the supervisor imposes that follower  $Comp_j$  should not care about local performance, but select its control actions only based on what is needed to reduce the load at the leader of  $\mathcal{S}_i$ . How the cost at a follower depends on the leaders' request will be explained for the urban traffic case study in the next section.

## 3. Coordination layer in urban traffic

### 3.1. Subsystem model

In this section the leader-follower paradigm is applied to the on-line coordination of switching times of traffic lights in an urban area modelled by a subgraph  $\mathcal{S}$  of a larger urban traffic network  $\mathcal{N}$ . The connected subgraph  $\mathcal{S}$  has one single heavily loaded intersection  $Comp_1$ , the leader of this subgraph, and  $N_{follower}$  signalized follower intersections with a lighter load.

The switching times of the traffic lights both at the leader  $Comp_1$  and at the followers are selected on-line so that a local performance measure  $J_i$  is minimized subject to local constraints  $F_i$ . Implicitly this assumes that the average cost experienced by all vehicles traveling through the network is additive: the cost experienced by a vehicle is the sum of the costs experienced at all all intermediate components along its path from source to destination. This is true for the cost used in this case study, where  $J_i$  is the average extra delay experienced by a vehicle due to queueing at intersection  $Comp_i$ . The delay that the vehicles encounter while traveling along the arcs  $(n, m) \in \mathcal{A}$  is not included in the cost since this is the same for all control values (assuming that the control strategy can prevent blocking of upstream intersections). Good performance depends on whether the coordination of the switching times can avoid waste of green capacity.

Coordination is achieved by the leader sending signals to its followers informing them of the times at which they should allow vehicles to enter the road towards the leader, in order to ensure that the platoons of vehicles arrive at such a time that no green capacity is wasted. The use as controlled variable is the next switching time of the local traffic light, not its cycle

time, red/green fraction and phase shift as in most existing traffic management systems ([9], [8]). The model used for designing the local controllers and their coordination, must therefore use a model of traffic that provides information on the next arrival times of vehicles, not just their flow rates. Looking at the arrival times of all the vehicles individually would lead, for a large network, to a state space that is too large. Therefore we opted in this case study to use, at the level of the coordinating control design, for the platoon based traffic model of [4].

This platoon based model aggregates into one platoon  $P_k^n$  those vehicles that travel at close distance of each other with approximately the same speed. The location at a given time of the  $H_k^n$  vehicles in platoon  $P_k^n$ , traveling along arc  $(m,n)$ , is described by the time  $T_k^n$  when the head of the platoon passes a given sensor location along the road  $(m,n)$ . Source node  $Comp_s$  generates platoons  $P_k^s$  with size  $H_k^s$  (a random variable with known distribution around its average  $EH^s$ ) at successive random times  $T_k^s$ . We assume that the time interval  $Int_\ell$  between the  $\ell$ -th and the  $\ell+1$ -st vehicles in the platoon is uniformly distributed in [1,3]. The distribution of the random time  $T_k^{s,tail}$  when the last vehicle - the tail - of the platoon  $P_k^s$  passes the sensor location is given by  $T_k^s$  plus the sum of  $H_k^s$  independent uniformly distributed random variables (on the average  $T_k^\ell + 2.EH^\ell$ ). The head of the next platoon  $P_{k+1}^i$  is randomly generated at  $T_{k+1}^\ell = T_k^\ell + 3 +$  an exponentially distributed random variable with mean  $\Delta_{inter}^s$ .

It has been shown in [4] that this platoon based model corresponds reasonably well to observed traffic data, also inside the traffic network (reaching an equilibrium between the platooning naturally caused by signalized intersections and limitations on overtaking, and the disaggregation of platoons caused by roundabouts, unsignalized side streets, etc.). It also appears from these real traffic data that an increase in traffic intensity usually leads to an increase in platoon size  $EH^s$  while  $\Delta_{inter}^s$  remains approximately constant. The average rate of traffic generated by a source  $Comp_s$ :  $\lambda_s = EH^s / (2.EH^s + \Delta_{inter}^s)$ . Conversely it is easy to find the parameter  $EH^s$  of a platoon based traffic model generating traffic at a given rate  $\lambda_s$ , given the value of  $\Delta_{inter}^s$  which is assumed to remain constant.

Hence traffic along an arc  $(m,n)$  connecting component  $Comp_m$  with  $Comp_n$  is represented by the platoon model, with the  $k$ -th platoon  $T_k^n$  described by its arrival time  $T_k^n, k = 1, \dots, T_{k+1}^n > T_k^n$  when the head of the  $k$ -th platoon passes a sensor location along this arc  $(n,m)$ , and by its size  $H_k^n$ . Passage times and platoon sizes are in this case not determined according to a probability distribution, but depend on the switching times

of the upstream traffic lights, and on the arrival stream of traffic upstream.

An arc  $(n,m)$ , representing a road connecting intersections  $Comp_n$  and  $Comp_m$ , is modeled by a random delay and a random change in platoon size: a platoon of size  $H_n$  leaving intersection  $Comp_n$  at time  $T_n$  arrives at intersection  $Comp_m$  at time  $T_m^l = T_n + D_{n,m}$  with size  $H_m^l = H_n + G_{n,m}$ . The delay  $D_{n,m}$  is a random variable with mean  $ED_{n,m}$ , depending on the length of the road  $(n,m)$  and on the speed at which vehicles can travel;  $G_{n,m}$  is a random variable representing the vehicles entering and leaving the platoon at intermediate unsignalized intersections, parking spaces, etc along  $(n,m)$ .

The model of an intersection  $Comp_n$  describes the evolution of the queues  $Q_n(t)$  at  $Comp_n$ . To simplify the cost calculation we will consider queue sizes below as generated by fluid flow models, which leads in fact to the same average cost. During the time interval  $t \in [T_k^n, T_k^n + 2.(H_k^n - 1))$  between the first and last vehicle of the platoon driving along arc  $(n,m)$ , the vehicles arrive at inflow direction  $n$  of  $Comp_m$  with flow rate  $\lambda_{n,m}(t) = 0.5\text{veh/sec}$  (the average time distance between successive vehicles is 2sec); outside these intervals, in between platoons, the flow rate is  $\lambda_{n,m}(t) = 0$ . The platoon based model thus provides a better definition of where vehicles are, compared to a pure fluid flow model that would use for all times the average flow rate  $EH^\ell / (2.EH^\ell + \Delta_{inter}^\ell)$ . At intersection  $Comp_m$  the arrival rate  $\lambda_{n,m}(t)$  and the departure rate  $\mu_{n,m}(t)$  thus are both piecewise constant. The queue sizes thus evolve according to a piecewise affine trajectory:

$$\dot{Q}_{n,m}(t) = \lambda_{n,m}(t) - \mu(n,m), \quad Q_{n,m}(t_0) \quad (1)$$

The total waiting time of all vehicles passing through a queue - the extra delay of the vehicles passing through that queue - is the integral of the queue size trajectory (provided the system is empty at the initial and the final time).

Hence it is very easy to write a recursive algorithm calculating this total waiting time, adding up the surface areas of the trapezoidal queue size, in between successive times when the value of  $\lambda_{n,m}(t) - \mu_{n,m}(t)$  changes. The ordered sequence of times  $\tilde{\theta}_k, k = 0, 1, \dots$  when the rates  $\lambda_{n,m}(t), \mu_{n,m}(t)$  change at inflow direction  $n$  of  $Comp_m$  includes the times  $T_k^n$ , resp.  $T_k^n + \sum_{\ell=1, \dots, H_k^n-1} Int_\ell$  when the head, resp. the tail of a platoon reaches the intersection, the times when average fluid flow rates change for source traffic, the switching times of the traffic light, and the times when the queue becomes empty (function of all the preceding variables).

Since by equation (1)

$$Q_\ell(\tilde{\theta}_{k+1}) = Q_\ell(\tilde{\theta}_k) + (\lambda_\ell(\tilde{\theta}_k) - \mu_\ell(\tilde{\theta}_k)) \cdot (\tilde{\theta}_{k+1} - \tilde{\theta}_k)$$

the total waiting time in queue  $Q_{n,m}$  over an interval  $[t, t + Hor]$  (where  $Hor$  is can then be written as

$$TW_{n,m}([t, t + Hor], Inf_{n,m}(t), v) = \sum_{t \leq \tilde{\theta}_k \leq t + Hor} (Q_{n,m}(\tilde{\theta}_{k+1}) + Q_{n,m}(\tilde{\theta}_k) \cdot (\tilde{\theta}_{k+1} - \tilde{\theta}_k) / 2)$$

### 3.2. Control design at leader and follower intersections

The choice of the control sequence  $v = \{\tau_1^v, \dots, \tau_{L^v}^v\}$ ,  $\tau_n^v \in [t_k, t_k + Hor]$ , defining the switching times  $\tau_n^v$  of the traffic light at intersection  $Comp_n$  is based on minimizing the total waiting time  $J_n(t_k, Inf_{\ell,n}(t_k), \ell \in^* Comp_n, v)$  over the prediction horizon  $[t, t + Hor]$ . This cost depends on the following information set  $Inf_{n,m}(t)$  at time  $t$ :

- the state of the traffic light at time  $t$  (which direction gets green, and when is the first allowed switching time, taking minimal green time into account)
- the arrival times and the sizes of the platoons arriving in the interval  $[t, t + D_{n,m})$  (available from messages sent by the upstream sensor over an interval  $[t - D_{n,m}, t]$ ); this information allows the calculation of the times when the head, resp. the tail of a platoon arrives at the intersection
- the evolution of the piecewise constant averaged arrival rates,  $t \in [t + D_{n,m}, t + Hor]$  where no platoon information is available and only fluid flow rates can be used for prediction purposes.

The local controller at leader  $Comp_n$  evaluates at successive times  $t_k = k \cdot \Delta$  the cost

$$J_n(t_k, Inf_{\ell,n}(t_k), \ell \in^* Comp_n, v) = \sum_{n \in^* Comp_n} TW_{n,m}([t_k, t_k + Hor], Inf_{n,m}(t_k), v) + \sum_j \frac{Q_j(t_m + Hor)^2}{\mu_j},$$

enumerating the (usually very limited set of) of switching times  $v^* = \{\tau_1^*, \dots, \tau_{L^*}^*\}$ ,  $\tau_n^* \in [t_k, t_k + Hor]$  satisfying maximal and minimal green time constraints ( $\tau_{\min} \leq \tau_{\ell+1}^v - \tau_{\ell}^v \leq \tau_{\max}$ ,  $\ell = 1, \dots, L^v - 1$ , and  $\tau_{\ell^v}^v$  satisfying the constraints imposed by the state of the traffic light at time  $t_k$ ). The control agent then selects the sequence with the lowest cost  $J_n(t_k, Inf_{\ell,n}(t_k), \ell \in^* Comp_n, v)$ . The last term  $\frac{Q_j(t_m + Hor)^2}{\mu_j}$  represents the time it takes the intersection to evacuate the remaining vehicles at the end of the prediction horizon, assuming that each direction gets half its departure rate  $\mu_j$ . This is necessary (see [13]) in order to ensure that the cost of ending

with a long queue (that would cause long delays outside the prediction horizon) is properly taken into account.

If the optimal sequence  $v^* = \{\tau_1^*, \dots, \tau_{L^*}^*\}$  requires switching the traffic light at the current time  $t_k$  then this switching is executed. If not, in MPC-like fashion a new search for an optimal sequence is carried out at  $t_{k+1} = (k + 1) \cdot \Delta$  using updated information  $Inf_{n,m}(t_{k+1})$ . Note that even though  $J_n(t_k, Inf_{\ell,n}(t_k), \ell \in^* Comp_n, v)$  is a quadratic form in the variables  $\tilde{\theta}_k$  it is not possible to find the optimal value  $v^*$  analytically because the constraints on minimal and maximal green time make the minimization into a combinatorial optimization problem (future work will involve finding efficient tools for this discrete event type optimization). In the current test phase for optimization in one single subsystem the optimum is found by enumeration.

The leader  $Comp_n$  sends at time  $t_k$  a message  $M_{n,\ell}(t_k)$  to its follower  $Comp_\ell$  sending traffic along arc  $(\ell, n)$ . This message  $M_{n,\ell}(t_k)$  provides the controller at  $Comp_\ell$  with information on  $v^*$  and on the current queue size  $Q_{\ell,n}(t_k)$ . How detailed this information is depends on the communication constraints of the system.

The controller at follower  $Comp_\ell$  then calculates (and implements it if it involves an immediate switching of the traffic signal), similar to the leader intersection controller, the best sequence  $v_\ell^*$  of successive switching times that minimize the local cost (subject to all the minimal and maximal green time constraints)

$$J_\ell = J_\ell(t_k, Inf_{q,\ell,n}(t_k), q \in^* Comp_\ell, v_\ell) + \omega_{\ell,n} \cdot J_{\ell,non-following}(v_\ell, M_{n,\ell}(t_k)) + \sum_q \frac{Q_q(t_k + Hor)^2}{\mu_j}$$

The optimal switching sequence  $v_\ell^*$  now depends not only on the local information  $Inf_{q,\ell,n}(t_k), q \in^* Comp_\ell$  that  $Comp_\ell$  measures locally or receives from sensors along its inflow roads  $(q, \ell)$ , but also on the message  $M_{n,\ell}(t_k)$  sent by the leader  $Comp_n$ . Indeed the non-following cost  $J_{\ell,non-following}(v_\ell, M_{n,\ell}(t_k))$  expresses the estimate, depending only on information available at  $Comp_\ell$ , of the increase in waiting time at the leader  $Comp_n$  due to the fact that the optimal  $v^*$  implemented by  $Comp_\ell$  deviates from the leader's request as encoded in  $M_{n,\ell}(t_k)$ .

The simplest way  $Comp_\ell$  can calculate the non-following cost  $J_{\ell,non-following}(v_\ell, M_{n,\ell}(t_k))$  starts by determining the first time  $T_{follow}(M_{n,\ell}(t_k))$  after time  $t_k + ED_{\ell,n}$ , where  $D_{\ell,n}$  is the random travel time along the road  $(\ell, n)$ , when the leader  $Comp_n$  can serve a new platoon of vehicles from  $Comp_\ell$ . The follower should then send a platoon at time  $T_{follow}(M_{n,\ell}(t_k)) - ED_{\ell,n}$ . Moreover the follower can calculate how many vehicles  $H_{follow}(M_{n,\ell}(t_k))$  can pass during the remaining green time at the time when the platoon it sends at

$T_{follow}(M_{n,\ell}(t_k))$  will arrive at  $Comp_n$ . Using the local information to minimize  $J_\ell = J_\ell(t_k, Inf_{q,\ell,n}(t_k), q \in \ast Comp_\ell, v_\ell)$  the follower  $Comp_\ell$  however sends a platoon of size  $H_\ell^\ast$  at time  $T_\ell^\ast$ . It is then easy to see that the non-following cost  $J_{\ell,non-following}(v, M_{n,\ell}(t_k))$  is a piecewise linear function of the difference  $\delta = T_{follow}(M_{n,\ell}(t_k)) - T_\ell^\ast$  between the next green switching time implemented at the follower and the time requested by the leader, and of the difference  $H_{follow}(M_{n,\ell}(t_k)) - H_\ell^\ast$  between the platoon size requested by the leader and the platoon size implemented by  $Comp_\ell$ . If the platoon arrives too early it will have to wait at the leader and eventually leave at the same time as if it arrived at time  $T_{follow}(M_{n,\ell}(t_k))$  causing extra delay at the leader (the delay reduction at the follower is weighed less heavily in the cost if  $\omega_{\ell,n} > 1$ ); if it arrives too late at the leader there will be a waste of green capacity leading to extra delay at the leader, and moreover part of the platoon may have to wait during the next red phase. The piecewise affine non-following cost therefore is steeper for  $\delta < 0$  than for  $\delta > 0$ . More sophisticated forms of the non-following cost can be considered, and calculated via simulations generating the queue sizes.

**Remark 1** The proposed calculation method assumes that the follower approximately knows the departure rate during the green periods at  $Comp_n$ , but in the current model we assume that departure rates are 0.5veh/sec everywhere, so this does not impose extra communication requirements;

**Remark 2** Due to the delay of vehicles along arc  $(\ell, n)$  the follower cannot change anything anymore about traffic arriving at the leader prior to  $t_k + ED_{\ell,n}$ . Thus  $Comp_\ell$  only searches for the best platoon arrival time after  $t_k + ED_{\ell,n}$ .

**Remark 3** In case some followers  $Comp_j$  send traffic indirectly to the leader, via an intermediate follower intersection  $Comp_\ell$ , then the message  $M_{n,j}$  can be determined directly by the leader, or may be calculated by the intermediate  $Comp_\ell$ .

### 3.3. Performance evaluation for a simple subsystem

Among the simulation studies reported in [2] we select here, in order to illustrate the performance improvement that can be achieved, the simple subsystem shown in fig. 1, with one central leader intersection. Leader and follower in fact in this case have similar fairly heavy loads, with an average queue size of  $\approx 8$  vehicles, if completely decentralized control were applied (i.e. no coordinating messages from leader to follower). Hence the case study shows that even for heavily loaded followers the leader-follower paradigm can

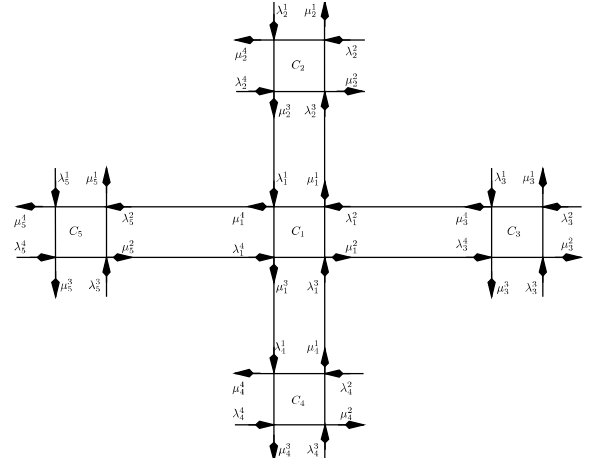


Figure 1. Topology of case study

improve the performance of the overall system. The value of  $\Delta = 1\text{sec}$  was used,  $\tau_{min} = 30\text{sec}$ ,  $\tau_{max} = 60\text{sec}$ . For the non-following cost  $J_{j,non-follow}$  a simple piecewise linear expression of the timing error was used (see fig. 5.17 in [2]).

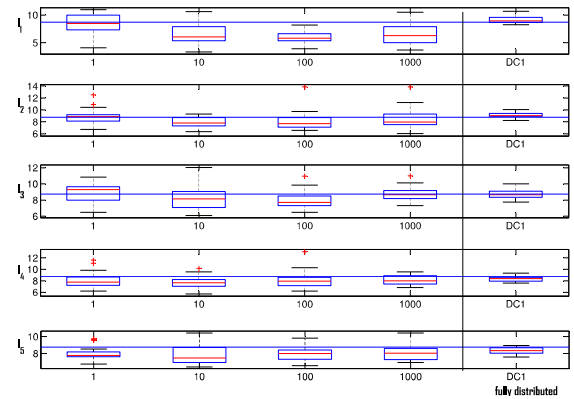


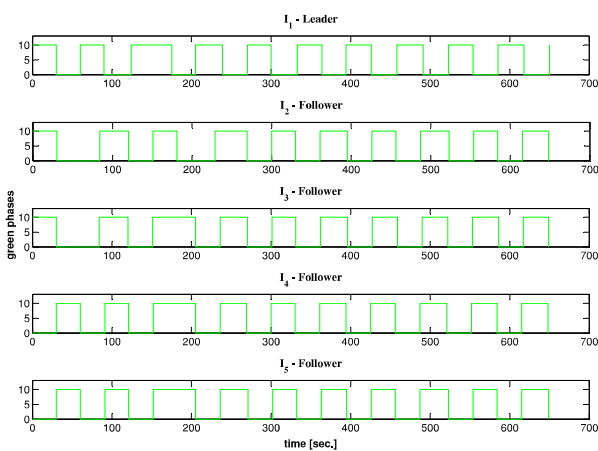
Figure 2. Box plots for simulated delays with leader-follower,  $\omega = 1, 10, 100, 1000$ .

$N = 30$  Monte Carlo experiments simulated network behavior for 700sec, starting with unsynchronized traffic lights, for different values of  $\omega$ . Fig. 2 shows, for the 5 different intersections, box plots (indicating 25% and 75% levels) of the total delay for all vehicles as obtained in these simulations, for different values of  $\omega$ . The column on the right indicates performance for an uncoordinated system. For small values of  $\omega$  there is little improvement. As soon as  $\omega \geq 10$  the histogram at the leader improves significantly. For  $\omega_j = 100$  the 75% upper limit of the waiting time at the leader is smaller than the 25% lower limit for the experiment with  $\omega_j = 1$ . The average delay at the leader for



$\omega_j = 100$  is about 33% smaller than for  $\omega_j = 1$  and for the uncoordinated case. It was found that for  $\omega_j = 100$  the performance improves significantly at the followers (probably thanks to coordination of the intersections reducing variability in the cycles).

Fig.3 shows the evolution of the switching times at each of the intersections for  $\omega = 100$ , representing the green times along the East-West direction for each intersection. It is clear that after about 4 cycles of the traffic lights there is an approximate synchronization, without any explicit form of synchronization being imposed in the cost functions. Fig. 2 also shows that the performance at the followers deteriorates when the tuning parameter  $\omega_j$  is chosen too large ( $\omega = 1000$ ), while the performance at the leader does not significantly improve. In that case the follower intersections no longer have enough freedom to react to local perturbations.



**Figure 3. The phases of the 5 intersections using the leader-follower framework for  $\omega = 100$ .**

## 4. Conclusion

This paper discusses a subproblem of the proposed leader-follower paradigm, showing that leader-follower coordination method improve performance in a single subsystem of the large network, partitioned by the supervisor so that each subsystem has one leader. This performance improvement can be achieved even for values of  $\omega$  that leave enough flexibility to the followers to adapt to local perturbation. Therefore we expect that, for proper choice of the tuning parameter  $\omega$ , it will be possible to obtain good performance for the case where the different subsystems of the large network are interacting with each other. Studying the performance of the leader-follower paradigm applied to several interacting subsystems of a large network will be the next step in the validation of the leader-follower paradigm. It should be emphasized that the requirements on model

knowledge for the design of such a leader-follower controller are very mild: the supervisor must know network topology; the leader must know who are its neighbors, and the model of the arcs connecting it to neighbors; leader and follower must know their own local model only so as to implement a predictive controller; leader and follower must agree on a common format for exchanging messages. The message  $M_{n,\ell}$  only contains a few integers, thus requiring minimal bandwidth. The approach is therefore very robust against modeling errors and communication failures.

## References

- [1] C.F. Daganzo, *Fundamentals of Transportation and Traffic Operations*, Pergamon-Elsevier, Oxford, U.K. (1997). Reprinted: Emerald Bookstore.
- [2] N.E. Marinică, R. Boel and A. Sarlette (co-promoters), *Distributed estimation and control of interacting hybrid systems for traffic applications*, PhD thesis, Ghent University, (January 2014).
- [3] N.E. Marinică, A. Sarlette and R.K. Boel, *Distributed particle filter for urban traffic networks using a platoon based model*, IEEE Trans. Intelligent Transportation Systems, 14(4) : 1918-1929, (2013).
- [4] N.E. Marinică, R.K. Boel, *Platoon based model for urban traffic control*, Proceedings of the American Control Conference, Montreal, July 2012.
- [5] N.E. Marinică, R.K. Boel, *A leader-follower approach for distributed coordination of interacting components*, Proceedings of the 20th International Symposium on Mathematical Theory of Networks and Systems, Melbourne, Nov. 2012.
- [6] N. Gartner, *OPAC: A demand-responsive strategy for traffic signal control*, Transportation Research Record 906, pp. 75-81.
- [7] P. Mirchandani, L. Head, *A real-time traffic signal control system: architecture, algorithms, and analysis*, Transportation Research Part C: Emerging Technologies 9(6), pp. 415-432, 2001.
- [8] <http://www.scoot-utc.com/WhatIsSCOOT.php>
- [9] <http://www.scats.com.au/>
- [10] Fei-Yue Wang, "Agent-based control for networked traffic management systems," Intelligent Systems, IEEE, vol.20, no.5, pp.92,96, Sept.-Oct. 2005 doi: 10.1109/MIS.2005.80
- [11] <http://www.utmc.uk.com/>
- [12] S. Laemmer and D. Helbing, *Self-Control of Traffic Lights and Vehicle Flows in Urban Road Networks*, <http://arxiv.org/abs/0802.0403>, 2008.
- [13] G. Newell, *The rolling horizon scheme of traffic signal control*, Transportation Research, part A: Policy and Practice, Vol. 31, issue 1, Jan. 1998.
- [14] FHWA Traffic signal timing manual, chapter 9, available on-line at: <http://ops.fhwa.dot.gov/publications/fhwahop08024>