Delft University of Technology

Delft Center for Systems and Control

Technical report 12-002

Efficient network-wide model-based predictive control for urban traffic networks*

S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn

If you want to cite this report, please use the following reference instead:
S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn, "Efficient network-wide model-based predictive control for urban traffic networks," *Transportation Research Part C*, vol. 24, pp. 122–140, Oct. 2012.

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: `https://www.dcsc.tudelft.nl`

*This report can also be downloaded via `https://pub.deschutter.info/abs/12_002.html`

Efficient network-wide model-based predictive control for urban traffic networks

Shu Lin[a,b,*], Bart De Schutter[a], Yugeng Xi[b], Hans Hellendoorn[a]

[a] *Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD, Delft, The Netherlands*

[b] *Department of Automation, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, No. 800 Dongchuan Road, Minhang District, 200240, Shanghai, China*

Abstract

Traffic congestion has become a stringent issue in urban areas. Traffic control systems are designed to make a better use of the existing traffic infrastructures in order to improve traffic conditions. Along with the fast development of the transportation infrastructures, traffic networks become larger and more complex. Therefore, network-wide traffic control systems that can coordinate the whole network and improve the utilization of the entire traffic infrastructure, are highly required. In this paper, a structured network-wide traffic controller is presented based on Model Predictive Control (MPC) theory. Two macroscopic models are proposed to be the prediction model of the MPC controller. One is more accurate, but correspondingly requires more computation time; the other sacrifices a certain amount of accuracy, but is computationally much more efficient. Based on these two models, MPC controllers are developed. Simulation results show that the MPC controllers are capable of coordinating an urban traffic network, especially in the situations that the traffic flow is not spread evenly through the network. Through reducing the prediction model, the corresponding MPC controller exhibits less on-line computational burden, and thus becomes more applicable in practice. Therefore, it becomes possible for the control system to deal with complex urban road networks more efficiently.

Keywords:
Urban Traffic Network Control, Urban Traffic Modeling, Model Predictive Control

1. Introduction

Due to the high concentration of population and economic activities, a lot of traffic congestion arises in urban areas. Therefore, traffic management systems are installed to improve the performance of the existing urban transportation infrastructure, and thus

*Corresponding author

Email addresses: lisashulin@gmail.com (Shu Lin), b.deschutter@tudelft.nl (Bart De Schutter), ygxi@sjtu.edu.cn (Yugeng Xi), j.hellendoorn@tudelft.nl (Hans Hellendoorn)

to alleviate traffic congestion. Network-wide coordinated traffic management systems, which automatically and in real time determine appropriate control strategies based on the current and future traffic conditions, provide an effective control approach for improving the performance of the transportation services in cities.

Since the emergence of traffic control, traffic control strategies have gone through various developments, from isolated intersection control to coordinated control and from fixed-time control to traffic-responsive control (Papageorgiou et al., 2003; van Katwijk, 2008; Webster and Cobbe, 1966). Isolated intersection controllers have been well developed as local controllers. However, as the intersections are not isolated from each other, even though the local controller works properly, one cannot guarantee that no congestion is caused in the other regions within the traffic network. To avoid this phenomenon, it is necessary to have a coordinated control strategy for the whole traffic network. A number of coordinated urban network control strategies have already been developed (Hale, 2005; Papageorgiou et al., 2003; Robertson and Bretherton, 1991). Fixed-time coordinated control strategies make control decisions off-line based on the traffic flow data collected and stored in the past. Traffic-responsive coordinated control strategies can in real time measure the traffic states in the network, and adapt the control schemes according to the current measured traffic states. Model-based coordinated control strategies (Aboudolas et al., 2009; Boillot et al., 1992; Dotoli et al., 2006; Farges et al., 1983; Gartner, 1983; Hegyi et al., 2005; Papageorgiou et al., 2003; Sen and Head, 1997; van den Berg et al., 2007) can optimize the sequence of the future traffic control actions for an entire traffic network, and also are able to update in real time the optimal control decision sequence according to the currently detected traffic states.

Model-based control methods (including Model Predictive Control, MPC) use a prediction model and optimization to find the best control decisions for the network. There are already many model-based control strategies developed for urban traffic. In the 1980s and 1990s, a number of model-based optimization control strategies based on simple traffic models emerged: OPAC (Gartner, 1983), PRODYN (Farges et al., 1983), CRONOS (Boillot et al., 1992), and RHODES (Sen and Head, 1997). The model used in these control approaches are mainly simple traffic flow forecasting models based on the traffic data measured by upstream detectors. After that, model-based control strategies (including MPC) were developed based on more detailed traffic models (Aboudolas et al., 2009; Dotoli et al., 2006; van den Berg et al., 2007), and they obtained good control performance. These model-based control approaches all share a similar control framework, which contains model-based prediction, on-line optimization, and rolling time horizon. The detailed prediction models are able to describe and predict the traffic flow dynamics in the future, and as result they enables the controller to look ahead into the future to avoid myopic decisions. Both multiple intersections and multiple control measures can be easily coordinated through such a model-based optimization. Moreover, by using the rolling horizon procedure, feedback is introduced, which makes the controllers more robust to disturbances and model mismatch errors. All these advantages make the model-based control methods very attractive. However,

despite of all these advantages, the real-time feasibility[1] is the most common practical issue encountered when implementing MPC in practice.

When the number of controlled intersections gets larger, the optimization problems of model-based control strategies (including MPC) become too computationally complex to be solved on-line. To improve the real-time feasibility, the following methods can be considered. First, dividing the network into small subnetworks and building distributed controllers (Boillot et al., 2006; Di Febbraro et al., 2004; Gartner et al., 2001; Kotsialos and Papageorgiou, 2004; Mirchandani and Head, 2001). Second, solving the optimization problem off-line, such as optimizing a feedback regulator off-line and using it with real-time measured traffic states to derive control decisions (Aboudolas et al., 2009; Papageorgiou et al., 2003). Third, finding efficient solution methods for the on-line optimization problems, such as the feasible-direction method in Aboudolas et al. (2009, 2010), and the Mixed-Integer Linear Programming method in Lin et al. (2011). Fourth, reducing the computational complexity of the control model for urban traffic networks. In this paper, we mainly focus on the last approach, in which we simplify the nonlinear traffic prediction model to reduce the on-line computation time. Given the initial traffic states, traffic demands, and future control decisions, any model that can predict the future traffic states of the urban traffic network, can theoretically be used as a prediction model for MPC controllers. But, different models provide different levels of modeling detail and may yield a different computational complexity. It is very important for a prediction model to offer a good trade-off between accuracy and computational complexity, so that it can be fast enough for controlling urban traffic networks, while at the same time also guaranteeing effective control. Therefore, macroscopic traffic models, which do not describe the details of individual vehicles, but use more aggregated values like traffic flows and traffic densities, are suitable and accurate enough for real-time traffic control purposes. In this paper, two macroscopic urban traffic network models are proposed as prediction model for MPC controllers. One is more detailed, but more complex in computation. The other is a simplified model proposed to improve the real-time feasibility. This model is much faster than the previous model, while only losing a limited amount of accuracy.

This paper is organized as follows. In Section 2, the control structure for an urban road network is discussed. In Section 3, two macroscopic models for roads in an urban traffic network are proposed. Then MPC controllers are designed for urban traffic networks in Section 4. Section 5 presents a simulation-based case study, and Section 6 concludes the paper.

2. Network-wide traffic control structure

Urban road networks grow larger and larger, and become more and more complex. To better manage a large urban road network, the network can be divided into subnetworks with smaller scales. Thus, the network-wide traffic controller contains multiple subnetwork controllers, and requires a structure under which the subnetwork

[1] Real-time feasibility means that the on-line optimization problem can be solved fast enough so that the result is found before the time instant at which the controller should generate the next control signal.

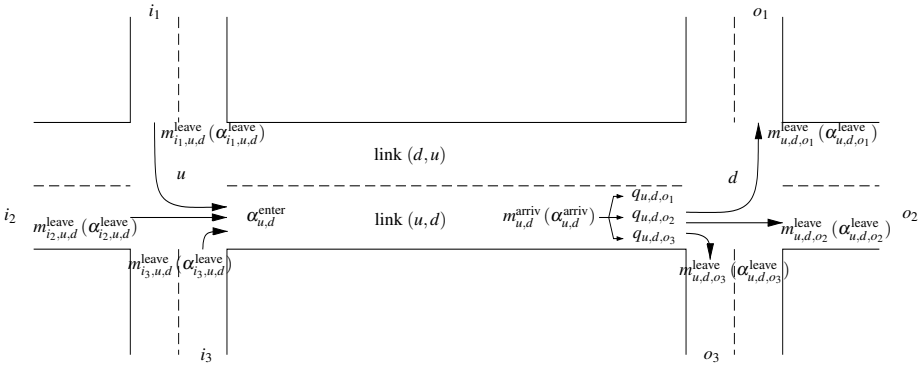


Figure 1: A link connecting two traffic-signal-controlled intersections

controllers coordinate with each other (Boillot et al., 2006; Di Febbraro et al., 2004; Gartner et al., 2001; Kotsialos and Papageorgiou, 2004; Mirchandani and Head, 2001).

The multi-subnetwork control structure makes a modular design for the controller of each subnetwork possible. As a result, both the flexibility and the scalability of the network-wide controller are increased. Furthermore, this approach also increases the reliability, the sustainability, and the robustness of the urban road network.

Under the multi-subnetwork control structure, centralized Model Predictive Control (MPC) controllers can be applied as the subnetwork controllers, which can guarantee a good control performance for the subnetworks and which can also provide coordination and cooperation among subnetwork controllers. Since MPC controllers can estimate and predict the future traffic state information based on the current traffic data, and exchange this information with other subnetwork controllers. However, due to the on-line computational burden of MPC, the subnetwork controller may become infeasible in practice, even with a small-scale traffic network. To develop an efficient MPC controller for the subnetworks is the premise of this research. Therefore, in this paper, we will mainly address the applicability of the subnetwork MPC controllers.

3. Two macroscopic urban road models

In this section we present the original urban traffic model proposed by Kashani and Saridis (1983) and then extended by van den Berg et al. (2007) and Lin and Xi (2008) (indicated as the BLX model), as well as a simplified model (called the S model) (Lin et al., 2009), which is related to the model of Aboudolas et al. (2009). However, compared to the latter model, the S model is able to consider more complex situations, such as the time delay needed for the vehicles running from the beginning of the link to the tail of the queues, and the available storage space in the downstream links. Now, we introduce some common notation for the BLX model and the S model.

3.1. Notation

Define J as the set of nodes (intersections), and L as the set of links (roads) in the urban traffic network. Link $(u,d) \in L$ is marked by its upstream node $u \in J$ and

downstream node $d \in J$. The sets of input and output nodes for link (u,d) are $I_{u,d} \subset J$ and $O_{u,d} \subset J$ (e.g., for the situation of Fig. 1 we have $I_{u,d} = \{i_1, i_2, i_3\}$ and $O_{u,d} = \{o_1, o_2, o_3\}$).

In order to describe the evolution of the models, we first define some variables (see also Fig. 1):

$I_{u,d}$	: set of input nodes of link (u,d),
$O_{u,d}$	: set of output nodes of link (u,d),
k	: simulation step counter,
$n_{u,d}(k)$	: number of vehicles in link (u,d) at step k,
$q_{u,d}(k)$	: queue length (expressed as the number of vehicles) at step k in link (u,d),
$q_{u,d,o}(k)$	: queue length of the sub-stream in link (u,d) turning to o at step k,
$m^{\text{leave}}_{u,d,o}(k)$	: number of cars leaving link (u,d) and turning to o at step k,
$m^{\text{arriv}}_{u,d}(k)$	: number of cars arriving *at the (end of the) queue* in link (u,d) at step k,
$m^{\text{arriv}}_{u,d,o}(k)$	: is the number of arriving cars *at the (end of the) queue* of the sub-stream going towards o in link (u,d) at step k,
$S_{u,d}(k)$	: available storage space of link (u,d) at step k expressed in number of vehicles,
$\alpha^{\text{leave}}_{u,d}(k)$	: average flow rate leaving link (u,d) at step k,
$\alpha^{\text{leave}}_{u,d,o}(k)$	: leaving average flow rate of the sub-stream going towards o in link (u,d) at step k,
$\alpha^{\text{arriv}}_{u,d}(k)$	: average flow rate arriving at the end of the queue in link (u,d) at step k,
$\alpha^{\text{arriv}}_{u,d,o}(k)$	: arriving average flow rate of the sub-stream going towards o,
$\alpha^{\text{enter}}_{u,d}(k)$	: average flow rate entering link (u,d) at step k,
$\beta_{u,d,o}(k)$	: relative fraction of the traffic in link (u,d) turning to o at step k,
$\mu_{u,d}$	: saturated flow rate leaving link (u,d),
$g_{u,d,o}(k)$	: green time length during step k for the traffic stream in link (u,d) going towards o,
$b_{u,d,o}(k)$	: boolean value indicating whether the traffic signal at intersection d for the traffic stream in link (u,d) turning to o is green (1) or red (0) at step k,
$v^{\text{free}}_{u,d}$	: free-flow vehicle speed in link (u,d),
$C_{u,d}$	: storage capacity of link (u,d) expressed in number of vehicles,
$N^{\text{lane}}_{u,d}$	: number of lanes in link (u,d),
$\Delta c_{u,d}$	: offset between node u and node d, i.e. the time difference between the beginnings of the cycle times of the two intersections,
l_{veh}	: average vehicle length.

3.2. BLX model

In the BLX model of van den Berg et al. (2007) and Lin and Xi (2008) a queue is modeled as follows. For the sake of simplicity, the assumption is made that at an intersection the cars going to the same destination link move into the correct lane, so that they do not block the traffic flows going to other destination links. For each lane

(or destination link), a separate queue is constructed (with queue lengths denoted by q). Further, the simulation time step T_s is typically set to 1 s and cars arriving at the end of a queue in simulation period $[kT_\mathrm{s},(k+1)T_\mathrm{s})$ are allowed to cross the intersection in that same period (provided they have green, there is enough space in the destination link, and there are no other restrictions).

Consider link (u,d) (see Fig. 1). For each $o \in O_{u,d}$ the number of cars leaving link (u,d) for destination o in the period $[kT_\mathrm{s},(k+1)T_\mathrm{s})$ is given by

$$m^{\mathrm{leave}}_{u,d,o}(k) = \begin{cases} 0 & \text{if } b_{u,d,o}(k) = 0 \\ \max\big(0, \min(q_{u,d,o}(k) + m^{\mathrm{arriv}}_{u,d,o}(k), & \\ \quad S_{d,o}(k),\, \beta_{u,d,o}(k)\cdot\mu_{u,d}\cdot T_\mathrm{s})\big) & \text{if } b_{u,d,o}(k) = 1\,. \end{cases}$$

The number of vehicles arriving at the end of the queue in link (u,d) is given by the number of vehicles entering the link via the upstream intersection delayed by the time $\tau(k)$ needed to drive from the upstream intersection to the end of the queue in the link. The time delay can be calculated by

$$\tau(k) = \frac{\big(C_{u,d} - q_{u,d}(k)\big)\cdot l_\mathrm{veh}}{N^{\mathrm{lane}}_{u,d}\cdot v^{\mathrm{free}}_{u,d}\cdot T_\mathrm{s}}. \tag{1}$$

Next, considering the relationship between average flow rate and number of vehicles

$$\alpha_{u,d} = m_{u,d}/T_\mathrm{s}, \tag{2}$$

and according to (A.21), the number of vehicles arriving at the end of queues is updated as follows:

$$m^{\mathrm{arriv}}_{u,d}(k) = \frac{T_\mathrm{s}-\gamma(k)}{T_\mathrm{s}}\cdot\sum_{i\in I_{u,d}} m^{\mathrm{leave}}_{i,u,d}\left(k-\delta(k)\right) + \frac{\gamma(k)}{T_\mathrm{s}}\cdot\sum_{i\in I_{u,d}} m^{\mathrm{leave}}_{i,u,d}\left(k-\delta(k)-1\right), \tag{3}$$

where $\delta(k)$ and $\gamma(k)$ can be obtained from the time delay through formulas (A.2) and (A.3) derived in the appendix.

The fraction of the arriving traffic in link (u,d) turning to $o \in O_{u,d}$ is

$$m^{\mathrm{arriv}}_{u,d,o}(k) = \beta_{u,d,o}(k)\cdot m^{\mathrm{arriv}}_{u,d}(k)\ . \tag{4}$$

The new queue lengths are given by the old queue lengths plus the arriving traffic minus the leaving traffic:

$$q_{u,d,o}(k+1) = q_{u,d,o}(k) + m^{\mathrm{arriv}}_{u,d,o}(k) - m^{\mathrm{leave}}_{u,d,o}(k) \tag{5}$$

for each $o \in O_{u,d}$ with

$$q_{u,d}(k) = \sum_{o\in O_{u,d}} q_{u,d,o}(k)\ . \tag{6}$$

The new available storage stage depends on the number of cars that enter and leave the link in the period $[kT_\mathrm{s},(k+1)T_\mathrm{s})$:

$$S_{u,d}(k+1) = S_{u,d}(k) - \sum_{i\in I_{u,d}} m^{\mathrm{leave}}_{i,u,d}(k) + \sum_{o\in O_{u,d}} m^{\mathrm{leave}}_{u,d,o}(k)\,. \tag{7}$$

The number of vehicles in link (u,d) in the period $[kT_s,(k+1)T_s)$ is then easily derived as

$$n_{u,d}(k+1) = C_{u,d} - S_{u,d}(k+1) \quad . \tag{8}$$

3.3. Simplified model (S model)

In order to increase the efficiency of computation, the BLX model is simplified by enlarging the simulation sampling time interval from 1 s to a cycle time. In the S model (Lin et al., 2009), every intersection takes its cycle time as the simulation time interval. The cycle times for intersection u and d, which are denoted by c_u and c_d respectively, can be different from each other. Moreover, the S model works with (average) flow rates rather than with numbers of cars for describing flows leaving or entering links.

Taking the cycle time c_d as the length of the simulation time interval for link (u,d) and k_d as the corresponding time step counter, the number of the vehicles in link (u,d) is updated according to the input and output average flow rate over c_d at every time step k_d by

$$n_{u,d}(k_d+1) = n_{u,d}(k_d) + \left(\alpha_{u,d}^{\text{enter}}(k_d) - \alpha_{u,d}^{\text{leave}}(k_d)\right) \cdot c_d \quad . \tag{9}$$

The leaving average flow rate is the sum of the leaving flow rates turning to each output link:

$$\alpha_{u,d}^{\text{leave}}(k_d) = \sum_{o \in O_{u,d}} \alpha_{u,d,o}^{\text{leave}}(k_d) \quad . \tag{10}$$

The leaving average flow rate over c_d is determined by the capacity of the intersection, the number of cars waiting or arriving, and the available space in the downstream link:

$$\begin{aligned} \alpha_{u,d,o}^{\text{leave}}(k_d) = \min \Big(& \beta_{u,d,o}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o}(k_d)/c_d, \\ & \frac{q_{u,d,o}(k_d)}{c_d} + \alpha_{u,d,o}^{\text{arriv}}(k_d), \\ & \frac{\beta_{u,d,o}(k_d)}{\sum_{u \in I_{d,o}} \beta_{u,d,o}(k_d)} \cdot \frac{C_{d,o} - n_{d,o}(k_d)}{c_d} \Big) \quad . \end{aligned} \tag{11}$$

The leaving flow rate is the minimum value of three flow rate values, average saturated flow rate, average unsaturated flow rate, and average over-saturated flow rate, which are given respectively by the three terms in (11). The first term calculates the average saturated flow rate, which depends on the saturation flow rate $\beta_{u,d,o}(k_d)\mu_{u,d}$ and the green time duration; the second term calculates the average unsaturated flow rate based on the vehicles waiting in and arriving at the queues; the third term calculates the average over-saturated flow rate depending on the proportional storage capacity of the downstream link.

The number of vehicles waiting in the queue turning to o is updated as

$$q_{u,d,o}(k_d+1) = q_{u,d,o}(k_d) + \left(\alpha_{u,d,o}^{\text{arriv}}(k_d) - \alpha_{u,d,o}^{\text{leave}}(k_d)\right) \cdot c_d. \tag{12}$$

Here we made the assumption that the vehicles getting into a link do not immediately separate for their turning directions. They run on the link freely until they reach the tail of the waiting vehicle queues. Then they will join the queues of the turning direction they intend to go. Thus, the number of vehicles waiting in link (u,d) is

$$q_{u,d}(k_d) = \sum_{o \in O_{u,d}} q_{u,d,o}(k_d) \ . \tag{13}$$

The flow rate entering link (u,d) will arrive at the end of the queues after a time delay

$$\tau(k_d) = \frac{\left(C_{u,d} - q_{u,d}(k_d)\right) \cdot l_{\text{veh}}}{N_{u,d}^{\text{lane}} \cdot v_{u,d}^{\text{free}} \cdot c_d}, \tag{14}$$

then according to (A.21), the delayed flow rate arriving at the end of queues is

$$\alpha_{u,d}^{\text{arriv}}(k_d) = \frac{c_d - \gamma(k_d)}{c_d} \cdot \alpha_{u,d}^{\text{enter}}(k_d - \delta(k_d)) + \frac{\gamma(k_d)}{c_d} \cdot \alpha_{u,d}^{\text{enter}}(k_d - \delta(k_d) - 1), \tag{15}$$

where $\delta(k_d)$ and $\gamma(k_d)$ are derived from the time delay by formulas (A.2) and (A.3).

Before reaching the tail of the waiting queues in link (u,d), the flow rate of arriving vehicles has to be divided according to the different destination links by multiplying it with the turning rates:

$$\alpha_{u,d,o}^{\text{arriv}}(k_d) = \beta_{u,d,o}(k_d) \cdot \alpha_{u,d}^{\text{arriv}}(k_d). \tag{16}$$

The flow rate entering link (u,d) is made up from the flow rates from all the input links:

$$\alpha_{u,d}^{\text{enter}}(k_d) = \sum_{i \in I_{u,d}} \alpha_{i,u,d}^{\text{leave}}(k_d). \tag{17}$$

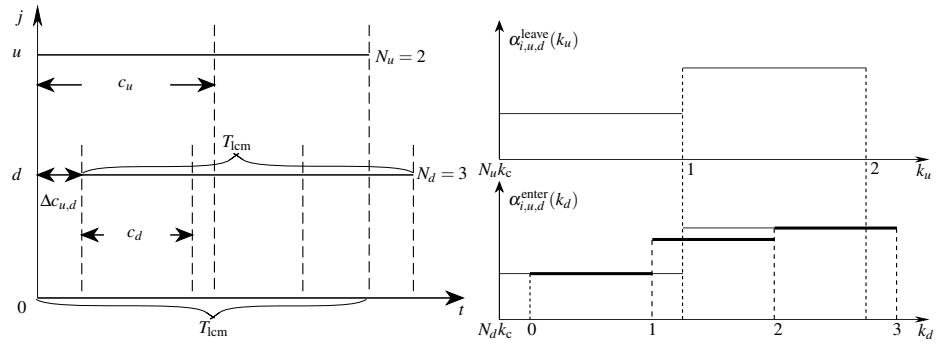
Thus the flow rate entering link (u,d) is provided by the combination of the flow rates leaving the upstream links. Recall that we have different cycle times between upstream and downstream intersections, and so the simulation time steps for the downstream links and the upstream links are not the same. Note that from (11) we can obtain the flow rates leaving the upstream links, $\alpha_{i,u,d}^{\text{leave}}(k_u)$ $(i \in I_{u,d})$, with step counter k_u, while what we need in (17) are the flow rates $\alpha_{i,u,d}^{\text{leave}}(k_d)$ $(i \in I_{u,d})$ with step counter k_d. Hence, some operations have to be carried out to synchronize the leaving and entering flow rates. To synchronize intersections, a least common multiple time interval is defined as

$$T_{\text{lcm}} = N_d \cdot c_d \qquad \text{for all } d \in J, \tag{18}$$

with N_d an integer, as illustrated in Fig. 2(a). This least common multiple time interval T_{lcm} is the smallest feasible control time interval for the MPC controllers of urban traffic networks.

Now we show how the flow rates expressed in the timing of intersection u can be recast into the timing of intersection d. As Fig. 2(b) shows, first, we transform the discrete-time leaving flow rates from the upstream links into continuous time using the zero-order hold strategy, as

$$\alpha_{i,u,d}^{\text{leave,Cont.}}(t) = \alpha_{i,u,d}^{\text{leave}}(k_u), \quad k_u \cdot c_u \le t < (k_u + 1) \cdot c_u, \tag{19}$$



(a) Relationship between cycle times and control time interval

(b) Illustration for synchronizing flow rates

Figure 2: Synchronization of upstream and downstream intersections

and then sample them again to obtain the average flow rates in time step k_d used by the downstream link:

$$\alpha^{\text{leave}}_{i,u,d}(k_d) = \frac{1}{c_d} \int_{k_d \cdot c_d + \Delta c_{u,d}}^{(k_d+1)\cdot c_d + \Delta c_{u,d}} \alpha^{\text{leave,Cont.}}_{i,u,d}(t)\,\mathrm{d}t \quad , \tag{20}$$

where $\Delta c_{u,d}$ represents the offset time between the cycle times of the upstream and the downstream intersections at the beginning of every control time step.

4. MPC subnetwork traffic controller

Model Predictive Control (Camacho and Bordons, 1995; Garcia et al., 1989; Rawlings and Mayne, 2009; Richalet et al., 1978) is a methodology that implements and repeatedly applies optimal control in a rolling horizon way. As Fig. 3 shows, in each control step k_c, an optimal control problem is solved over a prediction horizon, but only the first control sample of the optimal control sequence is implemented. Next, the horizon is shifted one sample and the optimization is restarted again with new information of the measurements. The optimization is redone based on the prediction model of the process and an estimation of the disturbance inputs.

Similar to optimal control, MPC is able to predict and find the optimal solution for the future. Different from optimal control, instead of doing the optimization over an infinite time horizon, MPC optimizes the future control actions over a finite time horizon, and repeatedly solves such an optimization problem online in a rolling horizon way. This rolling horizon procedure closes the control loop, which makes MPC robust to uncertainties of the process. The uncertainties can be caused by unpredictable disturbances, (slow) variation over time of the parameters, and model mismatches in the prediction model. Due to the existence of these uncertainties, predicting errors are unavoidable for all the models. But the errors can be made up to some extent by selecting a proper control approach, such as MPC. In addition, MPC is able to easily deal with multi-input and multi-output problems with constraints. Another advantage

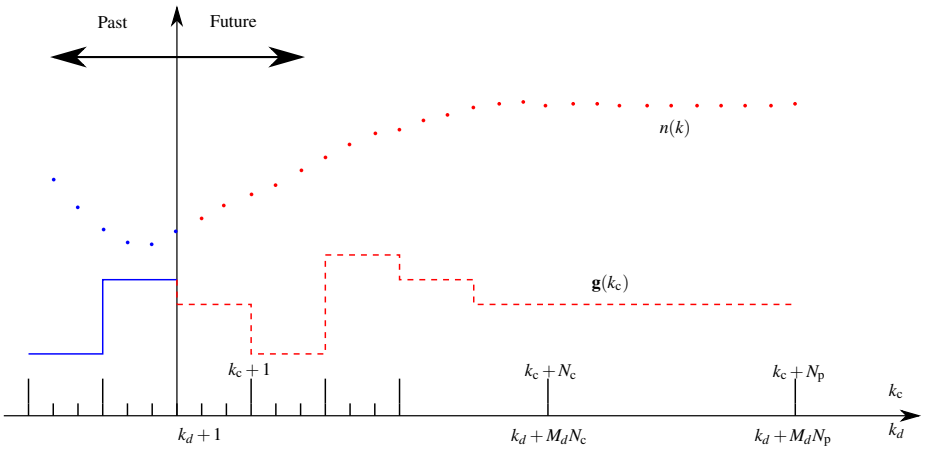


Figure 3: Principle of the receding horizon used in an MPC

of MPC is that it is modular such that one can freely select and replace the prediction model based on the control requirements or on the trade-off between accuracy and computational complexity.

In order to control the urban traffic subnetwork, a common control time interval needs to be defined, so that intersections within the subnetwork can communicate with each other and be synchronous. Thus, T_c is defined as

$$T_c = N \cdot T_x, \tag{21}$$

where T_x is either the simulation time interval T_s for the BLX model, or the least common multiple time interval T_{lcm} of the cycle times for all the intersections within the subnetwork for the S model, and N is an integer. For a given simulation time step counter k_d of intersection $d \in J$, the corresponding value of k_c is given by

$$k_c(k_d) = \left\lfloor \frac{k_d}{N \cdot N_d} \right\rfloor, \tag{22}$$

where $\lfloor x \rfloor$ with x a real number denotes the largest integer less than or equal to x. On the other hand, if we define $M_d = N \cdot N_d$, then a given value k_c of the control time step corresponds to the set $\{k_c M_d, k_c M_d + 1, \cdots, (k_c+1)M_d - 1\}$ of simulation time steps for intersection d. As Fig. 3 shows,the simulation time step k_d for the prediction model does not have to be the same as the control time step k_c, and so the model is able to update more frequently than the MPC controller.

Fig. 4 shows the structure of the MPC controller. The control process can be described by the following elements:

1. **Prediction model**. A model can be selected as the prediction model for MPC controllers, if it can predict the future traffic states used for evaluating the objective function based on the information of current measured traffic states, the predicted future network traffic demands (i.e. future input traffic flows to the network), and the future control inputs. Based on this prediction model, it becomes

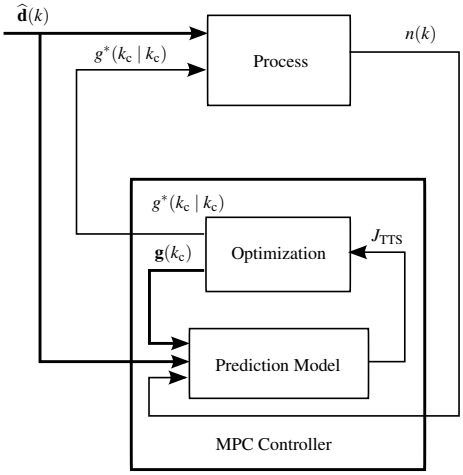


Figure 4: The framework of the MPC controller

possible to find the optimal control actions for the future through optimization. However, the control performance of MPC controllers can be either improved or deteriorated, when the predictions of the model are good or poor.

The BLX and S model described in Section 3 are able to predict the future traffic states through the evolutionary formulae based on the current traffic state information, the future network traffic demands, and the future control inputs. The future network traffic demands can be estimated according to the historical data, or provided by the neighbor subnetwork controllers. In this context, both the BLX and the S model can be selected as prediction models for MPC controllers, and they can be generally described as a nonlinear model

$$n_{u,d}(k_d+1) = f\big(n_{u,d}(k_d), g_d(k_c), d_{u,d}(k_d)\big) \quad \text{for all } (u,d) \in L, \tag{23}$$

where $n_{u,d}(k_d)$ is the traffic state (i.e. the number of vehicles in a link at simulation time step k_d), which is estimated for evaluating the objective function; $d_{u,d}(k_d)$ is the predicted network traffic demand for link (u,d) at time step k_d, which is specifically the future input traffic flow rate to the subnetwork; and $g_d(k_c)$ is the future control input of intersection d, e.g. the green time lengths.

2. **Optimization problem**. Based on the prediction model, optimization problems over a given prediction horizon are repeatedly solved online. Here, specifically given a prediction horizon N_p, the future traffic states for link (u,d) are predicted at simulation time step k_d as

$$\widehat{\mathbf{n}}_{u,d}(k_d) = [\widehat{n}_{u,d}(k_d+1|k_d)\ \widehat{n}_{u,d}(k_d+2|k_d)\cdots\widehat{n}_{u,d}(k_d+M_dN_p|k_d)]^T, \tag{24}$$

based on the predicted network traffic demands for link (u,d) at simulation time step k_d

$$\widehat{\mathbf{d}}_{u,d}(k_d) = [\widehat{d}_{u,d}(k_d|k_d)\ \widehat{d}_{u,d}(k_d+1|k_d)\cdots\widehat{d}_{u,d}(k_d+M_dN_p-1|k_d)]^T, \tag{25}$$

and based on the future traffic control inputs for node d at control step k_c

$$\mathbf{g}_d(k_c) = [g_d^T(k_c|k_c)\ g_d^T(k_c+1|k_c)\cdots\ g_d^T(k_c+N_p-1|k_c)]^T\ , \tag{26}$$

where $g_d(k_c+j|k_c)$ denotes the control input at the jth control step in the future counted from the current control time step k_c. Assume the nodes in J are numbered as $J=\{1,2,\ldots,\Phi\}$, then the optimized control input for the subnetwork is a vector expressed as $\mathbf{g}(k_c)=\left[\mathbf{g}_1^T(k_c)\ \mathbf{g}_2^T(k_c)\ \cdots\ \mathbf{g}_\Phi^T(k_c)\right]^T$. Therefore, the online optimization problem of MPC can be expressed as

$$\begin{aligned}
&\min_{\mathbf{g}(k_c)} J = \min_{\mathbf{g}(k_c)} \sum_{(u,d)\in L} J_{u,d}\left(\widehat{\mathbf{n}}_{u,d}(k_d), \mathbf{g}_d(k_c)\right)\\
\text{s.t.}\quad &\widehat{n}_{u,d}(k_d) = n_{u,d}(k_d);\\
&\widehat{n}_{u,d}(k_d+j+1) = f\left(\widehat{n}_{u,d}(k_d+j), g_d\left(k_c(k_d+j)\right), d_{u,d}(k_d+j)\right),\\
&k_c(k_d+j) = \left\lfloor \frac{k_d+j}{M_d} \right\rfloor,\\
&\text{for } j=0,\ldots,M_dN_p-1, \text{ for all } (u,d)\in L;\\
&\Phi(\mathbf{g}(k_c))=0 \quad \text{(cycle time constraints)};\\
&\mathbf{g}_{\min} \le \mathbf{g}(k_c) \le \mathbf{g}_{\max},
\end{aligned} \tag{27}$$

where $J_{u,d}$ is the objective function for link (u,d), and L is the set of links in the urban traffic subnetwork. A control horizon N_c ($N_c < N_p$) can be further defined to reduce the number of control variables that need to be optimized, as a consequence

$$g_d(k_c+i|k_c) = g_d(k_c+N_c-1|k_c) \text{ for } i=N_c,\cdots,N_p-1. \tag{28}$$

The optimization problem in (27) in general has a nonlinear, non-convex objective function and nonlinear, non-convex constraints, which means that there can be multiple local minima. To solve the optimization problem one therefore has to apply a global optimization method (such as simulated annealing, genetic algorithms, or tabu search) or a multi-start local optimization method, like Sequential Quadratic Programming (SQP) (Pardalos and Rosen, 1987, Chap 5). In the multi-start SQP approach adopted later on this paper, the SQP algorithm is run several times, each time from a different initial starting point. The resulting local optima are recorded and the one that has the lowest objective function value is selected as the final (optimal) solution.

3. **Rolling horizon**. After deriving the optimal control input $\mathbf{g}^*(k_c)$ from the online optimization, the first sample of the optimal control sequence, i.e.

$$g^*(k_c|k_c) = [g_1^{*T}(k_c|k_c)\ g_2^{*T}(k_c|k_c)\ \cdots\ g_\Phi^{*T}(k_c|k_c)]^T, \tag{29}$$

is transferred to the process and is implemented. When arriving to the next control step k_c+1, the whole time horizon is shifted one step forward, and the optimization over the new prediction horizon starts over again, based on the prediction model that is fed with the real measured traffic states obtained from the

process. This rolling horizon scheme closes the control loop, enables the system get feedback from the real traffic network, and makes the MPC controller adaptive to the uncertainties and disturbances of the traffic environment.

Remark. *The objective function in* (27) *is in the sequel of this paper selected as the Total Time Spent (TTS) of the subnetwork, i.e.*

$$\min_{\mathbf{g}(k_c)} J = \min_{\mathbf{g}(k_c)} \sum_{(u,d)\in L} \sum_{k_d=k_c M_d+1}^{(k_c+N_p)M_d} T_y \cdot \widehat{n}_{u,d}(k_d)\,, \tag{30}$$

where T_y is the simulation time interval T_s for the BLX model, or the cycle time c_d of intersection d for the S model.

5. Case studies

MPC has a comparatively high on-line computational complexity, which may impact its applicability. The on-line computational complexity can be decreased by increasing the efficiency, i.e. the computational speed of the prediction model. Therefore, macroscopic models are selected as prediction model, and within this class the S model is proposed to further reduce the computation time. In this section, we are going to design experiments to assess whether or not the prediction model selected is fast enough, meanwhile also accurate enough for control purposes, and also to test the functionality of the designed MPC controllers. The simulated urban road subnetwork is shown in Fig. 5. Nodes marked as "Sx" are the source nodes where traffic flows enter and leave the subnetwork, and also where subnetworks connect with each other.

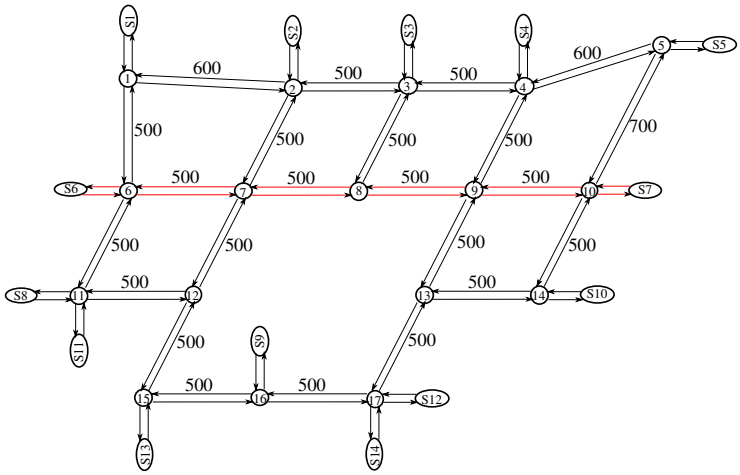


Figure 5: An urban road subnetwork with a main street (string S6-6-7-8-9-10-S7)

5.1. Model test

To evaluate the effectiveness of the proposed urban traffic models, the microscopic model CORSIM developed by FHWA (2001) is employed to simulate the real traffic. The comparisons are performed with two MOEs (Measure of Effectiveness) defined in CORSIM, "content" and "trips". "content" is the cumulative count of vehicles on a link, accumulated every time step. In fact, when multiplied with the simulation time step, the "content" corresponds to the TTS of a link, i.e. the output performance of the MPC controller. "trips" is the number of vehicles that have been discharged from a link since simulation begins. "trips" actually represents the Total Vehicle Departures (TVD) of a link, which illustrates the control effect of the corresponding control inputs (i.e. green time durations, cycle time, offset, phase) on the traffic flow of the link. For control purposes, a decisive factor considered when selecting a model is whether that model can provide a close enough dynamic relation of the control inputs and the control outputs. Thus both the MOEs are chosen as performance indicators to evaluate the urban road subnetwork model.

The layout of the simulated traffic network and the link lengths (in meter) are shown in Fig. 5. Each link has 3 lanes. The vehicle turning rates are considered to be constant, 1/3 for left, right, and through movements, since CORSIM only allows to define turning ratios instead of specifying route choices. The free-flow speed is 30 km/h, and the saturation flow rate is 2000 veh/h per lane. The network input flow rates are set equal to each other and constant in time, i.e. 1500 veh/h.

A fixed-time control strategy is executed in the traffic network, where the phases, the cycle times, and the green time durations are all constant. The fixed-time signals are designed based on the data for the saturated scenario (Papageorgiou, 1983; van Katwijk, 2008), i.e. the green times are proportional to the traffic demands from each direction, which depend on the saturated flow rates and the turning rates under the saturated scenario. More specifically, define r_p as the maximum saturation flow rate of phase p

$$r_p = \max_{l \in L_p} \{\mu_l\}, \tag{31}$$

where L_p is the set of lanes of phase p and μ_l is saturation flow rate for lane l. Then, define R as the summation of all maximum saturation flow rates corresponding to each of the n phases in one cycle:

$$R = \sum_{p=1}^{n} r_p. \tag{32}$$

For each phase p, its optimum green time, t_p^{green}, is then calculated by distributing the total available green time, i.e. $C - Y$ (C is the cycle time length, Y is the total yellow time length), in proportion to its maximum saturation flow rate:

$$t_p^{\text{green}} = \frac{r_p}{R}(C - Y). \tag{33}$$

Here, we define the cycle times as 60 s for all intersections in Fig. 5, except 30 s for Intersection 8, and all the intersections having two phases. Then the green time durations for the phases are identically split within the cycle time based on the method above, i.e. the green time durations are 30 s and 30 s for the two phases of all the intersections

in Fig. 5, except 15 s and 15 s for Intersection 8. According to Little et al. (1981), the offset value is derived based upon the distance between the master intersection and the desired travel speed, so as to maximize the bandwidth of the traffic signals on the arterial. Taking the string S6-6-7-8-9-10-S7 in Fig. 5 for example, the distances between successive intersections are all 500 m and the free flow speed is 30 km/h, then the desired progression time from one intersection to the next intersection is 60 s, which is the integer multiple of the cycle time length. Therefore, the maximized bandwidth of the traffic signals on the string S6-6-7-8-9-10-S7 is shown in Fig. 6, and the optimal offsets between successive intersections are all zero on the string S6-6-7-8-9-10-S7. Similarly, the offsets for other arterial are derived using the same method. Due to the setup of the network, the optimal offsets for most of the intersections are zero.

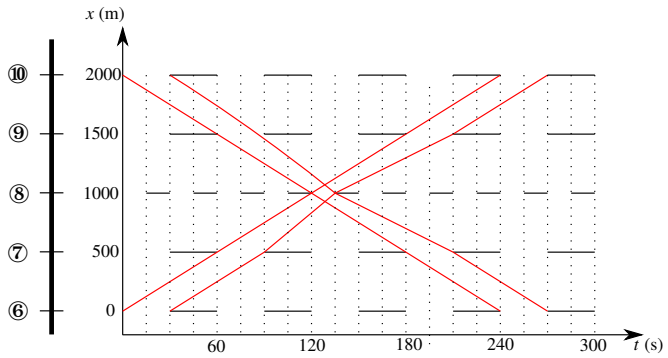


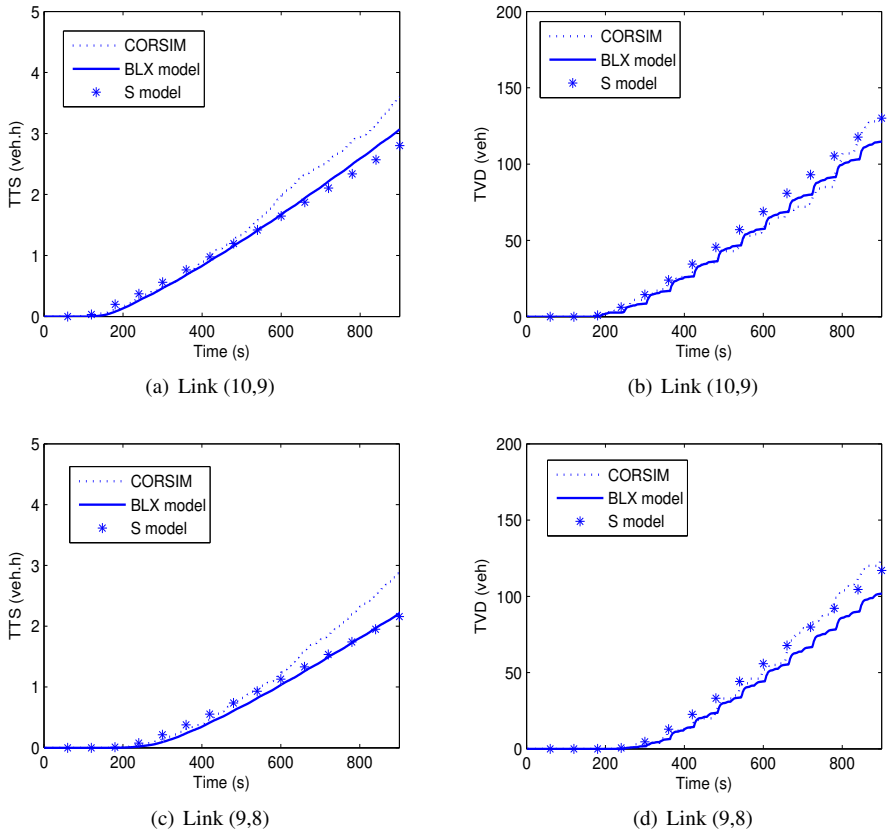
Figure 6: The maximized bandwidth for traffic signals on the string S6-6-7-8-9-10-S7

Macroscopic models are usually approximations of traffic dynamics, and they ignore some details of individual vehicles and make some simplifications; so macroscopic traffic models are in general not as accurate as the models with higher level-of-detail. However, this statement does not always hold in practice. On some occasions, macroscopic modeling approaches may provide better results than modeling approaches with a higher level-of-detail. Nevertheless, CORSIM, which describes details of individual vehicles, is selected to simulate the real traffic environment.

Fig. 7 shows the comparison of the S model, the BLX model, and CORSIM on both the output index (TTS) and the input index (TVD). As Fig. 7 shows, although they are macroscopic models, the BLX model and S model are still able to provide curves of the two indices that are consistent with that of microscopic traffic simulator, CORSIM, for both[2] Link (10,9) and Link (9,8). But the curves of the BLX model and S model for these input and output indices will drift away from the curve of CORSIM as the time grows. The reason for this is that the longer the macroscopic models run, the more errors will be accumulated for the models for the sake of neglecting some detailed driving behavior of individual vehicles.

The S model is a reduced model that is obtained through simplifying a more detailed model, the BLX model, and thus it inevitably sacrifices some accuracy for its

[2]These links are two representative links, for the rest links, the results are similar.



(a) Link (10,9) (b) Link (10,9)

(c) Link (9,8) (d) Link (9,8)

Figure 7: TTS and TVD comparison of the S model, BLX model, and CORSIM for different links

efficiency. But, Fig. 7 shows that the S model is still consistent enough with the BLX model in both the input and output indices. Therefore, both macroscopic models, the S model and the BLX model, are accurate enough to be selected as the prediction model of MPC controllers.

During this simulation, the computation speeds of the BLX and the S model are also compared. The results turn out to be that the S model is much faster than the BLX model (the computation time is reduced by 97.4% compared with the BLX model), which can further increase the on-line feasibility of MPC controllers, but with limited loss of the control performance.

5.2. Urban road subnetwork control using MPC

In fact, the BLX-based MPC controller was already compared with systems that are similar to existing dynamic urban traffic control systems like SCOOT and UTOPIA/SPOT in van den Berg et al. (2007), and the results illustrate the potential benefits of the MPC approach and motivate further development and improvement of the MPC control

method. In this paper, two MPC controllers are designed for the urban road subnetwork shown in Fig. 5, taking the BLX model and the S model as prediction models respectively, and they are compared based on the same simulation setup. The structure of the BLX-based MPC controller and the S-based MPC controller is shown in Fig. 8. Here, the real traffic environment is simulated by CORSIM, these traffic control strategies are implemented into CORSIM to generate control actions, and then the performance of these strategies is compared. A fixed-time (FT) controller, as mentioned in Section 5.1, is adopted as a benchmark for evaluating the improvement of these two model-based control strategies.

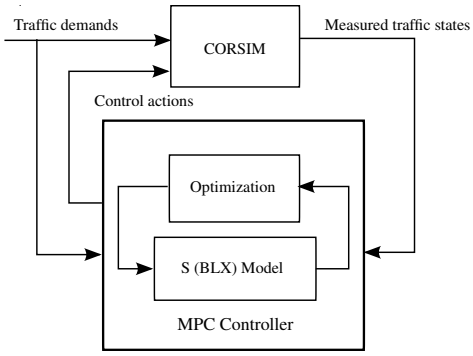


Figure 8: Illustration of MPC controllers

The setup of the network and the fixed-time controller are the same as that described in Section 5.1. During the experiments, the simulation time interval of the BLX model is set to 1 s, while in the S model, the simulation time intervals are 60 s for all intersections, except 30 s for Intersection 8. For both MPC controllers, the control time interval T_c is 120 s, the prediction horizon N_p is 5 (i.e. 10 min), and $N_c = N_p$. All the simulations implemented with different control strategies run for the same simulation period, 1 h. The Total Time Spent (TTS) is the control objective of the MPC controllers, but the performance indicators used for the evaluation are selected as the TTS and the Total Delay Time (TDT). The performance criterion TTS represents the accumulated amount of time spent by all the vehicles inside the road network since the beginning of the simulation. TDT is the difference between the total travel time of all vehicles inside the road network since the beginning of the simulation and the total free-flow travel time (i.e., the travel time when the vehicles would all be running at the free-flow speed), which is actually the total amount of time that the vehicles are delayed. Two traffic scenarios are considered:

1. *Balanced scenario:* The traffic demands (traffic flows) from all the source nodes flowing into the subnetwork are the same, and they all first increase and then decrease with time as shown in Fig. 9.
2. *Unbalanced scenario:* The traffic demands of all the source nodes are very low (500 veh/h), but the traffic demands for source nodes S6 and S7 are given very high (3000 veh/h). Therefore, the road between S6 and S7 becomes a busy and main street of the subnetwork, called “string” as shown in Fig. 5.

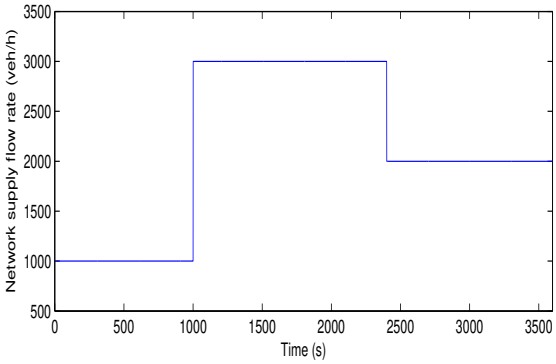
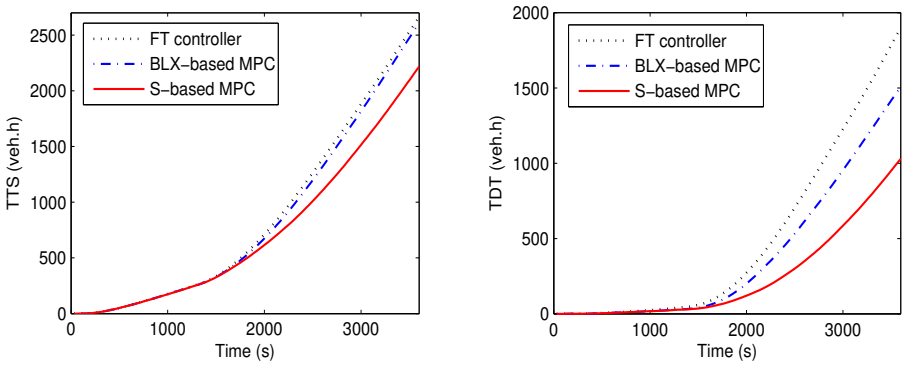


Figure 9: The variation of the supply flow rates for the network



(a) (b)

Figure 10: TTS and TDT comparisons for the subnetwork in Fig. 5 of the S-based MPC, the BLX-based MPC, and the FT controller at every control time step in the balanced scenario

For the balanced scenario, Fig. 10 illustrates the comparison of the two control performance indicators for all the control strategies (S-based MPC, BLX-based MPC, and FT controller). As Fig. 10(a) shows, the TTS of the S-based MPC controller is lower than the TTS of the FT controller and the BLX-based controller when the network traffic supply is comparatively high (i.e. during the later part of the simulation). But, when the network traffic supply stays low, the differences between the influences on TTS for different control strategies can be hardly seen (i.e. at the beginning of the simulation). Comparatively, the control effects on TDT for the control strategies show obvious differences. As Fig. 10(b) shows, compared with the FT controller, the TDT of the urban network is reduced by 22.6% for the BLX-based MPC controller, and even more, by 51.2%, for the S-based MPC controller. This indicates that the TDT incurred by the vehicles traveling with a speed lower than the free-flow speed can be significantly reduced by the MPC controllers with TTS as the cost function, which also means that

vehicles can run more smoothly through the urban traffic network.

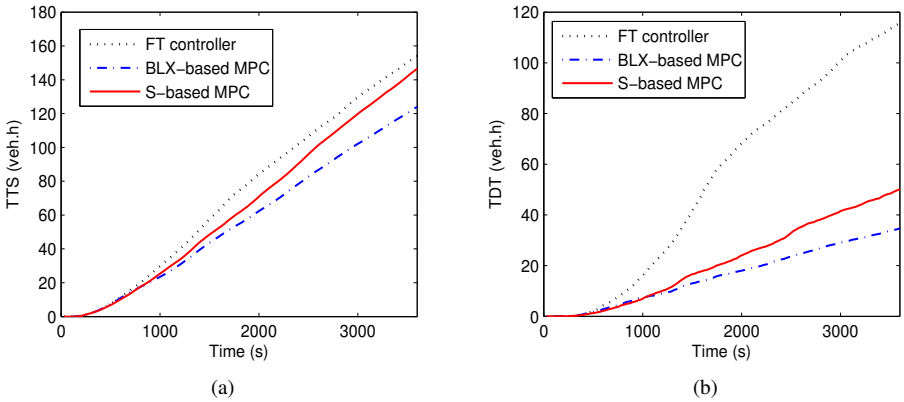


Figure 11: TTS and TDT comparisons for the string (S6-6-7-8-9-10-S7 in Fig. 5) of the S-based MPC, the BLX-based MPC, and the FT controller at every control time step in the unbalanced scenario

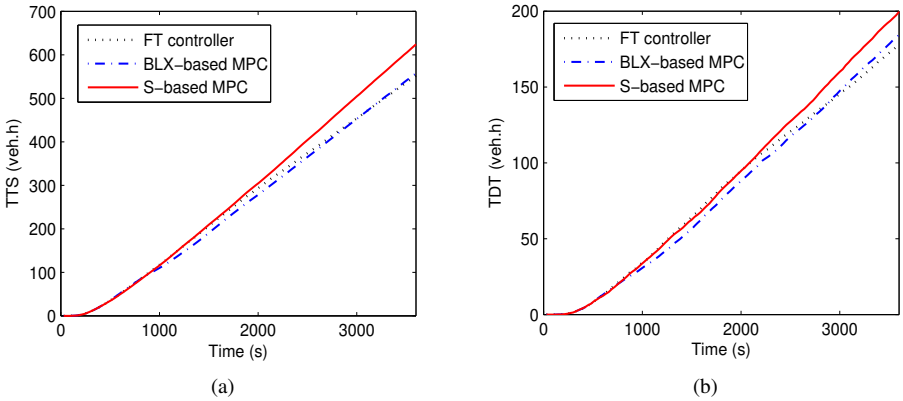


Figure 12: TTS and TDT comparisons for the links outside the string of the S-based MPC, the BLX-based MPC, and the FT controller at every control time step in the unbalanced scenario

For the unbalanced scenario, the TTS and the TDT are plotted respectively for the string (string S6-6-7-8-9-10-S7 in Fig. 5) and the rest of the links of the subnetwork in Fig. 11 and Fig. 12. In the string, where the traffic demand is very high, the MPC controllers improve the TTS slightly, but they reduce the TDT very obviously, by 70.5% for the BLX-based MPC controller, and by 59.1% for the S-based MPC controller, which indicates that the vehicles can pass the string much more efficiently and smoothly than the vehicles controlled by the FT controller. However, this improvement is obtained through using the other links of the subnetwork as a "buffer", i.e. the MPC controllers try to optimize the performance of the traffic flow in the string by sacrificing some

Table 1: TTS (veh·h) and TDT (veh·h) comparison for different control strategies in the two traffic scenarios

Strategy	FT		BLX-based MPC		S-based MPC	
Scenario	TTS	TDT	TTS	TDT	TTS	TDT
Balanced	2663.8	1899.8	2609.7 (-2.0%)	1508.6 (-20.6%)	2218.3 (-16.7%)	1030.3 (-45.8%)
Unbalanced						
String	154.0	115.7	124.0 (-19.5%)	34.7 (-70.0%)	146.5 (-4.9%)	50.1 (-56.7%)
Outside string	552.2	177.8	557.6 (1.0%)	184.4 (3.7%)	624.3 (13.1%)	199.6 (12.3%)
Network	706.2	293.5	681.7 (-3.5%)	219.1 (-25.3%)	770.8 (9.1%)	249.8 (-14.9%)

performance in the rest part of the subnetwork. As Fig. 12 shows, TTS and TDT for the other links of the subnetwork except the string keep almost the same values as that of the FT controller, and sometimes they are even worse. So from a global perspective, the MPC controllers are able to balance the traffic flow distribution and coordinate the control measures within the subnetwork to guarantee the overall performance. In particular, in the unbalanced scenario the MPC controllers coordinate the traffic signals within the subnetwork by sacrificing some performance of the links that are less crowded, so as to achieve a better overall performance.

The control performance and the relative improvements are listed in Table 1 for the three control strategies, the FT controller, the BLX-based MPC, and the S-based MPC, in both the balanced and unbalanced traffic scenarios. The MPC controllers can achieve better control performance, i.e. TTS and TDT, for the traffic network than the FT controller. In particular, the TDT can be reduced significantly by the MPC controllers. In the unbalanced scenario, both the TTS and TDT in the string are reduced obviously by the MPC controllers, while these performance measures are deteriorated to some extent in the area outside the string. In the unbalanced scenario, BLX-based MPC yields the best control performance, while in the balanced scenario, S-based MPC is the best control strategy. In theory, the difference of TTS and TDT between runs for the whole network should be equal, if the total demand is the same, the route choice is the same, and the traffic control signals are the same. For different control strategies, the MPC controllers are able to reduce the TDT on links of the network, and enable more vehicles run more smoothly through the links. Therefore, the differences between TTS and TDT (i.e. the total free flow travel time) under MPC controllers will be higher than that of FT controller.

In Fig. 13 and Fig. 14, the evolution of the TDT in some links of the traffic network in Fig. 5 are demonstrated for the balanced scenario and the unbalanced scenario respectively. These figures reveal how the TDT of vehicles in a link changes with time. In the balanced scenario, the TDT values of the FT controller are higher than that of the MPC controllers on most of the links. For Link (6,7), Link (7,8), and Link (9,10), it is obvious that the MPC controllers can control the TDT, so as to avoid the TDT to increase as fast as the TDT controlled by the FT controller. Moreover, for Link (8,9) and Link (12,7), the TDT can be stabilized by the MPC controllers some time after the network input traffic flows drop, but the FT controller fails to do that. Compared with the BLX-based MPC controller, the S-based MPC controller obtains even better control performance on the links in the string, at the cost of deteriorating the performance on some links outside the string, e.g. Link (12,7) and Link (5,10). In the unbalanced scenario, to keep the throughput efficiency of the string, where high traffic demands are supplied, the MPC controllers are able to reduce the TDT in the links of the string,

Table 2: CPU time (s) comparison for solving the optimization problems of the BLX-based and the S-based MPC controllers

Strategy	BLX-based MPC		S-based MPC	
Scenario	Avrg CPU (s)	Max CPU (s)	Avrg CPU (s)	Max CPU (s)
Balanced	485.8	862.3	33.9 (-93.0%)	67.3 (-92.2%)
Unbalanced	402.4	1024.2	56.5 (-85.9%)	104.5 (-89.8%)

particularly in Link (7,8) and Link (9,10), by means of slightly increasing the TDT in Link (12,7) and Link (5,10) (which are not in the string).

The evolution of the green time decisions of Phase 1 for Intersection 9, where Link (8,9) has the right of way, is given in Fig. 15 for both the balanced scenario and the unbalanced scenario. In the balanced scenario, the control decisions determined by the BLX-based MPC change more obviously than the control decisions of the S-based MPC. In general, the reason for this is that the simplification of the S model makes it lose some accuracy, and also makes it less sensitive to the control inputs. In the unbalanced scenario, the control decisions for both the BLX-based and S-based MPC controllers stay almost constant, due to the constant network traffic flow supply.

To solve the nonlinear optimization problem in (27) for the MPC controller on-line, the prediction model has to be simulated thousands of times. The speed for solving the optimization problem can be significantly improved by reducing the computation speed of the prediction model, just as Table 2 shows, where "Avrg CPU" represents the average CPU time used for solving the on-line optimization problems at each control time step, and "Max CPU" is the maximum CPU time. The average CPU time (and the maximum CPU time) used is reduced by around 85% - 93%, which indicates that the S-based controller is much more real-time efficient than the BLX-based controller. Meanwhile, the S-based controller is still able to keep a control performance that is similar to the one of the BLX-based controller. Therefore, the real-time computational feasibility of the MPC controller can be greatly improved by using the S model as the prediction model, by only sacrificing a limited amount of the performance.

6. Conclusions

In a structured urban road network control system, Model Predictive Control (MPC) can be applied in the subnetworks, which can provide coordination between the subnetwork controllers. As an advanced control methodology, MPC has many advantages, like robustness to disturbances, long-term sight, easy dealing with constraints, and so on. However, despite of all these advantages, it also inevitably gives rise to the problem of high on-line computational complexity. In this paper, an efficient but also effective MPC controller for urban road subnetworks is presented. The efficiency of the subnetwork controllers is also the basis of an efficient coordinating algorithm for the subnetworks.

To improve the applicability of the MPC controller in practice, the characteristics of the prediction models are considered. Two macroscopic models, the BLX model and the S model, are proposed and compared, where the S model is a reduced version of the BLX model, and resulting in a decreased computation speed. Simulation results

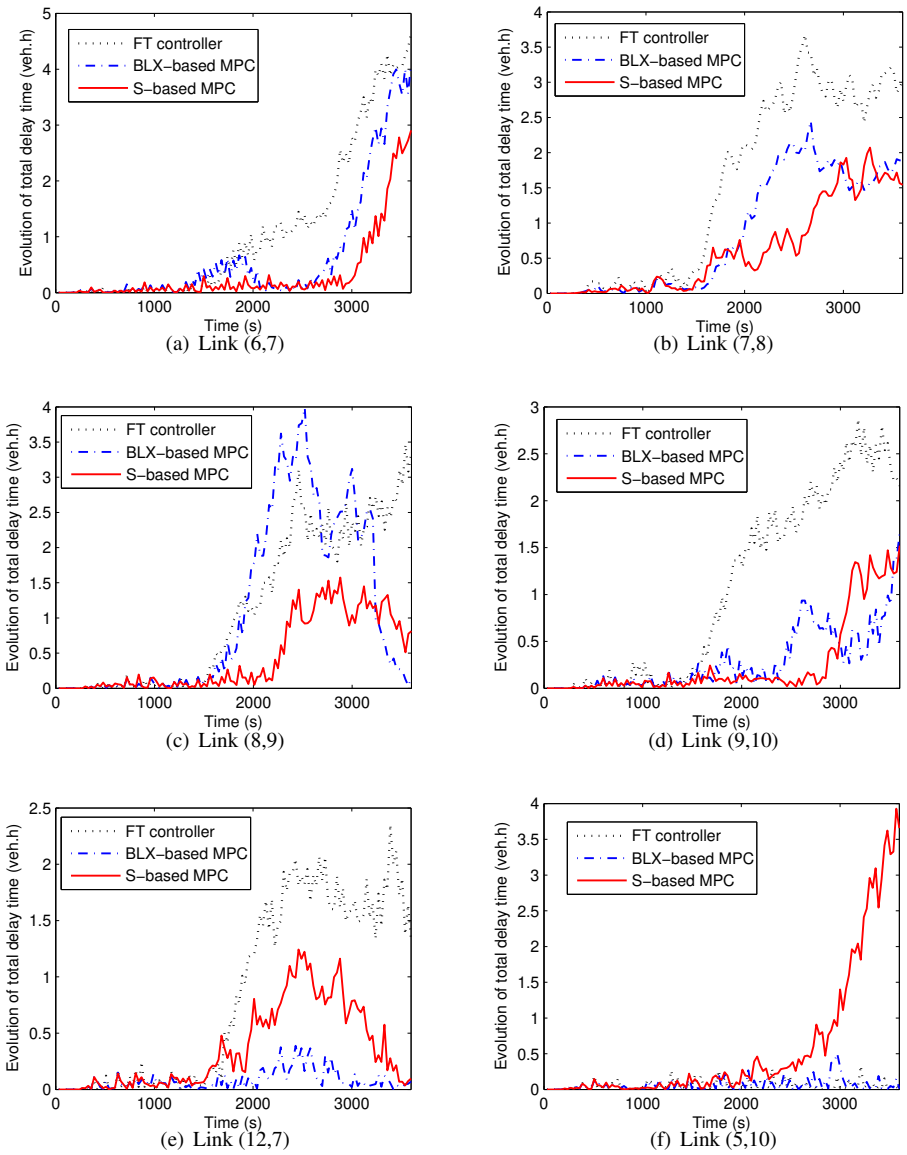
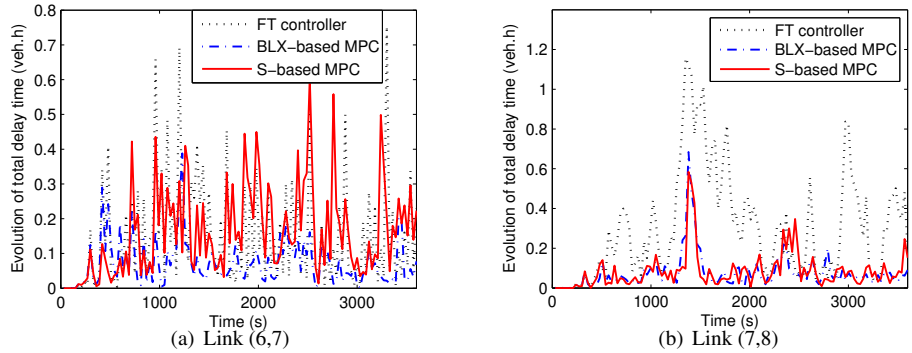
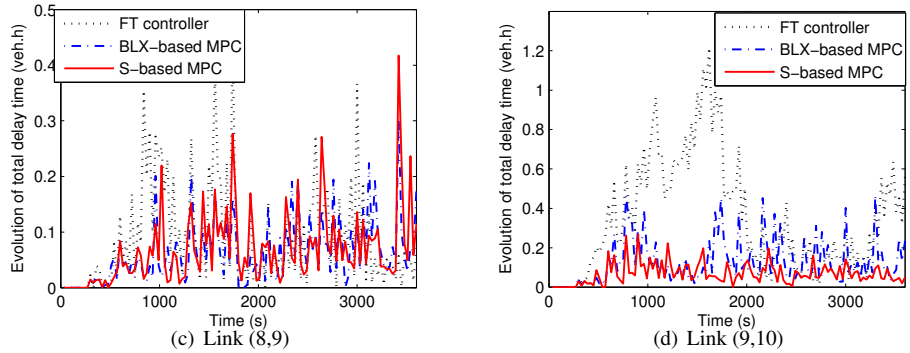


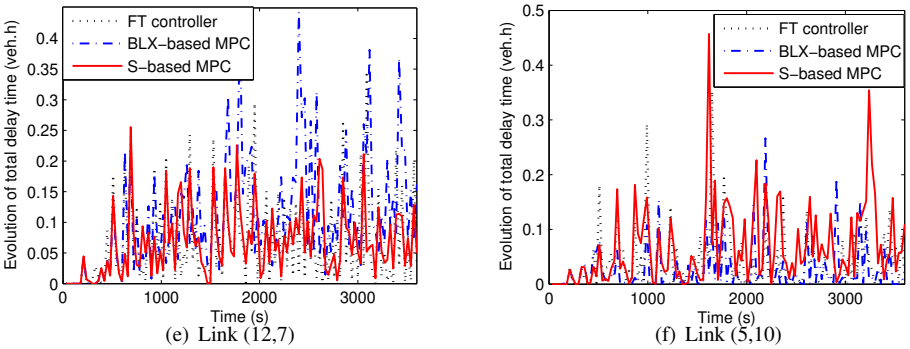
Figure 13: Evolution of the total delay time in some representative links under the balanced scenario



(a) Link (6,7) (b) Link (7,8)

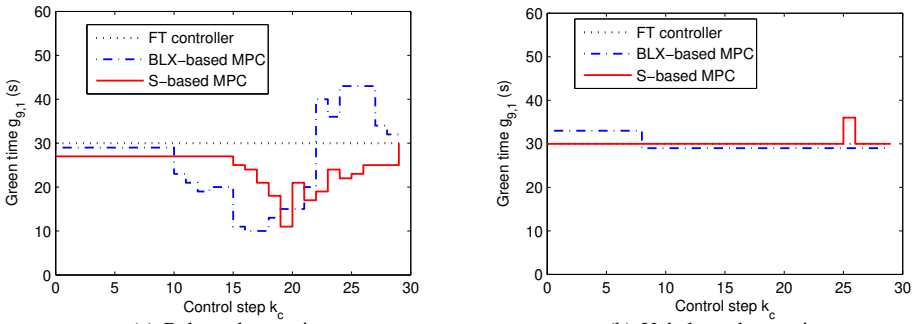


(c) Link (8,9) (d) Link (9,10)



(e) Link (12,7) (f) Link (5,10)

Figure 14: Evolution of the total delay time in some representative links under the unbalanced scenario



(a) Balanced scenario (b) Unbalanced scenario

Figure 15: Green time decision trajectories of Phase 1 for Intersection 9 at each control time step

show that both the S model and the BLX model are suitable as prediction model of MPC, and that the S model is much faster while still offering acceptable accuracy in the predictions. MPC controllers taking each of the models as prediction model respectively are constructed and investigated. The MPC controllers show great capability for coordinating the traffic measures and intersections within the subnetworks and achieve a good overall performance, both for the balanced and the unbalanced network traffic flow inputs. From a computational point of view, the S model-based MPC controller is much more efficient than the BLX model-based MPC controller, while only incurring a limited reduction of the control performance.

In the future, more case studies will be carried out to compare the presented MPC control strategy with other control strategies. The control structure for the network controller will be further considered, and the information transfer and coordination algorithms for the subnetworks will be investigated.

Appendix A. Discrete-time delay

In this paper, the urban traffic models are discrete-time models with discrete-time delays. The discrete-time delay represents the time taken by the vehicles entered a link reach (join) the end of the waiting queues in the link. According to Åström and Wittenmark (1996, Chap 2), this discrete-time delay can be derived by sampling a continuous-time system with time delays into a discrete-time system. This can be done as follows.

Let a linear continuous time-invariant system with time delay $\tau \in \mathbb{R}^+$ be described by[3]

$$\dot{\tilde{\mathbf{X}}}(t) = \mathbf{A}\tilde{\mathbf{X}}(t) + \mathbf{B}\tilde{\mathbf{U}}(t-\tau) \ . \tag{A.1}$$

Let us now sample this system using a sampling time interval T. Define

$$\delta = \text{floor}\left\{\frac{\tau}{T}\right\}, \qquad \gamma = \text{rem}\left\{\tau, T\right\}, \tag{A.2}$$

[3] In this appendix, $\tilde{\ }$ denotes a continuous variable.

where floor$\{x\}$ refers to the largest integer smaller than or equal to x, and rem$\{x,y\}$ is the remainder of the division of x by y. So δ is an integer, and the time delay τ can be expressed as

$$\tau = \delta \cdot T + \gamma \quad 0 \leq \gamma < T \ . \tag{A.3}$$

The zero-order-hold sampling of the system (A.1) can be calculated by integrating (A.1) over one sampling time interval T, as

$$\tilde{\mathbf{X}}(kT+T) = e^{\mathbf{A}T}\tilde{\mathbf{X}}(kT) + \int_{kT}^{kT+T} e^{\mathbf{A}(kT+T-s)}\mathbf{B}\tilde{\mathbf{U}}(s-\delta T-\gamma)\mathrm{d}s \ , \tag{A.4}$$

Assume $\tilde{\mathbf{U}}(t)$ is piecewise constant over each sampling time interval since $0 \leq \gamma < T$, the delayed input signal $\tilde{\mathbf{U}}(t-\tau)$ can then be split into two constant parts. Thus we have

$$\begin{aligned}
&\int_{kT}^{kT+T} e^{\mathbf{A}(kT+T-s)}\mathbf{B}\tilde{\mathbf{U}}(s-\delta T-\gamma)\mathrm{d}s \\
&\qquad = \int_{kT}^{kT+\gamma} e^{\mathbf{A}(kT+T-s)}\mathbf{B}\mathrm{d}s\,\tilde{\mathbf{U}}(kT-\delta T-T) \\
&\qquad\quad + \int_{kT+\gamma}^{kT+T} e^{\mathbf{A}(kT+T-s)}\mathbf{B}\mathrm{d}s\,\tilde{\mathbf{U}}(kT-\delta T) \\
&\qquad = \int_{0}^{T-\gamma} e^{\mathbf{A}s}\mathrm{d}s\,\mathbf{B}\,\tilde{\mathbf{U}}(kT-\delta T) \\
&\qquad\quad + e^{\mathbf{A}(T-\gamma)}\int_{0}^{\gamma} e^{\mathbf{A}s}\mathrm{d}s\,\mathbf{B}\,\tilde{\mathbf{U}}(kT-\delta T-T) \ .
\end{aligned} \tag{A.5}$$

Therefore, the sampled discrete-time system with the time delay becomes

$$\mathbf{X}(k+1) = \Phi\mathbf{X}(k) + \Gamma_0\mathbf{U}(k-\delta) + \Gamma_1\mathbf{U}(k-\delta-1) \ , \tag{A.6}$$

where

$$\Phi = e^{\mathbf{A}T} \tag{A.7}$$

$$\Gamma_0 = \int_{0}^{T-\gamma} e^{\mathbf{A}s}\mathrm{d}s\,\mathbf{B} \tag{A.8}$$

$$\Gamma_1 = e^{\mathbf{A}(T-\gamma)}\int_{0}^{\gamma} e^{\mathbf{A}s}\mathrm{d}s\,\mathbf{B} \ . \tag{A.9}$$

Normally , the vehicles that enter into a link will run with free-flow speed for a certain time period, and then finally join the tail of the queues. This time period is a time delay that is needed before the vehicles join the waiting queues at the stop-line of the link. Therefore, the queue length in a link can be updated by the number of vehicles leaving the link and the number of delayed vehicles entering the link. The differential equation describing the evolution of the queue length can be therefore written as

$$\dot{\tilde{q}}_{u,d,o}(t) = \tilde{\beta}_{u,d,o}(t)\tilde{\alpha}^{\text{enter}}_{u,d}(t-\tau) - \tilde{\alpha}^{\text{leave}}_{u,d,o}(t)\,, \tag{A.10}$$

i.e. the rate of change of the queue length ($\dot{\tilde{q}}_{u,d,o}(t)$) is equal to the difference between the input flow rate (delayed by τ and then divided by multiplying the current turning

rate) and the output flow rate. In (A.10), the traffic flow turning rate ($\tilde{\beta}_{u,d,o}(t)$) and the traffic flow rate entering or leaving the queue ($\tilde{\alpha}^{\text{enter}}_{u,d}(t)$ and $\tilde{\alpha}^{\text{leave}}_{u,d,o}(t)$) are all piecewise constant during the sampling time intervals. The traffic flow turning rate will be influenced by the traffic flows and traffic signals the drivers have experienced upstream, the traffic signals in front, and the origin-destination of the drivers. Then, according to the addition principle of linear equations, (A.10) can be divided into two equations, as

$$\dot{\tilde{q}}^1_{u,d,o}(t) = -\tilde{\alpha}^{\text{leave}}_{u,d,o}(t) \tag{A.11}$$

$$\dot{\tilde{q}}^2_{u,d,o}(t) = \tilde{\beta}_{u,d,o}(t)\tilde{\alpha}^{\text{enter}}_{u,d}(t-\tau), \tag{A.12}$$

such that

$$\tilde{q}_{u,d,o}(t) = \tilde{q}^1_{u,d,o}(t) + \tilde{q}^2_{u,d,o}(t)\,. \tag{A.13}$$

To sample differential equation (A.11) without a time delay into a discrete-time equation, we define $A = 0$ and $B = -1$, then according to (A.6), (A.8), and (A.9), we have

$$q^1_{u,d,o}(k+1) = \Phi q^1_{u,d,o}(k) + \Gamma\alpha^{\text{leave}}_{u,d,o}(k) \tag{A.14}$$

where

$$\begin{aligned}\Phi &= e^{AT} = 1\\ \Gamma &= \int_0^T e^{\mathbf{A}s}\mathrm{d}s\,\mathbf{B} = -T.\end{aligned} \tag{A.15}$$

Similarly, we can sample differential equation (A.12) with a time delay τ into a discrete-time equation. Since the time delay τ will vary slowly with time t, then according to (A.2) and (A.3) we can approximately have

$$\delta(k) = \text{floor}\left\{\frac{\tau(k)}{T}\right\}, \qquad \gamma(k) = \text{rem}\left\{\tau(k), T\right\}, \tag{A.16}$$

and

$$\tau(k) = \delta(k)\cdot T + \gamma(k) \quad 0 \le \gamma(k) < T \ . \tag{A.17}$$

Next, we define $A_\tau = 0$ and $B_\tau = 1$, and then according to (A.6), (A.8), and (A.9), (A.12) results in

$$\begin{aligned}q^2_{u,d,o}(k+1) =& \Phi_\tau q^2_{u,d,o}(k) + \beta_{u,d,o}(k)\big(\Gamma_0\alpha^{\text{enter}}_{u,d}(k-\delta(k))\\ &+\Gamma_1\alpha^{\text{enter}}_{u,d}(k-\delta(k)-1)\big),\end{aligned} \tag{A.18}$$

where

$$\begin{aligned}\Phi_\tau &= e^{AT} = 1\\ \Gamma_0 &= \int_0^{T-\gamma(k)} e^{\mathbf{A}s}\mathrm{d}s\,\mathbf{B}_\tau = T-\gamma(k)\\ \Gamma_1 &= e^{\mathbf{A}(T-\gamma(k))}\int_0^{\gamma(k)} e^{\mathbf{A}s}\mathrm{d}s\,\mathbf{B}_\tau = \gamma(k)\end{aligned} \tag{A.19}$$

Therefore, by adding (A.14) and (A.18) together, we derive

$$\begin{aligned} q_{u,d,o}(k+1) = q_{u,d,o}(k) - T\alpha^{\text{leave}}_{u,d,o}(k) \\ + \beta_{u,d,o}(k)\big((T-\gamma(k))\alpha^{\text{enter}}_{u,d}(k-\delta(k)) \\ + \gamma(k)\alpha^{\text{enter}}_{u,d}(k-\delta(k)-1)\big), \end{aligned} \tag{A.20}$$

and the arriving average traffic flow at the tail of the queues is

$$\alpha^{\text{arriv}}_{u,d}(k) = \frac{T-\gamma(k)}{T}\alpha^{\text{enter}}_{u,d}(k-\delta(k)) + \frac{\gamma(k)}{T}\alpha^{\text{enter}}_{u,d}(k-\delta(k)-1). \tag{A.21}$$

Based on the number of vehicles waiting in queues ($q_{u,d,o}(k)$) and the average distance headway of two successive vehicles in queues (h), we can approximately calculate the back of the queues by

$$d_{u,d,o}(k) = q_{u,d,o}(k)h \ , \tag{A.22}$$

where $d_{u,d,o}(k)$ represents the distance from the back of the queue to the stop line.

Acknowledgments

This research is supported by a Chinese Scholarship Council (CSC) grant, the National Science Foundation of China (Grant No. 60674041, 60934007, 61104160), the European COST Actions TU0702 and TU1102, the 7th framework European STREP project "Hierarchical and distributed model predictive control (HD-MPC)" (contract number INFSO-ICT-223854), the European Union Seventh Framework Programme [FP7/2007-2013] under grant agreement no. 257462 HYCON2 Network of Excellence, the Delft Research Center Next Generation Infrastructures, and the Transport Research Centre Delft.

References

Åström, K., Wittenmark, B., 1996. Computer-Controlled Systems: Theory and Design. Prentice Hall New York.

Aboudolas, K., Papageorgiou, M., Kosmatopoulos, E., 2009. Store-and-forward based methods for the signal control problem in large-scale congested urban road networks. Transportation Research Part C: Emerging Technologies 17 (2), 163–174.

Aboudolas, K., Papageorgiou, M., Kouvelas, A., Kosmatopoulos, E., 2010. A rolling-horizon quadratic-programming approach to the signal control problem in large-scale congested urban road networks. Transportation Research Part C: Emerging Technologies 18 (5), 680–694.

Boillot, F., Blosseville, J., Lesort, J., Motyka, V., Papageorgiou, M., Sellam, S., 1992. Optimal signal control of urban traffic networks. In: Proc. of Conference on Road Traffic Monitoring and Control. pp. 75–79.

Boillot, F., Midenet, S., Pierrelee, J. C., 2006. The real-time urban traffic control system CRONOS: Algorithm and experiments. Transportation Research Part C: Emerging Technologies 14 (1), 18–38.

Camacho, E., Bordons, C., 1995. Model Predictive Control in the Process Industry. Springer-Verlag, Berlin, Germany.

Di Febbraro, A., Giglio, D., Sacco, N., Dec. 2004. Urban traffic control structure based on hybrid petri nets. IEEE Transactions on Intelligent Transportation Systems 5 (4), 224–237.

Dotoli, M., Fanti, M. P., Meloni, C., 2006. A signal timing plan formulation for urban traffic control. Control Engineering Practice 14 (11), 1297–1311.

Farges, J., Henry, J., Tufal, J., 1983. The PRODYN real-time traffic algorithm. In: Proc. 4th IFAC Symposium of Transportation Systems. Baden Baden, Germany, pp. 307–312.

FHWA, 2001. Traffic Software Integrated System Version 5.1 User's Guide.

Garcia, C., Prett, D., Morari, M., 1989. Model predictive control: Theory and practice–A survey. Automatica 25 (3), 335–348.

Gartner, N., 1983. Simulation study of OPAC: A demand-responsive strategy for traffic signal control. Transportation and Traffic Theory, 233–250.

Gartner, N. H., Pooran, F. J., Andrews, C. M., 2001. Implementation of the OPAC adaptive control strategy in a traffic signal network. In: Proc. of the 2001 IEEE International Intelligent Transportation Systems Conference. Oakland (CA), USA, pp. 195–200.

Hale, D., 2005. Traffic Network Study Tool, TRANSYT-7F, United States Version. McTrans Center in the University of Florida, Gainesville, Floria.

Hegyi, A., De Schutter, B., Hellendoorn, H., 2005. Model predictive control for optimal coordination of ramp metering and variable speed limits. Transportation Research Part C: Emerging Technologies 13 (3), 185–209.

Kashani, H., Saridis, G., 1983. Intelligent control for urban traffic systems. Automatica 19 (2), 191–197.

Kotsialos, A., Papageorgiou, M., 2004. Efficiency and equity properties of freeway network-wide ramp metering with AMOC. Transportation Research Part C: Emerging Technologies 12 (6), 401–420.

Lin, S., De Schutter, B., Xi, Y., Hellendoorn, H., Sep. 2011. Fast model predictive control for urban road networks via MILP. IEEE Transactions on Intelligent Transportation Systems 12 (3), 846–856.

Lin, S., De Schutter, B., Xi, Y., Hellendoorn, J., 2009. A simplified macroscopic urban traffic network model for model-based predictive control. In: Proc. 12th IFAC Symposium Control Transportation Systems. Redondo Beach (CA), USA, pp. 286–291.

Lin, S., Xi, Y., 2008. An efficient model for urban traffic network control. In: Proc. of the 17th World Congress The International Federation of Automatic Control. Seoul, Korea, pp. 14066–14071.

Little, J., Kelson, M., Gartner, N., 1981. Maxband: A program for setting signals on arteries and triangular networks. Transportation Research Record (795).

Mirchandani, P., Head, L., 2001. A real-time traffic signal control system: Architecture, algorithm, and analysis. Transportation Research Part C: Emerging Technologies 9 (6), 415–432.

Papageorgiou, M., 1983. Applications of Automatic Control Concepts to Traffic Flow Modeling and Control. Springer-Verlag New York, Inc. Secaucus, NJ, USA.

Papageorgiou, M., Diakaki, C., Dinopoulou, V., Kotsialos, A., Wang, Y., 2003. Review of road traffic control strategies. Proceedings of the IEEE 91 (12), 2043–2067.

Pardalos, P., Rosen, J., 1987. Constrained Global Optimization: Algorithms and Applications. Springer-Verlag, Berlin.

Rawlings, J., Mayne, D., 2009. Model Predictive Control: Theory and Design. Nob Hill Publishing, Madison, Wisconsin.

Richalet, J., Rault, A., Testud, J., Papon, J., 1978. Model predictive heuristic control: Applications to industrial processes. Automatica 14 (5), 413–428.

Robertson, D., Bretherton, R., 1991. Optimizing networks of traffic signals in real time - The SCOOT method. IEEE Transactions on Vehicular Technology 40 (1), 11–15.

Sen, S., Head, K., 1997. Controlled optimization of phases at an intersection. Transportation Science 31 (1), 5–17.

van den Berg, M., Hegyi, A., De Schutter, B., Hellendoorn, J., Sep. 2007. Integrated traffic control for mixed urban and freeway networks: A model predictive control approach. European Journal of Transport and Infrastructure Research 7 (3), 223–250.

van Katwijk, R., 2008. Multi-agent look-ahead traffic-adaptive control. Ph.D. thesis, Delft University of Technology.

Webster, F., Cobbe, B., 1966. Traffic signals. Road Research, 56, Her Majestys Stationery Office, London, England.