Technical report 09-040

Study on fast model predictive controllers for large urban traffic networks*

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

* This report can also be downloaded via https://pub.bartdeschutter.org/abs/09_040.html

Study on Fast Model Predictive Controllers for Large Urban Traffic Networks

Shu Lin, Bart De Schutter, Yugeng Xi and Hans Hellendoorn

Abstract— **Traffic control is both an efficient and effective way to alleviate the traffic congestion in urban areas. Model Predictive Control (MPC) has advantages in controlling and coordinating urban traffic networks. But, the real-time computational complexity of MPC increases exponentially, when the network scale and the predictive time horizon grow. To improve the real-time feasibility of MPC, a simplified macroscopic urban traffic model is developed. Two MPC controllers are built based on the simplified model and a more detailed model. Simulation results of the two controllers show that the on-line optimization time is reduced dramatically by applying the simplified model, only losing a limited amount of control effectiveness. Additional techniques, like applying a control time horizon and an aggregation scheme, are implemented to reduce the computational complexity further. Simulation results show positive effects of these techniques.**

I. Introduction

Traffic jams occur frequently in urban areas, when people need to use the common infrastructures with limited capacity at the same time, especially in rush hours. Traffic congestion can give rise to traffic delays, economic losses, traffic pollution, and even non-safety. Constructing more new roads is not only consuming but also impractical for some cities. In this situation, traffic control strategies are the most efficient and also effective method to solve the congestion problem.

There are already many control strategies [1] for urban traffic. Among them, model-based optimization methods have the same basic frameworks with three typical features: prediction model, on-line optimization, and rolling time horizon. Due to these features, model-based optimization control methods can deal with the uncertainty of real traffic, and avoid myopic control schemes. Model Predictive Control (MPC) is a control strategy within this category.

In the 1980s and 1990s, a number of model-based optimization control strategies emerged: OPAC, PRODYN, CRONOS, and RHODES. The prediction models for these strategies are almost the same. They predict the future traffic demands at the intersections through historical data measured from the upstream detectors and the detectors in the upstream links. These strategies showed advantages compared with the traffic-responsive strategies which are without predictions [2]. However, this kind of prediction models are limited in the length of the time horizon over which they can predict. The longest prediction horizon corresponds to the time taken by the vehicles running from the upstream detector to the stop-line of the intersection. Therefore, the control strategies cannot look ahead far enough due to this limitation. In recent years, some macroscopic urban traffic models have been developed. These models can describe the traffic dynamic mechanics of the whole urban traffic network, and overcome the drawbacks of the previous models. Model-based optimization control strategies [3], [4] (including MPC) were developed based on these prediction models, and obtained good control effects. However, the challenge for these strategies is the trade-off between the accuracy of the models and the on-line computational complexity. In fact, all the model-based optimization strategies inevitably encounter the same problem, i.e. the real-time computational feasibility. Almost all of them are not real-time feasible for controlling traffic networks with more than 5-10 intersections. To avoid this problem, other types of strategies are developed instead. One is to divide the network into intersections and control them under a distributed structure [5]–[8]. The other is to solve the optimization problem off-line, such as with a feedback regulator and optimal control [9]. These methods succeeded in improving the real-time feasibility. But, to keep the advantages of centralized MPC controllers, other approaches need to be considered.

The main factors influencing the real-time feasibility are the complexity of the model, the optimization algorithm, the prediction horizon, and the scale of the network. To improve the real-time feasibility, the following two methods can be considered. First, to simplify the urban traffic model. Even for macroscopic models, there exist different levels of modeling details. A better trade-off between model accuracy and computational complexity has to be found, i.e. the model should be guaranteed as simple and fast as possible provided that the model is accurate enough for controller. Second, to adopt some tricks to speed up the optimization algorithms.

We developed a simplified model to reduce the computational complexity of the previous macroscopic urban traffic network model [4], [10]. Two MPC controllers are built based on two different models. The real-time feasibility of the MPC controllers is investigated for different prediction horizons and traffic network scales. We demonstrate that the controller based on the simplified model is much faster than the other one, with a limited amount of loss of control effect. Moreover, different control horizons and aggregation techniques are adopted to reduce the computational burden of the optimization algorithm one step further.

S. Lin and Y. Xi are with Department of Automation, Shanghai Jiao Tong University, Shanghai, P. R. China; S. Lin is a visiting researcher at the Delft Center for Systems and Control, Delft University of Technology. lisashulin@gmail.com, ygxi@sjtu.edu.cn

B. De Schutter and J. Hellendoorn are with the Delft Center for Systems and Control, Delft University of Technology, The Netherlands; B. De Schutter is also with the Marine & Transport Technology department of Delft University of Technology. b.deschutter@dcsc.tudelft.nl, j.hellendoorn@tudelft.nl

II. Two Macroscopic Urban Traffic Models

In this section we present the original model of [4] and [10] (indicated as the BLX model) as well as a new simplified model (called the S model). But first we introduce some common notation for both models.

Define J the set of nodes (intersections), and L the set of links (roads) in the urban traffic network. Link (u,d) is marked by its upstream node u ($u \in J$) and downstream node d ($d \in J$). The sets of input and output links for link (u,d) are $I_{u,d} \subset L$ and $O_{u,d} \subset L$ (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 links of link (u,d),
$O_{u,d}$	: set of output links of link (u,d),
k	: simulation step counter for the urban traffic model,
$n_{u,d}(k)$	: 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_m} is the queue length of the sub-stream turning to link o_m,
$m^{\mathrm{l}}_{u,d,o_m}(k)$	: number of cars leaving link (u,d) and turning to o_m,
$m^{\mathrm{a}}_{u,d}(k)$	: number of cars arriving *at the (end of the) queue* in link (u,d) at step k, $m^{\mathrm{a}}_{u,d,o_m}(k)$ is the number of arriving cars in the sub-stream going towards o_m,
$S_{u,d}(k)$	: available storage space of link (u,d) at step k expressed in number of vehicles,
$\alpha^{\mathrm{l}}_{u,d}(k)$	: average flow rate leaving link (u,d) at step k, $\alpha^{\mathrm{l}}_{u,d,o_m}(k)$ is the leaving average flow rate of the sub-stream going towards o_m,
$\alpha^{\mathrm{a}}_{u,d}(k)$	: average flow rate arriving at the end of the queue in link (u,d) at step k, $\alpha^{\mathrm{a}}_{u,d,o_m}(k)$ is the arriving average flow rate of the sub-stream going towards o_m,
$\alpha^{\mathrm{e}}_{u,d}(k)$	: average flow rate entering link (u,d) at step k,
$\beta_{u,d,o_m}(k)$	: relative fraction of the traffic in link (u,d) turning to o_m at step k,
$\mu_{u,d}$	: saturated flow rate leaving link (u,d),
$g_{u,d,o_m}(k)$	: green time length during step k for the traffic stream in link (u,d) going towards o_m,
$b_{u,d,o_m}(k)$	: boolean value indicating whether the traffic signal at intersection d for the traffic stream in link (u,d) turning to o_m is green (1) or red (0) at step k,
$v^{\mathrm{free}}_{u,d}$	: free-flow vehicle speed in link (u,d),
$C_{u,d}$	: capacity of link (u,d) expressed in number of vehicles,
$N^{\mathrm{lane}}_{u,d}$	: number of lanes in link (u,d),
$\Delta c_{u,d}$	: offset between node u and node d,
l_{veh}	: average vehicle length.

A. BLX model

In the BLX model 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 move into the correct lane, so that they do not block the traffic flows going to other destinations. For each lane (or destination), 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 $o_m \in O_{u,d}$ the number of cars leaving link (u,d) for destination o_m in the period $[kT_{\mathrm{s}}, (k+1)T_{\mathrm{s}})$ is given by

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

The traffic arriving at the end of the queue in link (u,d) is given by the traffic entering the link via the upstream intersection delayed by the time $\tau(k) \cdot T_{\mathrm{s}} + \gamma(k)$ needed to drive from the upstream intersection to the end of the queue in the link; to this extent $m^{\mathrm{a}}_{u,d}$ is updated as follows:

$$\begin{aligned} m^{\mathrm{a}}_{u,d}(k) \quad &= \quad (1-\gamma(k)) \cdot \sum_{i_m \in I_{u,d}} m^{\mathrm{l}}_{i_m,u,d}(k-\tau(k)) + \\ &\qquad \gamma(k) \cdot \sum_{i_m \in I_{u,d}} m^{\mathrm{l}}_{i_m,u,d}(k-\tau(k)-1) \,, \end{aligned} \tag{2}$$

where

$$\begin{aligned} \tau(k) &= \mathrm{floor}\left\{ \frac{\big(C_{u,d} - q_{u,d}(k)\big) \cdot l_{\mathrm{veh}}}{N^{\mathrm{lane}}_{u,d} \cdot v^{\mathrm{free}}_{u,d} \cdot T_{\mathrm{s}}} \right\}, \\ \gamma(k) &= \mathrm{rem}\left\{ \frac{\big(C_{u,d} - q_{u,d}(k)\big) \cdot l_{\mathrm{veh}}}{N^{\mathrm{lane}}_{u,d} \cdot v^{\mathrm{free}}_{u,d} \cdot T_{\mathrm{s}}} \right\}, \end{aligned} \tag{3}$$

with floor(x) referring to the largest integer smaller than or equal to x, and rem(x) is the remainder. The fraction of the arriving traffic in link (u,d) turning to $o_m \in O_{u,d}$ is

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

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

$$q_{u,d,o_m}(k+1) = q_{u,d,o_m}(k) + m^{\mathrm{a}}_{u,d,o_m}(k) - m^{\mathrm{l}}_{u,d,o_m}(k) \tag{5}$$

for each $o_m \in O_{u,d}$, and

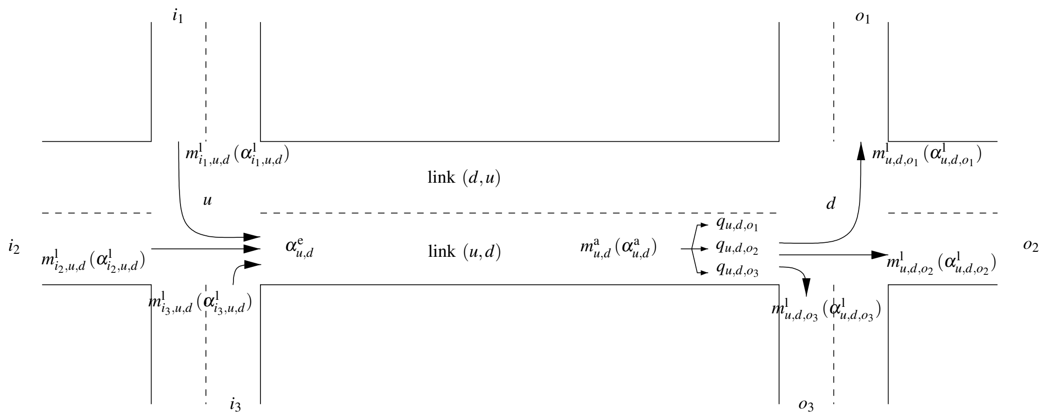
$$q_{u,d}(k) = \sum_{o_m \in O_{u,d}} q_{u,d,o_m}(k) \ . \tag{6}$$

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

$$S_{u,d}(k+1) = S_{u,d}(k) - \sum_{i_m \in I_{u,d}} m^{\mathrm{l}}_{i_m,u,d}(k) + \sum_{o_m \in O_{u,d}} m^{\mathrm{l}}_{u,d,o_m}(k) \ , \tag{7}$$

then the number of vehicles in link (u,d) in the period $[kT_{\mathrm{s}}, (k+1)T_{\mathrm{s}})$ is easily derived as

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



i_1
o_1
$m^{\mathrm{l}}_{i_1,u,d}\,(\alpha^{\mathrm{l}}_{i_1,u,d})$
link (d,u)
$m^{\mathrm{l}}_{u,d,o_1}\,(\alpha^{\mathrm{l}}_{u,d,o_1})$
u
d
i_2
$m^{\mathrm{l}}_{i_2,u,d}\,(\alpha^{\mathrm{l}}_{i_2,u,d})$
$\alpha^{\mathrm{e}}_{u,d}$
link (u,d)
$m^{\mathrm{a}}_{u,d}\,(\alpha^{\mathrm{a}}_{u,d})$
q_{u,d,o_1}
q_{u,d,o_2}
q_{u,d,o_3}
o_2
$m^{\mathrm{l}}_{u,d,o_2}\,(\alpha^{\mathrm{l}}_{u,d,o_2})$
$m^{\mathrm{l}}_{i_3,u,d}\,(\alpha^{\mathrm{l}}_{i_3,u,d})$
$m^{\mathrm{l}}_{u,d,o_3}\,(\alpha^{\mathrm{l}}_{u,d,o_3})$
i_3
o_3

Fig. 1. A link connecting two traffic-signal-controlled intersections

B. Simplified Model (S Model)

In the simplified model, every intersection takes the cycle time as its 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 number 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

$$n_{u,d}(k_d+1) = n_{u,d}(k_d) + \Big(\alpha^{\mathrm{e}}_{u,d}(k_d) - \alpha^{\mathrm{l}}_{u,d}(k_d)\Big)\cdot c_d \ . \quad (9)$$

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

$$\alpha^{\mathrm{l}}_{u,d}(k_d) = \sum_{o_m \in O_{u,d}} \alpha^{\mathrm{l}}_{u,d,o_m}(k_d) \ . \quad (10)$$

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:

$$\begin{aligned}\alpha^{\mathrm{l}}_{u,d,o_m}(k_d) = \min\Big(&\beta_{u,d,o_m}(k_d)\cdot\mu_{u,d}\cdot g_{u,d,o_m}(k_d)/c_d, \quad (11)\\ &q_{u,d,o_m}(k_d)/c_d + \alpha^{\mathrm{a}}_{u,d,o_m}(k_d),\\ &\beta_{u,d,o_m}(k_d)\left(C_{o_m} - n_{o_m}(k_d)\right)/c_d\Big) \ .\end{aligned}$$

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

$$\begin{aligned}q_{u,d,o_m}(k_d+1) \quad = \quad & q_{u,d,o_m}(k_d) + \\ & \Big(\alpha^{\mathrm{a}}_{u,d,o_m}(k_d) - \alpha^{\mathrm{l}}_{u,d,o_m}(k_d)\Big)\cdot c_d . (12)\end{aligned}$$

Then, the number of waiting vehicles in link (u,d) is

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

The flow rate entering link (u,d) will arrive at the end of the queues after a time delay $\tau(k_d)\cdot c_d + \gamma(k_d)$, i.e.,

$$\begin{aligned}\alpha^{\mathrm{a}}_{u,d}(k_d) \quad = \quad & (1-\gamma(k_d))\cdot\alpha^{\mathrm{e}}_{u,d}(k_d - \tau(k_d)) + \\ & \gamma(k_d)\cdot\alpha^{\mathrm{e}}_{u,d}(k_d - \tau(k_d) - 1), \quad (14)\end{aligned}$$

with

$$\tau(k_d) = \mathrm{floor}\left\{\frac{\left(C_{u,d} - q_{u,d}(k_d)\right)\cdot l_{\mathrm{veh}}}{N^{\mathrm{lane}}_{u,d}\cdot v^{\mathrm{free}}_{u,d}\cdot c_d}\right\},$$

$$\gamma(k_d) = \mathrm{rem}\left\{\frac{\left(C_{u,d} - q_{u,d}(k_d)\right)\cdot l_{\mathrm{veh}}}{N^{\mathrm{lane}}_{u,d}\cdot v^{\mathrm{free}}_{u,d}\cdot c_d}\right\}. \quad (15)$$

Before reaching the tail of the waiting queues in link (u,d), the flow rate of arriving vehicles need be divided by multiplying it with the turning rates:

$$\alpha^{\mathrm{a}}_{u,d,o_m}(k_d) = \beta_{u,d,o_m}(k_d)\cdot\alpha^{\mathrm{a}}_{u,d}(k_d). \quad (16)$$

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

$$\alpha^{\mathrm{e}}_{u,d}(k_d) = \sum_{i_m \in I_{u,d}} \alpha^{\mathrm{l}}_{i_m,u,d}(k_d). \quad (17)$$

So 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, so the simulation time steps are not the same. Some operations have to be carried out to synchronize the leaving and entering flow rates.

In order to control the urban traffic network, a common control time interval needs to be defined for the network model, so that intersections can communicate with each other and be synchronous. So T_{c} is defined as the least common multiple of all the intersection cycle times in the traffic network, i.e.

$$T_{\mathrm{c}} = N_u\cdot c_u = N_d\cdot c_d. \quad (18)$$

Now we show how the flow rates expressed in the timing of intersection u can be recast into the timing of intersection d. First, we transform the discrete time leaving flow rates from the upstream links into continues time using the zero-order hold strategy, as

$$\alpha^{\mathrm{l,C}}_{i_m,u,d}(t) = \alpha^{\mathrm{l}}_{i_m,u,d}(k_u), \quad k_u\cdot c_u \le t < (k_u+1)\cdot c_u, \quad (19)$$

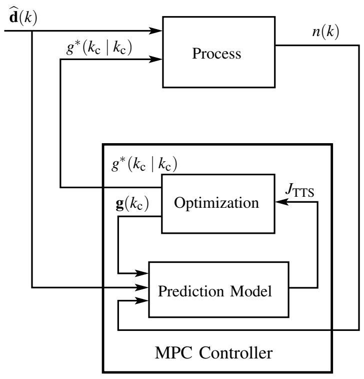


Fig. 2. The framework of the MPC controller

and then sample them again to obtain the average flow rates in time step k_d so as to be able used by the downstream link

$$\alpha^{1}_{i_m,u,d}(k_d) = \frac{\int_{k_d \cdot c_d + \Delta c_{u,d}}^{(k_d+1)\cdot c_d + \Delta c_{u,d}} \alpha^{1,\mathrm{C}}_{i_m,u,d}(t)}{c_d}\,\mathrm{d}t \ , \tag{20}$$

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.

III. MPC Controller For Urban Traffic Network

Model Predictive Control [11] is a methodology that implements and repeatedly applies Optimal Control in a rolling horizon way. In each control step, 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 estimate of the disturbances.

Similar to Optimal Control, MPC can predict and find the optimal solution for the future. Different with Optimal Control, MPC has the ability to deal with the uncertainty of the process, which can be caused by the unpredictable disturbances, the (slow) variation over time of the parameters, and model mismatches in the prediction model. But, in practice, to maintain the on-line computational feasibility, the optimization will be calculated over a finite time span instead of up to infinity. MPC can also easily deal with multi-input and multi-output problems with constraints. Another advantage of MPC is that one can easily select and replace the prediction model based on the control requirements.

Fig. 2 shows the structure of the MPC controller. The control process can be described by the following steps:

1) **Prediction model**. The model can be selected as prediction model for MPC controller, if it can predict the future traffic states used for evaluating the objective function based on the information of current states, predicted disturbances, and future control inputs. Therefore, both the traffic models presented in Section II above can be used as the prediction models for the MPC controller. They can be generally described as $n(k+1) = f\big(n(k), g(k_c), d(k)\big)$, where $n(k)$ is the traffic state (the number of vehicles in a link at time step k, see (8) and (9)) needed in evaluating the objective function; $d(k)$ is the predicted disturbance (the traffic demand), which is the input traffic flow rate for the network in the future; $g(k_c)$ is the future control input, e.g. the green time splits.
2) **Optimization problem**. Suppose the control interval T_c and simulation interval T_s satisfy $T_c = MT_s$ and the prediction horizon is N_p, then the future traffic states are predicted at simulation time step k as

$$\hat{\mathbf{n}}(k) = [\hat{n}^{\mathrm{T}}(k+1 \mid k)\ \hat{n}^{\mathrm{T}}(k+2 \mid k) \cdots \hat{n}^{\mathrm{T}}(k+MN_p \mid k)]^{\mathrm{T}},$$

based on the predicted traffic demands at simulation time step k

$$\hat{\mathbf{d}}(k) = [\hat{d}^{\mathrm{T}}(k \mid k)\ \hat{d}^{\mathrm{T}}(k+1 \mid k) \cdots \hat{d}^{\mathrm{T}}(k+MN_p-1 \mid k)]^{\mathrm{T}},$$

and the future traffic control inputs at control step k_c

$$\begin{aligned}\mathbf{g}(k_c) = {} & [g^{\mathrm{T}}(k_c \mid k_c)\ g^{\mathrm{T}}(k_c+1 \mid k_c) \cdots \\ & g^{\mathrm{T}}(k_c+N_p-1 \mid k_c)]^{\mathrm{T}} \ .\end{aligned}$$

The optimization problem with Total Time Spent (TTS) as objective function can be expressed as

$$\begin{aligned} \min_{\mathbf{g}(k_c)} J_{\mathrm{TTS}} &= J\big(\hat{\mathbf{n}}(k), \mathbf{g}(k_c)\big) \\ &= \sum_{k=Mk_c+1}^{M(k_c+N_p)} \sum_{l \in L} T_s \cdot \hat{n}_l(k) \\ s.t. \quad & \text{model constraints} \\ & \Phi(\mathbf{g}(k_c)) = 0 \quad \text{(cycle time constraints)} \\ & g_{\min} \le \mathbf{g}(k_c) \le g_{\max} \end{aligned} \tag{21}$$

where $\hat{n}_l(k)$ is the estimated number of vehicles on link l at time step k. However, because the optimization problem is computed on-line, the time taken to solve it is always a big issue for MPC. Two methods can be chosen to reduce the on-line computational complexity: (1) define control horizon N_c with $N_c < N_p$; (2) adopt aggregation techniques. Both of the two methods decrease the computational complexity by reducing the number of control variables optimized.

In method (1), the vector of the optimized variables is

$$\begin{aligned}\mathbf{g}(k_c) = {} & [g^{\mathrm{T}}(k_c \mid k_c)\ g^{\mathrm{T}}(k_c+1 \mid k_c) \cdots \\ & g^{\mathrm{T}}(k_c+N_c-1 \mid k_c)]^{\mathrm{T}},\end{aligned}$$

and we set

$$g(k_c+i \mid k_c) = g(k_c+N_c-1 \mid k_c) \quad i = N_c, \cdots, N_p-1.$$

In the second method, the vector of the optimized control variables $\mathbf{g}(k_c) \in \mathbb{R}^N$ is substituted by a vector $\mathbf{v}(k_c) \in \mathbb{R}^s$ with lower dimension by introducing an aggregation (or blocking) matrix $\mathbf{H}$: $\mathbf{g}(k_c) = \mathbf{H} \cdot \mathbf{v}(k_c)$, where $\mathbf{H} \in \mathbb{R}^{N \times s}$ with $s < N$. The aggregation matrix is the key factor of aggregation techniques. Different aggregation matrices allow different aggregation schemes. One of the most typical aggregation schemes

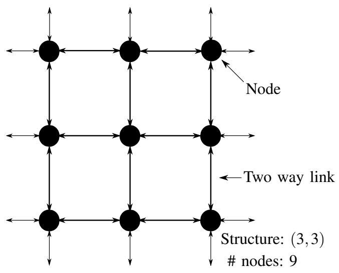


Fig. 3. The layout of an urban traffic network

is the blocking scheme, which groups the optimized variables into several blocks, in each block the variables are set equal to each other. By defining the blocking matrix, the number of the variables needed to be optimized is reduced, and the real-time computational complexity of the MPC controller also decreases. However, the control effect also deteriorates to some extent, because the aggregation constraint reduces the number of degrees of freedom of the optimization.

3) **Rolling horizon**. The optimal control input $\mathbf{g}^*(k_c)$ is derived from the optimization, then the first sample of the optimal results, $g^*(k_c \mid k_c)$, is transferred to the process and implemented. When arriving to the next control step k_c+1, the prediction model is fed with real measured traffic states, the whole prediction horizon is shifted one step forward, and the optimization starts over again. 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 uncertainty and disturbances.

IV. Simulation Experiments

As a macroscopic urban traffic model, BLX model is still accurate enough to follow the tendency of the traffic flow from microscopic traffic model, which is demonstrated in [10]. Therefore, we use the BLX model here to simulate the real traffic process, and design two MPC controllers taking the BLX model and the S model as prediction models respectively. However, the BLX model is still not fast enough when the size of the controlled network is getting larger. To reduce the computation time, the S model is proposed. Simulations are carried out to test whether the controller based on the S model can reduce the on-line optimization time compared with the controller based on the BLX model for both different network scales and different prediction time horizons. Moreover, the control effects are also investigated by comparing the TTS of the two MPC controllers with the TTS of a fixed-time (FT) control strategy.

The network considered is a grid-like network, where the "Structure" of the network is expressed as the number of rows and the number of columns, as "(line,column)". The number of the nodes in the network can be calculated as "line$\times$column". For example, Fig. 3 shows the layout of a $(3,3)$ network containing 9 nodes.

During the experiment, the simulation time interval of the BLX model is set to 1 s, while the simulation time intervals of the S model are cycle times which are 120 s, the same for all intersections in the network. For both controllers, the control time interval T_c is 120 s, and the control simulations run for the same time period of 1200s for all the experiments. The results of the experiments are listed in Table I-IV. Here, "tavrg" is the average optimization time over all the control steps, and "tmax" is the maximum optimization time.

As Table I and II show, the S model-based controller requires much less on-line optimization time than the BLX model-based controller for both different network scales and different prediction horizons. This shows that great improvement is made to the real-time computational feasibility by applying the S model as the prediction model of MPC controller. Moreover, the larger the traffic network is, the longer the controller predicts, and the more optimization time will be saved. Both the BLX model-based controller and the S model-based controller obtain a lower TTS, and have better control effect than the fixed-time control strategy. The "TTS ratio" in Table I and II shows the percentage of reduction of the TTS for the MPC controller compared with the fixed-time controller. From the ratio, we can see that the S model-based controller only sacrifices a limited amount of control effect, but obtains much faster on-line optimization speed. It guarantees the feasibility of implementing in practice the S model-based controller to larger urban traffic networks.

However, even for the S model-based controller, the on-line time feasibility becomes an issue again, when the network scale and the predictive horizon grow larger and longer. In this situation, we can define control horizon N_c, that is smaller than predictive horizon N_p, to reduce the number of optimized variables so as to reduce the on-line optimization time. As Table III shows, the optimization times are reduced further, but with little influence on the control effects. Instead of N_c, we can also adopt blocking technique. In Table IV, the blocking matrix is simply represented by a vector, in which each element gives the number of variables in the corresponding block. The results of Table IV show that better control effects can be obtained, if the blocking matrix is chosen properly.

V. Conclusions

We have applied Model Predictive Control (MPC) for coordinated control of urban traffic networks. MPC has many advantages, like robustness to disturbances, long-term sight, easy dealing with constraints, and so on. But despite of these advantages, it also inevitably gives rise to the problem of high on-line computational complexity. We propose two ways to address the scalability issue:

First, we simplify the urban traffic network model to reduce the computation time as much as possible, while guaranteeing that it is still accurate enough. MPC controllers are built based on both the simplified model and a macroscopic urban traffic model (BLX) with a higher level of modeling detail. Simulation experiment results show that the S model-based controller is much faster than the BLX model-based controller, both for different network scales and for different

TABLE I

SIMULATION RESULTS FOR DIFFERENT NETWORK SCALES WHEN $N_p = N_c = 10$ (HERE FT MEANS FIX-TIME CONTROL, BLX REFERS TO THE MODEL OF [4], [10], AND S REFERS TO THE SIMPLIFIED MODEL IN SECTION II)

Network	FT	BLX				S			
	TTS (veh· h)	tmax (s)	tavrg (s)	TTS (veh· h)	TTS ratio (%)	tmax (s)	tavrg (s)	TTS (veh· h)	TTS ratio (%)
(1,1)	100.1	138.8	17.4	66.58	33.47	0.8276	0.4916	67.56	32.50
(1,2)	192.1	462.5	60.7	134.3	30.12	3.495	1.257	135.9	29.28
(2,2)	365.5	646.7	162.3	267.5	26.82	12.79	6.726	273.1	25.27
(3,3)	770.8	8602.9	2137.4	600.3	22.12	255.0	96.51	617.3	19.91

TABLE II

SIMULATION RESULTS FOR DIFFERENT PREDICTIVE TIME HORIZONS WHEN NETWORK=(3,3) (HERE FT MEANS FIX-TIME CONTROL, BLX REFERS TO THE MODEL OF [4], [10], AND S REFERS TO THE SIMPLIFIED MODEL IN SECTION II)

$N_p = N_c$	FT	BLX				S			
	TTS (veh· h)	tmax (s)	tavrg (s)	TTS (veh· h)	TTS ratio (%)	tmax (s)	tavrg (s)	TTS (veh· h)	TTS ratio (%)
2	770.8	13.048	12.707	609.8	20.89	1.568	0.9229	619.7	19.60
4	770.8	123.81	51.218	610.5	20.80	29.12	12.52	622.8	19.20
6	770.8	1241.1	366.37	603.5	21.70	69.95	23.01	619.8	19.59
8	770.8	5280.4	865.56	604.1	21.63	110.9	40.15	618.9	19.72
10	770.8	8602.9	2137.4	600.3	22.12	255.0	96.51	617.3	19.91

prediction time horizons. Meanwhile, it only loses limited amount of the control effectiveness.

Second, we try different techniques to reduce the number of optimization variables, like defining a smaller control horizon and using different aggregation schemes, to improve the speed of solving the optimization problem. Better simulation results are obtained, and it is shown that these techniques can improve the real-time feasibility of the MPC controller.

In the future, we will do further analysis, simulations, and practical experiments on MPC controllers for large urban traffic network, taking also more realistic traffic conditions into consideration. Moreover, research will continue on developing more efficient optimization algorithms, so as to further improve the real-time feasibility.

VI. ACKNOWLEDGMENTS

Research supported by a Chinese Scholarship Council (CSC) grant, the National Science Foundation of China (Grant No. 60674041), the Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20070248004), the EU COST Action TU0702, EU project "Hierarchical and distributed model predictive control (HD-MPC)" (INFSO-ICT-223854), the BSIK projects TRANSUMO and NGI, the Delft Research Center Next Generation Infrastructures, and the Transport Research Centre Delft.

REFERENCES

[1] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," *Proceeding of the IEEE*, vol. 91, no. 12, pp. 2043–2067, 2003.

TABLE III

SIMULATION RESULTS FOR DIFFERENT CONTROL TIME HORIZONS WHEN NETWORK STRUCTURE=(3,3), $N_p = 10$

N_c	tmax (s)	tavrg (s)	TTS (veh· h)
$N_c = N_p = 10$	255.0	96.51	617.33
2	38.45	11.92	618.5
4	224.6	69.84	621.3
6	107.9	35.57	620.7

TABLE IV

SIMULATION RESULTS FOR DIFFERENT BLOCKING SCHEMES WHEN NETWORK STRUCTURE=(3,3), $N_p = 10$

Blocking	tmax (s)	tavrg (s)	TTS (veh· h)
None	255.0	96.51	617.3
[4 6]	132.3	26.86	612.8
[1 2 3 4]	227.3	67.64	621.9
[1 1 1 2 2 3]	329.6	93.87	622.0

[2] R. van Katwijk, P. van Koningsbruggen, B. De Schutter, and J. Hellendoorn, "Test bed for multiagent control systems in road traffic management," *Transportation Research Record*, no. 1910, pp. 108–115, 2005.

[3] M. Dotoli, M. P. Fanti, and C. Meloni, "A signal timing plan formulation for urban traffic control," *Control Engineering Practice*, vol. 14, no. 11, pp. 1297–1311, 2006.

[4] M. van den Berg, A. Hegyi, B. De Schutter, and J. Hellendoorn, "Integrated traffic control for mixed urban and freeway networks: A model predictive control approach," *European Journal of Transport and Infrastructure Research*, vol. 7, no. 3, pp. 223–250, 2007.

[5] N. H. Gartner, F. J. Pooran, and C. M. Andrews, "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, 2001, pp. 195–200.

[6] F. Boillot, S. Midenet, and J. C. Pierrelee, "The real-time urban traffic control system CRONOS: Algorithm and experiments," *Transportation Research Part C: Emerging Technologies*, vol. 14, no. 1, pp. 18–38, 2006.

[7] P. Mirchandani and L. Head, "A real-time traffic signal control system: Architrcture, algorithm, and analysis," *Transportation Research Part C: Emerging Technologies*, vol. 9, no. 6, pp. 415–432, 2001.

[8] A. Di Febbraro, D. Giglio, and N. Sacco, "Urban traffic control structure based on hybrid petri nets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 224–237, Dec. 2004.

[9] K. Aboudolas, M. Papageorgiou, and E. Kosmatopoulos, "Store-and-forward based methods for the signal control problem in large-scale congested urban road networks," *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 2, pp. 163–174, 2009.

[10] S. Lin and Y. Xi, "An efficient model for urban traffic network control," in *Proc. of the 17th World Congress The International Federation of Automatic Control*, Seoul, Korea, 2008, pp. 14 066–14 071.

[11] E. Camacho and C. Bordons, *Model Predictive Control in the Process Industry*. Berlin, Germany: Springer-Verlag, 1995.