Delft University of Technology

Delft Center for Systems and Control

Technical report 10-011

An efficient model-based method for coordinated control of 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, "An efficient model-based method for coordinated control of urban traffic networks," *Proceedings of the 2010 IEEE International Conference on Networking, Sensing and Control*, Chicago, Illinois, p. 8-13, Apr. 2010.

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/10_011.html`

An efficient model-based method for coordinated control of urban traffic networks

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

Abstract— **Traffic control is an effective and also efficient approach to reduce traffic jams. To alleviate the traffic congestion in an urban traffic network, a traffic control strategy that can coordinate the whole traffic network from a global point of view, is required. In this paper, an advanced control strategy, i.e. Model Predictive Control (MPC), is applied to control and coordinate urban traffic networks. However, the on-line computational complexity becomes a big challenge when the scale of the traffic network gets larger. To overcome this problem, the MPC control strategy is reformulated and solved efficiently on-line by a Mixed-Integer Linear Programming (MILP) solver. An MPC controller based on MILP is established and studied for the urban traffic network in different traffic scenarios. The simulation results show that the MILP-based MPC controller is a promising approach to reduce the on-line computational complexity of MPC controllers for urban traffic networks.**

I. Introduction

Urban areas are usually both population centers and economic centers. Some big cities have the highest population density and the busiest economics through the world. Moreover, urban areas are also the places where a high-standard transportation service is needed. However, urban areas are also the places where traffic congestion most likely happens, when people need to use the common infrastructures with limited capacity at the same time, especially during rush hours. Huge losses may be caused by traffic jams. Traffic delays grow because of the congestion, and the economic losses and the traffic pollution will also increase accordingly. Moreover, traffic congestion may also threaten the safety of the public transportation. Expanding the transportation infrastructure can alleviate the congestion to some extent, but it is too time and money consuming, and it is also limited by the existing geography of cities. Therefore, traffic control strategies are an attractive method to address congestion problems of urban areas.

Since the emergence of traffic control, traffic control strategies have gone through various developments from isolated intersection control to coordinated control, from fixed-time control to traffic-responsive control [1]. Isolated intersection controllers have been well developed as local controllers. However, even though the local controller works properly, it still 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 [1]. Fixed-time coordinated control strategies control and coordinate the control measures of the network based on the historical traffic data. But the fixed-time control strategy hardly respond to the real-time variations of the traffic demands and the disturbances. Traffic-responsive coordinated control strategies measure the traffic states in the network, and adapt real-time the control schemes according to the measured traffic states. Although the traffic-responsive control strategy can respond in real-time to the variations of traffic states, it is still a myopic control method, which does not look ahead. To overcome these disadvantages, more advanced coordinated control strategies, i.e. model-based optimization methods (including Model Predictive Control (MPC)), are applied to control urban traffic networks.

In the 1980s and 1990s, a number of model-based optimization control strategies emerged: OPAC, PRODYN, CRONOS, and RHODES. These control strategies predict the future traffic demands at the intersections through traffic flow data measured by upstream detectors and the detectors in upstream links. But the prediction horizon is influenced by the locations of the detectors. In recent years, some macroscopic urban traffic models, which can describe the traffic dynamic mechanics of the whole urban traffic network, have been developed. Model-based optimization control strategies [2], [3] based on these prediction models were developed, and avoid the problem of previous strategies. However, the biggest challenge for implementing the model-based optimization control strategy is the on-line computational complexity. Although these strategies have a lot of advantages, the on-line computational complexity is high.

In this paper, we reformulate and linearize a macroscopic urban traffic network model of [4] into mixed-integer linear equality and inequality constraints. An MPC controller is established based on the reformulated traffic model. Then, the original nonlinear optimization problem of the MPC controller is turned into a Mixed-Integer Linear Programming (MILP) problem, which can be solved by existing MILP solvers. Simulation experiments are carried out to test the MILP-based MPC controller.

II. Macroscopic urban traffic model (S model)

In the macroscopic urban traffic model of [4], called the S model, define J the set of nodes (intersections), and L the set

S. Lin and Y. Xi are with Department of Automation, Shanghai Jiao Tong University, Shanghai, P. R. China; S. Lin is also with 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@tudelft.nl, j.hellendoorn@tudelft.nl

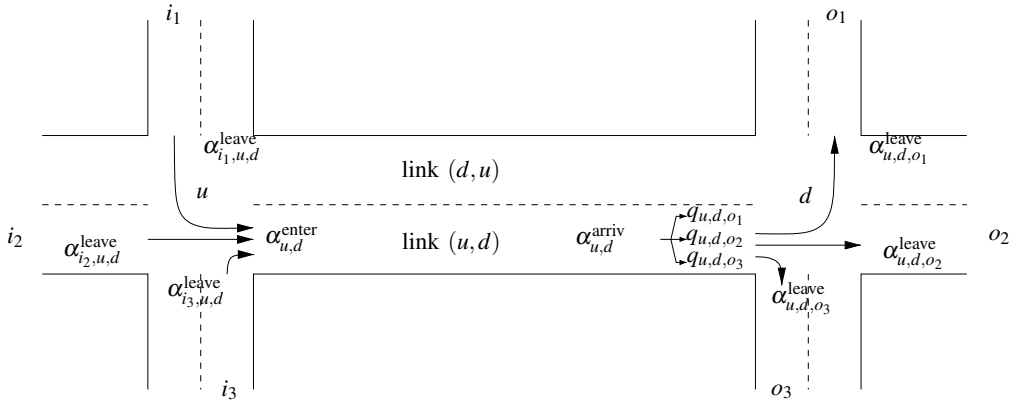


Fig. 1. A link connecting two traffic-signal-controlled intersections

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 input and output links of link (u,d) can be also specified by the upstream and downstream nodes. 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 at step k in link (u,d), $q_{u,d,o}$ is the queue length of the sub-stream turning to link o,
- $\alpha^{\text{leave}}_{u,d}(k)$: flow rate leaving link (u,d) at step k, $\alpha^{\text{leave}}_{u,d,o}(k)$ is the leaving flow rate of the sub-stream towards o,
- $\alpha^{\text{arriv}}_{u,d}(k)$: flow rate arriving at the end of the queue in link (u,d) at step k, $\alpha^{\text{arriv}}_{u,d,o}(k)$ is the arriving flow rate of the sub-stream towards o,
- $\alpha^{\text{enter}}_{u,d}(k)$: flow rate entering link (u,d) at step k, $\alpha^{\text{enter}}_{i,u,d}(k)$ is the flow rate entering link (u,d) from i,
- $\beta_{u,d,o}(k)$: relative fraction of the traffic 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 towards o in link (u,d),
- $v^{\text{free}}_{u,d}$: free-flow vehicle speed in link (u,d),
- $C_{u,d}$: capacity of link (u,d) expressed in number of vehicles,
- $N^{\text{lane}}_{u,d}$: number of lanes in link (u,d),
- $\Delta c_{u,d}$: offset between node u and node d, which represents the offset time between the cycle times of the upstream and the downstream intersections at the beginning of every control time step,
- l_{veh} : average vehicle length.

In S 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, as Fig. 2(a) illustrates. In this situation, the simulation step counters of different intersections are not same. As cycle times are the simulation time intervals, the input and output flow rates of the link are averaged over the cycle times in the S model.

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

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

The leaving average flow rate is the sum of the leaving flow rates $\alpha^{\text{leave}}_{u,d,o}(k_d)$ turning to each output link $o \in O_{u,d}$.

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

$$\begin{aligned}\alpha^{\text{leave}}_{u,d,o}(k_d) = \min\Big(& \beta_{u,d,o}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o}(k_d)/c_d, \\ & q_{u,d,o}(k_d)/c_d + \alpha^{\text{arriv}}_{u,d,o}(k_d), \\ & \beta_{u,d,o}(k_d)\left(C_{d,o} - n_{d,o}(k_d)\right)/c_d\Big) \ . \end{aligned} \quad (2)$$

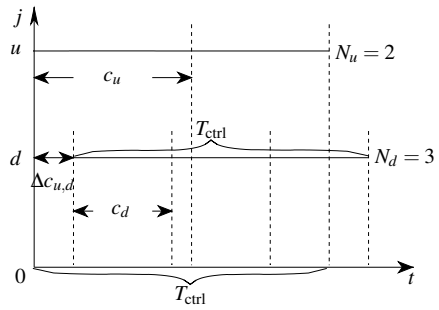
The number of vehicles waiting in the queue turning to link o is updated as

$$q_{u,d,o}(k_d+1) = q_{u,d,o}(k_d) + \left(\alpha^{\text{arriv}}_{u,d,o}(k_d) - \alpha^{\text{leave}}_{u,d,o}(k_d)\right) \cdot c_d \ . \quad (3)$$

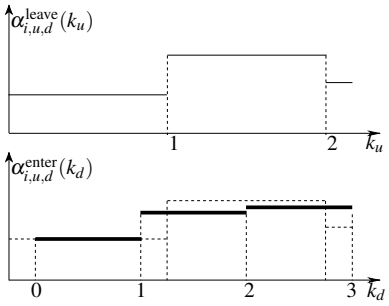
The flow of vehicles that entered link (u,d) will arrive at the end of the queues after a time delay $\tau(k_d) \cdot c_d + \gamma(k_d)$, i.e.,

$$\begin{aligned}\alpha^{\text{arriv}}_{u,d}(k_d) = & (1-\gamma(k_d)) \cdot \alpha^{\text{enter}}_{u,d}(k_d - \tau(k_d)) + \\ & \gamma(k_d) \cdot \alpha^{\text{enter}}_{u,d}(k_d - \tau(k_d) - 1), \end{aligned} \quad (4)$$

$$\begin{aligned}\tau(k_d) &= \text{floor}\left\{\frac{\left(C_{u,d} - q_{u,d}(k_d)\right) \cdot l_{\text{veh}}}{N^{\text{lane}}_{u,d} \cdot v^{\text{free}}_{u,d} \cdot c_d}\right\}, \\ \gamma(k_d) &= \text{rem}\left\{\frac{\left(C_{u,d} - q_{u,d}(k_d)\right) \cdot l_{\text{veh}}}{N^{\text{lane}}_{u,d} \cdot v^{\text{free}}_{u,d} \cdot c_d}\right\}. \end{aligned} \quad (5)$$



(a) Relationship between cycle times and control time interval



(b) Illustration for synchronizing flow rates

Fig. 2. Synchronization of upstream and downstream intersections

When reaching the end of the link, the arriving flow rate is separated into sub-streams by multiplying it with the turning rate $\beta_{u,d,o}(k_d)$.

The flow rate entering link (u,d) is the sum of the flow rates entering from all the upstream links:

$$\alpha^{\text{enter}}_{u,d}(k_d) = \sum_{i \in I_{u,d}} \alpha^{\text{enter}}_{i,u,d}(k_d) = \sum_{i \in I_{u,d}} \alpha^{\text{leave}}_{i,u,d}(k_u). \qquad (6)$$

In this formula, we see that the total flow rate entering link (u,d) is provided by the sum of the flow rates leaving the upstream links. Recall that we have different cycle times between the upstream and downstream intersections, so the simulation time steps are not the same between the leaving and entering flow rates. Some operations need to be carried out to synchronize the leaving and entering flow rates.

A common control time interval is adopted by all the intersections in the network, with N_j an integer, as

$$T_{\text{ctrl}} = N_j \cdot c_j, \quad \text{for all } j \in J \ . \qquad (7)$$

T_{ctrl} is defined as the Least Common Multiple of all the intersection cycle times in the traffic network, which ensures all the intersections can communicate with each other and be synchronized. For Fig. 2(a), we have $T_{\text{ctrl}} = N_u \cdot c_u = N_d \cdot c_d$.

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

$$\alpha^{\text{leave,cont}}_{i,u,d}(t) = \alpha^{\text{leave}}_{i,u,d}(k_u), \quad k_u \cdot c_u \le t < (k_u+1) \cdot c_u, \quad (8)$$

and then we convert the result again to obtain the average entering flow rates in time step k_d so as to make them can

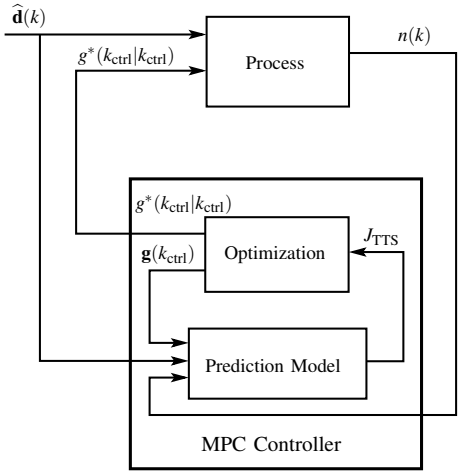


Fig. 3. The framework of the MPC controller

be used by the downstream link as follows

$$\alpha^{\text{enter}}_{i,u,d}(k_d) = \frac{\int_{k_d \cdot c_d + \Delta c_{u,d}}^{(k_d+1)\cdot c_d + \Delta c_{u,d}} \alpha^{\text{leave,cont}}_{i,u,d}(t)}{c_d} \, \mathrm{d}t \ . \qquad (9)$$

III. Model-based urban traffic network control

Model Predictive Control [5] is a methodology that implements and repeatedly applies Optimal Control in a rolling horizon way. The optimization problem to be solved is built based on the prediction model of the process and an estimate of the disturbances. In each control step, only the first control sample of the optimal control sequence is implemented; subsequently the horizon is shifted one sample and the optimization is restarted again with new information of the measurements.

Similar to Optimal Control, MPC can predict and approximate the optimal solution for the future. In contrast to Optimal Control, through the use of feedback and the rolling horizon approach, 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. 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. 3 shows the structure of the MPC controller and how it works on the process. The MPC approach can be described by the following three steps:

1) **Prediction model**. A model can be selected as prediction model for MPC controller, if it can sufficiently accurately 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, the traffic model presented in Section 2 above can be used as prediction model for the MPC controller. It can be generally described as

$$n(k+1) = f\big(n(k), g(k_{\text{ctrl}}(k)), d(k)\big) \qquad (10)$$

where $n(k)$ is the traffic state (the number of vehicles in a link at simulation time step k, see (1)) requested for 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; $g(k_{\text{ctrl}}(k))$ is the future control input, e.g. the green time splits. Having the control time interval T_{ctrl} and the simulation time interval T_{sim}, we define $T_{\text{ctrl}} = MT_{\text{sim}}$. For a given k, the corresponding value k_{ctrl} of the control time step is given by

$$k_{\text{ctrl}}(k) = \left\lfloor \frac{k}{M} \right\rfloor, \tag{11}$$

where $\lfloor x \rfloor$ with x a real number denotes the largest integer less than or equal to x. On the other hand, a given value k_{ctrl} of the control time step corresponds to the set $\{k_{\text{ctrl}}M, k_{\text{ctrl}}M+1, \cdots, (k_{\text{ctrl}}+1)M-1\}$ of simulation time steps.

2) **Optimization problem**. When the prediction horizon is N_{p}, the future traffic states are predicted at simulation time step k by the model as

$$\widehat{\mathbf{n}}(k) = [\widehat{n}^{\text{T}}(k+1|k)\ \widehat{n}^{\text{T}}(k+2|k)\ \cdots\ \widehat{n}^{\text{T}}(k+MN_{\text{p}}|k)]^{\text{T}},$$

based on the predicted traffic demands at simulation time step k

$$\widehat{\mathbf{d}}(k) = [\widehat{d}^{\text{T}}(k|k)\ \widehat{d}^{\text{T}}(k+1|k)\ \cdots\ \widehat{d}^{\text{T}}(k+MN_{\text{p}}-1|k)]^{\text{T}},$$

and the future traffic control inputs at control step k_{ctrl}

$$\begin{aligned}\mathbf{g}(k_{\text{ctrl}}) = &[g^{\text{T}}(k_{\text{ctrl}}|k_{\text{ctrl}})\ g^{\text{T}}(k_{\text{ctrl}}+1|k_{\text{ctrl}})\ \cdots \\ &g^{\text{T}}(k_{\text{ctrl}}+N_{\text{p}}-1|k_{\text{ctrl}})]^{\text{T}}\ .\end{aligned}$$

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

$$\begin{aligned}&\min_{\mathbf{g}(k_{\text{ctrl}})} J_{\text{TTS}} = J\big(\widehat{\mathbf{n}}(k), \mathbf{g}(k_{\text{ctrl}})\big) \\ &s.t.\quad \text{model constraints from (1) to (9)} \\ &\qquad \Phi(\mathbf{g}(k_{\text{ctrl}})) = 0 \quad \text{(cycle time constraint)} \\ &\qquad g_{\min} \le \mathbf{g}(k_{\text{ctrl}}) \le g_{\max}\end{aligned} \tag{12}$$

where the cycle time constraint holds for every intersection at each control time step and states that the cycle time equals to the sum of the green time splits for all the phases. The nonlinear optimization problem (12) can be solved by the (multi-start) Sequence Quadratic Programming (SQP) algorithm.

3) **Rolling horizon**. Once the optimal control input $\mathbf{g}^*(k_{\text{ctrl}})$ is derived from the optimization, the first sample of the optimal result, $g^*(k_{\text{ctrl}}|k_{\text{ctrl}})$, is transferred to the process and implemented. When arriving to the next control step $k_{\text{ctrl}}+1$, the prediction model is fed with the 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 to get feedback from the real traffic network, and makes the MPC controller robust to the uncertainty and disturbances.

IV. URBAN TRAFFIC MODEL REFORMULATION

Due to its nonlinear non convex nature, the optimization problem inevitably suffers from an exponentially growing computational complexity when the scale of the controlled traffic network grows. Moreover, a multi-start method is usually applied to achieve better result, which needs even more computing time. From the S model, we can conclude that the nonlinear feature of the traffic model is caused by the Max-Min-Plus-Scaling (MMPS) formula (2), and the nonlinear formulas (4), (8) and (9). An MMPS model can be equivalently transformed into a mixed-integer linear model [6], which can be expressed by mixed-integer equalities and inequalities. Therefore, the S model can be reformulated into a mixed-integer linear model. Then, the optimization problem of the MPC controller can be solved by MILP solvers efficiently.

Two assumptions are made to build the MILP problem: (i) We assume the estimated traffic state $\widehat{n}(k)$ by the traffic model stays constant during the simulation time interval T_{sim}, then the objective function TTS can be approximated as in (12); (ii) We assume the time delay of the vehicles traveling from the beginning of the link to the end of the queues in the link is constant, then (4) becomes linear as

$$\begin{aligned}\alpha^{\text{arriv}}_{u,d}(k_d) = &(1-\gamma_{\text{const}}) \cdot \alpha^{\text{enter}}_{u,d}(k_d - \tau_{\text{const}}) + \\ &\gamma_{\text{const}} \cdot \alpha^{\text{enter}}_{u,d}(k_d - \tau_{\text{const}} - 1)\,,\end{aligned} \tag{13}$$

where τ_{const} and γ_{const} are constant values obtained by (5) with the queue length fixed.

A. Rules for equivalent transformation into mixed-integer linear model

According to [6], consider the statement $f(x) \le 0$, where $f : \mathbb{R}^n \to \mathbb{R}$. Assume that $x \in \mathscr{X}$, where $\mathscr{X} \subset \mathbb{R}^n$ is a given bounded set, and define $M = \max_{x\in\mathscr{X}} f(x)$, $m = \min_{x\in\mathscr{X}} f(x)$.

Then, by introducing in $\delta \in \{0,1\}$, the following equivalence holds

$$[f(x) \le 0] \Leftrightarrow [\delta = 1] \text{ true iff } \begin{cases} f(x) \le M(1-\delta) \\ f(x) \ge \varepsilon + (m-\varepsilon)\delta \end{cases}, \tag{14}$$

where ε is a small tolerance, typically the machine precision.

Moreover, $\delta f(x)$ can be replaced by the auxiliary real variable $z = \delta f(x)$ which satisfies $[\delta = 0] \Rightarrow [y = 0]$, $[\delta = 1] \Rightarrow [y = f(x)]$. Then $z = \delta f(x)$ is equivalent to [6]

$$\begin{cases} z \le M\delta \\ z \ge m\delta \\ z \le f(x) - m(1-\delta) \\ z \ge f(x) - M(1-\delta) \end{cases}. \tag{15}$$

B. Model reformulation from MMPS model into mixed-integer linear model

With the equivalences above, MMPS formula (2) can be reformulated as mixed-integer linear equations and inequalities. Let

$$a = \beta_{u,d,o}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o}(k_d)/c_d$$

$$
\begin{aligned}
b &= q_{u,d,o}(k_d)/c_d + \alpha^{\text{arriv}}_{u,d,o}(k_d) \\
c &= \beta_{u,d,o}(k_d)\left(C_{d,o} - n_{d,o}(k_d)\right)/c_d \\
d &= \min(a,b),
\end{aligned} \tag{16}
$$

then (2) becomes

$$
\alpha^{\text{leave}}_{u,d,o}(k_d) = \min(a,b,c) = \min(d,c) \ . \tag{17}
$$

Let

$$
f_1 = b - a, \tag{18}
$$

and define

$$
\delta_1 = \begin{cases} 1 & \text{if } f_1 \le 0 \\ 0 & \text{if } f_1 > 0 \end{cases} \tag{19}
$$

then we have

$$
d = a + (b-a)\cdot\delta_1 = a + f_1\cdot\delta_1 \ . \tag{20}
$$

Similarly, let

$$
f_2 = c - d, \tag{21}
$$

and define

$$
\delta_2 = \begin{cases} 1 & \text{if } f_2 \le 0 \\ 0 & \text{if } f_2 > 0 \end{cases} \tag{22}
$$

then we have

$$
\min(d,c) = d + (c-d)\cdot\delta_2 = d + f_2\cdot\delta_2 \ . \tag{23}
$$

Let

$$
z_1 = f_1\cdot\delta_1 \tag{24}
$$

$$
z_2 = f_2\cdot\delta_2 \tag{25}
$$

and substitute (20) into (23), then (17) becomes linear as

$$
\alpha^{\text{leave}}_{u,d,o}(k_d) = a + z_1 + z_2 \ . \tag{26}
$$

Then, (19), (22), (24) and (25) can be equivalently rewritten into inequality constraints as in (14) and (15).The maximum value and the minimum value of f_1 are $M_1 = C_{u,d}/c_d$ and $m_1 = -\mu_{u,d}$, and the maximum value and the minimum value of f_2 are $M_2 = C_{d,o}/c_d$ and $m_2 = -\min(\mu_{u,d}, C_{u,d}/c_d)$.

Therefore, by introducing the additional auxiliary binary variables δ_1 and δ_2, and the auxiliary real variables f_1, f_2, z_1, and z_2, the original formula (2) in the urban traffic model is equivalently reformulated as mixed-integer linear equations (18), (21) and (26), and inequalities.

C. Reformulation of the model synchronization

In (9), $\alpha^{\text{leave,const}}_{i,u,d}(t)$ is a piecewise continuous function with intervals $\delta_\ell, \cdots, \delta_{\ell+N_\ell}$ and function values $\alpha^{\text{leave}}_{i,u,d}(\ell), \cdots, \alpha^{\text{leave}}_{i,u,d}(\ell+N_\ell)$. Hence,

$$
\alpha^{\text{enter}}_{i,u,d}(k_d) = \frac{1}{c_d}\sum_{j=0}^{N_\ell}\delta_{\ell+j}\alpha^{\text{leave}}_{i,u,d}(\ell+j), \tag{27}
$$

which is a linear expression in the $\alpha^{\text{leave}}_{i,u,d}$ values.

V. MILP-BASED MPC CONTROLLER

After the model reformulation, the optimization problem of the MPC controller in (12) can be rewritten as

$$
\begin{aligned}
\min_{\mathbf{u}(k_{\text{ctrl}})} \ & J_{\text{TTS}} = \sum_{k_d = Mk_{\text{ctrl}}+1}^{M(k_{\text{ctrl}}+N_{\text{p}})} \ \sum_{(u,d)\in L} c_d\cdot\widehat{n}_{u,d}(k_d) \\
s.t. \quad & \mathbf{A}\mathbf{u}(k_{\text{ctrl}}) \le \mathbf{b} \\
& \mathbf{A}_{\text{eq}}\mathbf{u}(k_{\text{ctrl}}) = \mathbf{b}_{\text{eq}} \\
& \mathbf{u}_{\min} \le \mathbf{u}(k_{\text{ctrl}}) \le \mathbf{u}_{\max}
\end{aligned} \tag{28}
$$

where $\widehat{n}_{u,d}(k)$ is the estimated number of vehicles on link (u,d) at time step k, and the TTS is a linear objective function of $\widehat{n}_{u,d}(k_d)$; the constraints contain the linear inequality constraints, the linear equality constraints, and the lower and upper bounds of the optimization variables.

For any link $(u,d) \in L$, $I_{u,d}$ and $O_{u,d}$ are the sets of input nodes and output nodes to link (u,d) respectively. In (28), the inequality constraints are the mixed-integer inequality constraints obtained through Section IV-A and IV-B for all the traffic streams in the network and for all the predicted simulation time steps in future (i.e. for $k_d, k_d+1, \cdots, k_d+MN_{\text{p}}-1$). The equality constraints contain the linear traffic states update equations (1) and (3), the linear equations (18), (21) and (26) obtained from the model reformulation, the linearized equation (13), the pre-specified reformulated synchronization equations (27), and the cycle time constraints for the green times of every intersection.

The vector of optimized variables at control time step k_{ctrl} in optimization problem (28) is

$$
\begin{aligned}
\mathbf{u}(k_{\text{ctrl}}) = [& u^{\text{T}}(k_{\text{ctrl}}|k_{\text{ctrl}}) \ u^{\text{T}}(k_{\text{ctrl}}+1|k_{\text{ctrl}}) \ \cdots \\
& u^{\text{T}}(k_{\text{ctrl}}+N_{\text{p}}-1|k_{\text{ctrl}})]^{\text{T}} \ ,
\end{aligned} \tag{29}
$$

where $u(k_{\text{ctrl}})$ at control step k_{ctrl} consists of control variables (i.e. green time splits), state variables, and auxiliary variables for all the nodes and links in the network. All the optimized variables are real values except the binary variables $\delta_1(k_{\text{ctrl}})$ and $\delta_2(k_{\text{ctrl}})$. Supplied with initial traffic states and traffic demands of the network, the optimization problem can be solved at the control time step k_{ctrl} by the MILP solver. The optimal control inputs for the first control time step will be implemented to the traffic network. Rolling one step ahead, a new MILP optimization problem will be built, etc.

VI. CASE STUDIES

Experiments are designed to compare the MILP-based MPC controller and the SQP-based MPC controller. We use the BLX model of [7] to simulate the real traffic process, and design MPC controllers based on the two optimization algorithms to derive the control schemes for the BLX model. Although it is a macroscopic urban traffic model, the BLX model is still accurate enough, which is demonstrated in [7], [8] by comparing it with a microscopic traffic model.

The urban traffic network investigated is a grid network including 4 nodes. The cycle time of the odd numbered nodes is 120 s, the cycle time of the even numbered nodes is 60 s. The control interval is 120 s. The prediction horizon

is 10 control intervals. The control simulations run for the same time period of 1200s for all the experiments. Different traffic scenarios are studied by initializing the traffic network with different levels of traffic densities. The results of the experiments for unsaturated and saturated scenarios are listed in Table I. Here, "tavrg" is the average computing time over all the control steps, and "tmax" is the maximum computing time. The cost function is the TTS for the entire simulation. The number of initial points for the SQP algorithm is 5.

TABLE I

COMPARISON OF COST FUNCTIONS AND COMPUTING TIMES FOR DIFFERENT OPTIMIZATION ALGORITHMS IN THE SATURATED AND THE UNSATURATED TRAFFIC SCENARIOS

Scenario	Algorithm	TTS (veh·h)	tavrg (s)	tmax (s)
Saturated	SQP	1755.9	453.4	552.5
	MILP	1744.4	2.7	3.3
Unsaturated	SQP	639.7	452.4	526.4
	MILP	654.2	3.9	4.6

As Table I shows, for the saturated scenario, the MILP approach obtains a lower TTS than the SQP approach within a dramatically shorter computing time, only a few seconds. So in this case, MILP is an efficient approach to solve the real-time optimization problem of the MPC controller, and, at the same time, it also obtains even better control effects than the multi-start SQP approach. However, for the unsaturated scenario, although the computing time of the MILP approach is still short, the TTS fails to be better than that of the SQP approach. This is mostly caused by assumption (ii) made during the model reformulation. Recall that in order to transform the optimization problem into an MILP problem, assumption (ii) is made to linearize the original model. In the assumption, the time delay for vehicles running from the beginning of the link to the end of the queues is considered to be constant. In the saturated scenario, the number of leaving vehicles depends on the saturated flow rate of the link. In that case, the assumption almost does not have influence on the results of MILP. However, in the unsaturated scenario, the number of leaving vehicles depends mainly on the number of waiting vehicles in the queues, which will be affected by the vehicles arriving from upstream after a certain time delay in the link. Therefore, the assumption caused a mismatch of the reformulated MILP problem from the original optimization problem. As a result, the MILP approach fails to achieve better results than the SQP approach in the unsaturated scenario. But, considering that the MILP approach is very fast, that the relative difference in performance is only 2.3%, and that the constant time delay can be refined and calibrated beforehand, the MILP approach is still an excellent choice for the MPC controller in the unsaturated scenario.

VII. CONCLUSIONS

An MPC controller has been established to control and coordinate urban traffic networks. To reduce the on-line computational complexity, the nonlinear traffic network model is linearized and reformulated into mixed-integer linear equations and inequalities. The reformulated optimization of the MPC controller can be solved much faster by MILP solvers.

In the simulation experiments, the MILP-based approach is compared with the SQP-based approach. The simulation results show that the MILP-based approach reduces the computing time dramatically in both saturated and unsaturated scenarios. Despite of the limited performance loss under unsaturated scenario, the MILP-based approach is a competitive method to reduce the on-line computational complexity of MPC controllers for large urban traffic networks.

In the future, more experiments will be carried out to compare MILP with other optimization algorithms, and further study the MILP-based MPC controller for urban traffic networks.

VIII. ACKNOWLEDGMENTS

This research is supported by a Chinese Scholarship Council (CSC) grant, the National Science Foundation of China (Grant No. 60674041, 60934007), the European COST Action TU0702, the 7th framework STREP project "Hierarchical and distributed model predictive control (HD-MPC)" (contract number INFSO-ICT-223854), the BSIK projects "Transition to Sustainable Mobility (TRANSUMO)" and "Next Generation Infrastructures (NGI)", the Delft Research Center Next Generation Infrastructures, and the Transport Research Center Delft.

REFERENCES

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

[2] M. Dotoli, M. P. Fanti, and C. Meloni, "A signal timing plan formulation for urban traffic control," *Ctrl. Eng. Prac.*, vol. 14, no. 11, pp. 1297–1311, 2006.

[3] 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," *Eur. J. Transp. Infrastructure Res.*, vol. 7, no. 3, pp. 223–250, 2007.

[4] S. Lin, B. De Schutter, Y. Xi, and J. Hellendoorn, "A simplified macroscopic urban traffic network model for model-based predictive control," in *Proc. 12th IFAC Symp. Ctrl. Transp. Syst.*, Redondo Beach (CA), USA, 2009, pp. 286–291.

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

[6] D. Christiansen, *Electronics Engineers' Handbook*, 4th ed. New York: IEEE Press/McGraw Hill, 1997.

[7] S. Lin and Y. Xi, "An efficient model for urban traffic network control," in *Proc. 17th World Cong. Int. Fed. Autom. Ctrl.*, Seoul, Korea, 2008, pp. 14 066–14 071.

[8] S. Lin, Y. Xi, and Y. Yang, "Short-term traffic flow forecasting using macroscopic urban traffic network model," in *Proc. 11th Inter. IEEE Conf. Intell. Transp. Syst.*, Beijing, China, 2008, pp. 134–138.