Delft University of Technology

Delft Center for Systems and Control

Technical report 15-033

Sustainable model-predictive control in urban traffic networks: Efficient solution based on general smoothening methods*

A. Jamshidnejad, I. Papamichail, M. Papageorgiou, and B. De Schutter

If you want to cite this report, please use the following reference instead:

A. Jamshidnejad, I. Papamichail, M. Papageorgiou, and B. De Schutter, "Sustainable model-predictive control in urban traffic networks: Efficient solution based on general smoothening methods," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 3, pp. 813–827, May 2018. doi:10.1109/TCST.2017.2699160

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.bartdeschutter.org/abs/15_033.html

Sustainable model-predictive control in urban traffic networks: Efficient solution based on general smoothening methods

Anahita Jamshidnejad, Ioannis Papamichail, Markos Papageorgiou, Bart De Schutter

***Abstract*—Traffic-responsive control approaches, including model-predictive control, are efficient methods for making the best use of the available network capacity. Moreover, gradient-based approaches, which can be applied to smooth optimization problems, have proven their efficiency, both computationally and performance-wise, in finding optima of optimization problems. In this paper, we propose a model-predictive control system for an urban traffic network that applies a gradient-based optimization approach to solve the control optimization problem. The controller uses a new smooth integrated flow-emission model to find a balanced trade-off between reduction of the congestion and of the total emissions. We also introduce efficient smoothening methods for nonsmooth mathematical models of physical systems. The effectiveness of the proposed approach is demonstrated via a case study.**

***Index Terms*—model-predictive control, smoothening, gradient-based optimization, urban traffic control.**

I. Introduction

Traffic congestion increases the fuel consumption, since vehicles will idle in standing queues for a while. In addition to that, these vehicles will distribute harmful substances such as nitrogen oxides (NO_x), hydrocarbon (HC), carbon monoxide (CO) and dioxide (CO_2) in the environment [1]–[4]. The mentioned consequences are the source of huge economical and environmental cost for modern societies. Hence, from the economical and environmental points-of-view, efficient actions should be undertaken to reduce both traffic congestion and emissions, especially in urban areas.

Inefficient use of the available capacity of the urban traffic networks is one of the main reasons of traffic congestion [5]. Real-time traffic-responsive control strategies, including optimization-based and especially model-predictive control (MPC) approaches, can be used to manage the road capacity in both freeways and urban areas [6]–[10]. MPC is an optimization-based control approach that originates from the process industry, and unlike regular optimal control that works in an entirely open-loop scheme, it uses at every time step new measurements of the outputs and/or states of the system. Hence, a system that is controlled via a model-predictive controller is more robust towards external disturbances.

The focus of this paper is on the development of MPC strategies for urban traffic networks in order to reduce both congestion and emissions. The MPC controller needs a model of the system that is accurate, and at the same time simple and fast for real-time applications. Different models have been proposed for urban traffic flow modeling, such as the store-and-forward model [11], which has also been used in [12] and [6]; the BLX model [13]; the S-model, and a macroscopic urban traffic flow model developed by Barisone et al. in [14].

Moreover, to solve the optimization problem of the MPC controller, we want to apply efficient gradient-based approaches. However, many of the available mathematical models, including traffic models, have nonsmooth functions or discrete variables in their formulations. Applying these models as prediction models of MPC controllers results in a nonsmooth MPC optimization problem, which restricts application of gradient-based approaches, e.g., methods that are based on Pontryagin's principle [15], [16].

Hence, we propose general smoothening methods for mathematical models of physical systems. For illustration purposes, we consider the S-model [17], which is a nonlinear flow model for urban traffic networks. However, we should note that the proposed smoothening methods can be applied to many other mathematical models too.

One of the efficient gradient-based approaches that has been widely used [18], [19] is the feasible direction algorithm proposed by Papageorgiou et al. [20]. To identify an efficient search direction for the optimization, we will apply the latest version of a well-known approach called the resilient back-propagation (RProp) given by Bailey in [21], while the main approach had been initially introduced by Riedmiller et al. in [22].

The structure of this paper is as follows; in Section II, we introduce general smoothening approaches for mathematical models of physical systems that involve nonsmooth functions. Section III introduces a general approach for transforming a time-delayed differential equation with time-varying time delays into an equivalent discrete-time time-delayed difference equation. Section IV discusses the urban traffic flow model, called the S-model, used in this paper for estimation of the urban traffic flows. We propose extensions to the S-model that allows the model perform more accurately, and for more possible traffic scenarios. Section V proposes a new integrated framework that results in an urban traffic flow and emission model that can be used as the prediction model of an MPC controller. In Section VI an MPC controller for urban traffic networks is developed that finds a balanced trade-off between reduction of the total time spent and total emissions of the vehicles. We also consider a gradient-based optimization solver that benefits from the resilient back-propagation (RProp) method. Section VII presents the results of a case study.

Contributions of the paper

We propose general smoothening methods for mathematical models of physical systems, where the proposed methods can be applied to various models with nonsmooth functions, including transportation models.The resulting smooth models can be used in model-based optimal control techniques, e.g., in MPC, where the optimization problem can be solved by efficient smooth and gradient-based solvers.

We also develop a general formulation for transforming a time-delayed system with a time-varying delay given in the continuous-time domain into an equivalent time-delayed system in the discrete-time domain. The resulting formulation can be used in discrete-time physical models of systems that involve time-varying time delays, and can give much more accurate results than a time-independent delay approximation.

Implementing the proposed general smoothening methods, we show that even for nonsmooth and nonlinear flow models, we can easily apply an efficient gradient-based optimization approach that uses resilient back-propagation (RProp) and solves the optimization problem of the MPC controller much faster than nonsmooth optimization approaches. Finally, all these methods are combined to develop a novel approach to urban traffic network control aiming at optimizing a trade-off of congestion mitigation versus emission reduction.

II. General smoothening methods for mathematical models of physical systems

Many of the available mathematical models of physical systems, including transportation networks, involve non-smooth functions in their formulations. Some examples of such transportation models include the cell transmission model (CTM) [23], the dynamic IFTN model [24] applied in modeling the container flows, and also max-plus-linear models, e.g., see [25], that are used to model railway transportation systems. It is very beneficial if we can use these models as the prediction model of an MPC controller, and still apply efficient gradient-based approaches (e.g., Pontryagin's approach [15]) to solve the optimization problem of the MPC. For this aim, we first need to find general approaches to render these non-smooth functions smooth and hence, differentiable. In this section, we develop smooth approximate functions for the nonsmooth indicator, maximum, minimum, floor, ceiling, and remainder functions that appear frequently in mathematical models of physical systems.

A. Indicator function

The indicator function $I_{x\geq a}$ is defined by

$$I_{x\geq a}=\begin{cases} 1, & x\geq a \\ 0, & x<a \end{cases}, \tag{1}$$

where in [26] the smooth form of the indicator function is approximated by a sigmoid function (see the first plot of Figure 1) $I_{x\geq 0}\approx (1+e^{-\alpha x})^{-1}$ with $\alpha>0$ a smoothening parameter.

B. Maximum and minimum functions

It is easy to verify that for $\alpha \gg 1$ we have (also see the second plot of Figure 1)

$$\max_{i=1,\dots,n}\{x_i\}\approx \frac{1}{\alpha}\log\sum\nolimits_{i=1}^{n} e^{\alpha x_i}, \tag{2}$$

$$\min_{i=1,\dots,n}\{x_i\}\approx -\frac{1}{\alpha}\log\sum\nolimits_{i=1}^{n} e^{-\alpha x_i}. \tag{3}$$

C. Floor and ceiling functions

The smooth form of the floor and ceiling functions can be constructed from pieces of differently transformed sigmoid functions:

$$\lfloor x\rfloor \approx -0.5+\sum_{k\in\mathbb{Z}}\left(\left(1+e^{-\alpha(x-k)}\right)^{-1}-0.5\right), \tag{4}$$

$$\lceil x\rceil \approx \quad 0.5+\sum_{k\in\mathbb{Z}}\left(\left(1+e^{-\alpha(x-k)}\right)^{-1}-0.5\right), \tag{5}$$

with $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ indicating the floor and ceiling function, respectively. The third plot of Figure 1 illustrates the floor function (solid curve) and its smooth form (red dotted curve).

D. Remainder function

The remainder function $\mathrm{rem}\{a,b\}$ yields the remainder value of the division of a by b, i.e., $\mathrm{rem}\{a,b\}=a-qb$, where q is the quotient. The fourth plot of Figure 1 illustrates the graph representing $\mathrm{rem}\{a,b\}$ for $b=2$. We consider the concept of Euclidean division:

- for a positive divisor $b>0$, we have $q=\left\lfloor \frac{a}{b}\right\rfloor$,
- for a negative divisor $b<0$, we have $q=\left\lceil \frac{a}{b}\right\rceil$.

The above expressions can equivalently be expressed by

$$q=\frac{1+\mathrm{sign}(b)}{2}\left\lfloor\frac{a}{b}\right\rfloor+\frac{1-\mathrm{sign}(b)}{2}\left\lceil\frac{a}{b}\right\rceil, \tag{6}$$

with $\mathrm{sign}(x)=\begin{cases} 1 & x>0 \\ -1 & x<0 \end{cases}$. We can write $\mathrm{sign}(x)=I_{x\geq 0}-I_{x\leq 0}$. Therefore,

$$\mathrm{sign}(x)\approx \left(1+e^{-\alpha x}\right)^{-1}-\left(1+e^{\alpha x}\right)^{-1}, \tag{7}$$

and for the remainder function we obtain

$$\begin{aligned}\mathrm{rem}\{a,b\}\approx & a-b\sum_{k\in\mathbb{Z}}\left(\left(1+e^{-\alpha\left(\frac{a}{b}-k\right)}\right)^{-1}-0.5\right)\\ &+\frac{b}{2}\left(\left(1+e^{-\alpha b}\right)^{-1}-\left(1+e^{\alpha b}\right)^{-1}\right).\end{aligned} \tag{8}$$

III. Discretization of a differential equation with time-varying time delay

Consider the time-delayed differential equation

$$\dot{x}(t)=Ax(t)+Bu(t-\tau(t))+Ce(t), \tag{9}$$

with $u(\cdot)$ and $e(\cdot)$ two continuous-time functions representing the inflow and outflow functions of the dynamic system. We should first discretize the model to find $\tilde{x}(k+1)$ from the current state $\tilde{x}(k)$, where the discrete-time variables are identified by a tilde, and the time step counter is denoted by k. Additionally, $\tau(k)=\delta(k)h+\gamma(k)$, with h the sampling time and

$$\delta(k)=\left\lfloor \tau(k)/h\right\rfloor,\ \gamma(k)=\mathrm{rem}\{\tau(k),h\}, \tag{10}$$

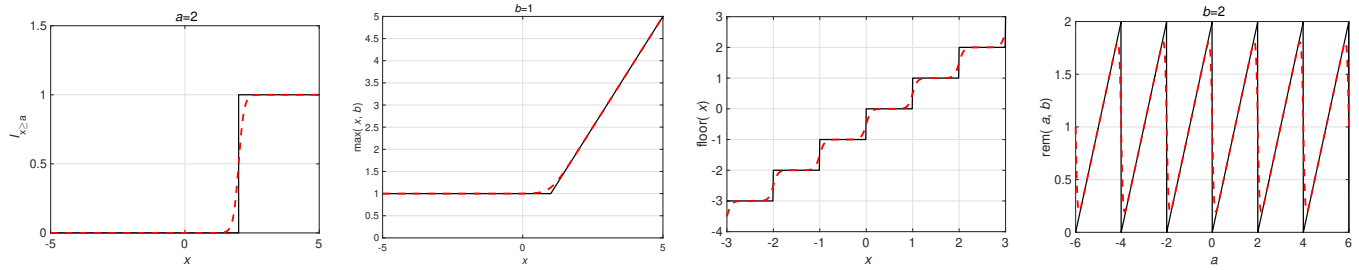

Figure 1. Nonsmooth functions (black curves) and their smooth approximators (red curves); first plot (from left): indicator function, second plot: maximum function, third plot: floor function, fourth plot: remainder function.

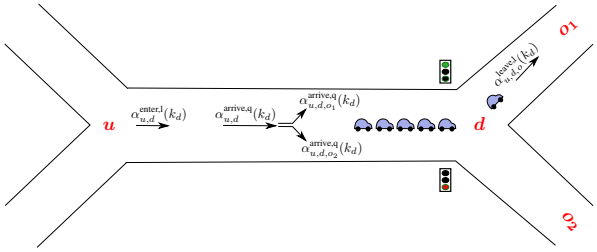

Figure 2. Illustration of a link in an urban traffic network at time step k_d, and with the entering, arriving, and leaving flows.

Lemma III.1. *The equivalent difference equation in the discrete-time domain of the delayed differential equation* (9)-(10)*, and with an input function* $u(t)$ *that is piece-wise constant in the interval* $[ch, (c+1)h)$, $c \in \mathbb{Z}$, *is*

$$\tilde{x}(k+1) = \tilde{A}\tilde{x}(k) + \sum_{i=0}^{\delta(k-1)-\delta(k)+1} \tilde{B}_i(k)\tilde{u}(k-\delta(k-1)-2+i) + \tilde{C}\tilde{e}(k), \tag{11}$$

with $\tilde{B}_0(k) = \gamma(k-1)/h$, $\tilde{B}_i(k) = 1$ *for* $i \in \{1, \ldots, \delta(k-1)-\delta(k)\}$, *and* $\tilde{B}_{\delta(k-1)-\delta(k)+1}(k) = (h-\gamma(k))/h$.

For the proof, see Appendix A.

IV. Urban Traffic Flow Model

A. Original S-model

The S-model is a nonlinear and discrete-time urban traffic flow model that provides an appropriate balance between low computation time and desirable accuracy. The model was introduced by Lin et al. [17], where the simulation sampling time of the S-model is one cycle time of the downstream intersection of a link. Therefore, the model updates the states less frequently than other macroscopic traffic models that typically have a sampling time of 1 s (such as the BLX model [13], the model of Kashani and Saridis [27], and the model proposed by van den Berg et al. [9]). This characteristic that helps the S-model to be faster for model-based control applications, and to provide a trade-off between accuracy and computation time. Moreover, the simulation sampling time for each link may differ from other links.

In the S-model, a network is modeled as a collection of nodes and links, where each node represents an intersection and each link represents a road. We use the pair (u, d) to indicate a link with node u as its starting node (i.e., the upstream intersection of the corresponding road), and node d as its end node (i.e., the downstream intersection of the corresponding road). The set of all links and intersections within the network are denoted by, respectively, $\mathcal{L}$ and $\mathcal{N}$, and the state variables of the model include the total number of vehicles $n_{u,d}(k_d)$ on link (u, d), and the number of vehicles $q_{u,d,o}(k_d)$ in the queue on link (u, d) that intend to move to the outgoing link (d, o) at time step k_d for all $(u, d) \in \mathcal{L}$. Both $n_{u,d}(k_d)$ and $q_{u,d,o}(k_d)$ are given in "number of vehicles". The set of all downstream nodes of the outgoing links of link (u, d) is denoted by $\mathcal{O}_{(u,d)}$, and the set of all upstream nodes of the incoming links of link (u, d) is denoted by $\mathcal{I}_{(u,d)}$.

The state variables of the S-model are updated at every simulation time step k_d of the link (u, d) by

$$n_{u,d}(k_d+1) = n_{u,d}(k_d) + \left(\alpha^{\text{enter,l}}_{u,d}(k_d) - \alpha^{\text{leave,l}}_{u,d}(k_d)\right) c_d, \tag{12}$$

$$q_{u,d,o}(k_d+1) = q_{u,d,o}(k_d) + \left(\alpha^{\text{arrive,q}}_{u,d,o}(k_d) - \alpha^{\text{leave,l}}_{u,d,o}(k_d)\right) c_d, \tag{13}$$

with $q_{u,d}(k_d) = \sum_{o \in \mathcal{O}_{(u,d)}} q_{u,d,o}(k_d)$. Note that c_d is the cycle time of the downstream intersection d, $\alpha^{\text{enter,l}}_{u,d}(k_d)$ and $\alpha^{\text{leave,l}}_{u,d}(k_d)$ are the average entering and exiting flow rates of link (u, d) within the time interval $[k_d c_d, (k_d+1)c_d)$, and $\alpha^{\text{arrive,q}}_{u,d,o}(k_d)$ and $\alpha^{\text{leave,l}}_{u,d,o}(k_d)$ are the average arriving flow rate at the tail of the waiting queue and the average leaving flow rate of the sub-stream that intends to move towards o within $[k_d c_d, (k_d+1)c_d)$. Note that these flow rates are computed for all the intermediate links of the S-model using the equations that will be given in this section, except for $\alpha^{\text{enter,l}}$ of the source links and $\alpha^{\text{leave,l}}$ of the exit links in the network. The entering flow rates of the source links are indeed the demand profiles that should either be given to the network as an input, or that should be determined by a prediction model. Similarly, the leaving flow rates of the exit links should be defined/given as the boundary conditions of the network or should be predicted by a model. Moreover, $q_{u,d}(k_d)$ is the total number of vehicles waiting in the queue on link u, d at time step k_d (note that $n_{u,d}$ and $q_{u,d,o}$ admit real non-negative values in the S-model). Moreover, we have (also see Figure 2):

$$\alpha^{\text{enter,l}}_{u,d}(k_d) = \sum_{i \in \mathcal{I}_{(u,d)}} \alpha^{\text{enter,l}}_{i,u,d}(k_d), \tag{14}$$

$$\alpha^{\text{leave,l}}_{u,d}(k_d) = \sum_{o \in \mathcal{O}_{(u,d)}} \alpha^{\text{leave,l}}_{u,d,o}(k_d), \tag{15}$$

$$\alpha^{\text{leave,l}}_{u,d,o}(k_d) = \min \Bigg(\beta_{u,d,o}(k_d)\mu_{u,d}g_{u,d,o}(k_d)/c_d, \\ q_{u,d,o}(k_d)/c_d + \alpha^{\text{arrive,q}}_{u,d,o}(k_d), \\ \beta_{u,d,o}(k_d) \Big/ \sum_{i\in\mathcal{I}_{(d,o)}} \beta_{i,d,o}(k_d) \frac{C_{d,o} - n_{d,o}(k_d)}{c_d} \Bigg), \tag{16}$$

$$\alpha^{\text{arrive,q}}_{u,d}(k_d) = \\ \left(c_d - \gamma_{u,d}(k_d)\right)/c_d \sum_{i\in\mathcal{I}_{(u,d)}} \alpha^{\text{leave,l}}_{i,u,d}(k_d - \delta_{u,d}(k_d)) + \\ \gamma_{u,d}(k_d)/c_d \sum_{i\in\mathcal{I}_{(u,d)}} \alpha^{\text{leave,l}}_{i,u,d}(k_d - \delta_{u,d}(k_d) - 1), \tag{17}$$

$$\alpha^{\text{arrive,q}}_{u,d,o}(k_d) = \beta_{u,d,o}(k_d) \cdot \alpha^{\text{arrive,q}}_{u,d}(k_d), \tag{18}$$

where

$$\delta_{u,d}(k_d) = \left\lfloor \tau_{u,d}(k_d)/c_d \right\rfloor, \tag{19}$$

$$\gamma_{u,d}(k_d) = \text{rem}\{\tau_{u,d}(k_d), c_d\}, \tag{20}$$

with $\mu_{u,d}$ the saturated flow rate of link (u,d), $g_{u,d,o}(k_d)$ the green time length during $[k_d c_d, (k_d+1)c_d)$ for the traffic sub-stream that leaves link (u,d) towards node o, $\beta_{u,d,o}(k_d)$ the fraction of vehicles within link (u,d) that intend to turn to o, $\alpha^{\text{arrive,q}}_{u,d}(k_d)$ the average within $[k_d c_d, (k_d+1)c_d)$ of the flow rate of vehicles arriving at the tail of the queue in link (u,d), and $\tau_{u,d}(k_d)$ the average delay time (from now on, we just call it the delay time) of the vehicles on link (u,d) within the interval $[k_d c_d, (k_d+1)c_d)$, i.e., the time vehicles entering the link need to reach the tail of the waiting queue. For more details about the S-model, we refer the readers to [17].

In the S-model the simulation sampling time of different links might not be the same and hence, their time steps may not be synchronized automatically. However, especially for the neighboring links, for which some variables are shared, synchronization of the joint variables between neighboring links is essential. For instance, $\alpha^{\text{leave,l}}_{i,u,d}(k_d)$ for link (i,u) forms a fraction of $\alpha^{\text{enter,l}}_{u,d}(k_d)$ for link (u,d), while $\alpha^{\text{leave,l}}_{i,u,d}$ is updated only at $k_u \in \mathbb{K}_u$, which may not be synchronized with $k_d \in \mathbb{K}_d$.

Synchronization of the joint variables: Figure 3 illustrates $\alpha^{\text{leave,l}}_{i,u,d}(\cdot)$, which is updated at time steps $k_u \in \mathbb{K}_u$, while we need to find $\alpha^{\text{enter,l}}_{i,u,d}$ at time step k_d. We first make the discrete-time functions $\alpha^{\text{leave,l}}_{i,u,d}$ and $\alpha^{\text{enter,l}}_{i,u,d}$ continuous in time, assuming that the continuous-time functions $\alpha^{\text{leave,l,c}}_{i,u,d}(\cdot)$ and $\alpha^{\text{enter,l,c}}_{i,u,d}(\cdot)$ are piecewise constant. Hence, $\alpha^{\text{leave,l,c}}_{i,u,d}(\cdot)$ between any two consecutive time steps k_u and k_u+1 equals $\alpha^{\text{leave,l}}_{i,u,d}(k_u)$. Similarly, $\alpha^{\text{enter,l,c}}_{i,u,d}(\cdot)$ between any two consecutive time steps k_d and k_d+1 equals $\alpha^{\text{leave,l}}_{i,u,d}(k_d)$. Since the number of vehicles that leave link (i,u) and enter link (u,d) between any two consecutive time steps k_d and k_d+1 are equal, the highlighted surfaces in Figure 3 should have the same area, i.e.,

$$\alpha^{\text{enter,l}}_{i,u,d}(k_d) = \frac{1}{c_d} \int_{k_d c_d}^{(k_d+1)c_d} \alpha^{\text{leave,l,c}}_{i,u,d}(t) \cdot \mathrm{d}t. \tag{21}$$

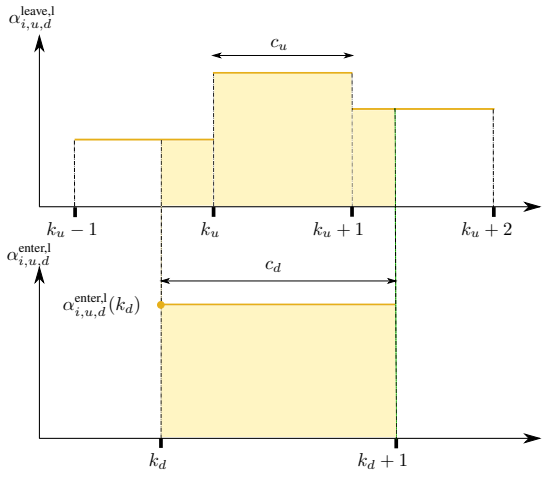

Figure 3. Synchronization of joint variables for the neighboring links.

B. Extensions for the S-model

In the original S-model, only the dynamics of the system within the network of interest is considered. However, the dynamics of the source nodes, which are located on the boundaries of the network, is missing. This may not bring issues when the source links of the network are under-saturated. However, for larger demands at the source nodes that may result in saturated and over-saturated links, defining the appropriate dynamics for the boundaries of the network is essential. This is because the additional vehicles that cannot immediately enter the network should not be injected into the entering links. However, this issue occurs with the original S-model and may lead to negative states with large absolute values. In contrast, those vehicles should be kept in queues that will gradually form at the sources of the network, and they should be injected into the network only when the available free space of the source links allows it.

In addition, in the original S-model the formula for the computation of the arriving flow $\alpha^{\text{arrive,q}}_{u,d,o}(k_d)$ (see (17)) is extracted based on the (hidden) assumption that the delay times for the vehicles are time-independent (see [28] for more details). Numerical experiments show that with this assumption and by using (17), in some cases the error in computation of the updated states can grow up to 50% (this error is larger when the difference between the previous and current delay times is prominent, e.g., when the traffic scenario on the link changes from under-saturated to saturated or over-saturated in between two consecutive time steps).

To solve the first issue, we propose an additional network element for the S-model called the source link, which is a link that has one of the source nodes of the network as its downstream node (see Figure 4), and that feeds the network with new demand at every time step. Moreover, we develop a general formulation for discrete-time systems with time-varying delays, which produces accurate results compared with the results produced by the continuous-time formulation. We later use this discrete-time representation to compute $\alpha^{\text{arrive,q}}_{u,d,o}(k_d)$.

1) Formulating the source queues for the S-model: Suppose that $\mathcal{S}$ is the set of all source nodes of the network, s is

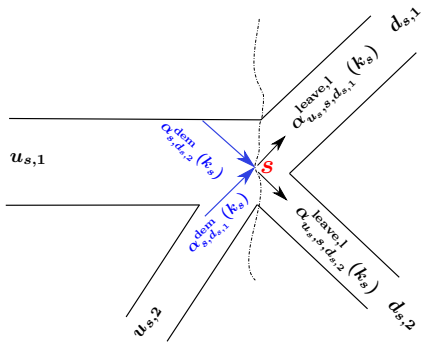

Figure 4. Two source links $(u_{s,1}, s)$ and $(u_{s,2}, s)$ that feed links $(s, d_{s,1})$ and $(s, d_{s,2})$ of the network via source node s.

a source node, $\mathcal{U}(s)$ and $\mathcal{D}(s)$ are the sets of all upstream nodes of s outside of the network and all downstream nodes of s inside the network, $u_{s,i} \in \mathcal{U}(s),\ i \in \{1, \dots, \text{card}\,(\mathcal{U}(s))\}$ denotes the upstream nodes of source node s outside of the network, and $d_{s,j} \in \mathcal{D}(s),\ j \in \{1, \dots, \text{card}\,(\mathcal{D}(s))\}$ denotes the downstream nodes of source node s inside of the network. A queue will be formed within a source link $(u_{s,i}, s)$, if there is not enough free space for the given demand of link $(u_{s,i}, s)$ on those links of the network that are connected to this source link via node s (i.e., links $(s, d_{s,j}),\ j \in \{1, \dots, \text{card}\,(\mathcal{D}(s))\}$). We define $q_s^{\text{source}}(k_s)$, called the source queue, as the overall queue length in [veh/s] at source node s at time step k_s. Thus the source queue q_s^{source} is the summation of all queue lengths for the source links for node s. We then have

$$q_s^{\text{source}}(k_s+1) = q_s^{\text{source}}(k_s) + \left(\alpha_s^{\text{dem}}(k_s) - \alpha_s^{\text{leave,l}}(k_s)\right) c_s, \tag{22}$$

where $\alpha_s^{\text{dem}}(k_s)$ is the cumulative demand flow in [veh/s] at source node s at time step k_s, given by

$$\alpha_s^{\text{dem}}(k_s) = \sum\nolimits_{j=1}^{\text{card}(\mathcal{D}(s))} \alpha_{s,d_{s,j}}^{\text{dem}}(k_s), \tag{23}$$

and $\alpha_s^{\text{leave,l}}(k_s)$ is the cumulative leaving flow in [veh/s] at source node s at time step k_s (i.e., the total flow of vehicles that can enter the network via source node s during $[k_s c_s, (k_s+1)c_s)$). We have

$$\alpha_s^{\text{leave,l}}(k_s) = \min\left\{\mu_s, \alpha_s^{\text{dem}}(k_s) + \frac{q_s^{\text{source}}(k_{d_s})}{c_s}, \sum\nolimits_{j=1}^{\text{card}(\mathcal{D}(s))} \frac{C_{s,d_{s,j}} - n_{s,d_{s,j}}(k_{d_s})}{c_s}\right\}, \tag{24}$$

with $\mu_s = \sum_{i=1}^{\text{card}(\mathcal{U}(s))} \mu_{u_{s,i},s}$ the cumulative saturated leaving flow rate at source node s.

2) Extended formulation of the time-delayed equation for arriving flow: To compute $\alpha_{u,d}^{\text{arrive,q}}(k_d)$, we should consider the time the entering vehicles in link (u,d) need to reach the tail of the waiting queue, i.e., the delay time $\tau_{u,d}(k_d)$, and the entering rate of the vehicles in the link $\tau_{u,d}(k_d)$ time units ago. So, for the queue length $q_{u,d,o}(k_d)$ we actually have a time-delayed differential equation (in the discrete-time domain). Using the results of Section III, $\tilde{x}(k)$ will be substituted by $q_{u,d,o}(k_d)$, and the factors $\tilde{A}$ and $\tilde{C}$ by 1 and -1. We obtain

$$q_{u,d,o}(k_d+1) = q_{u,d,o}(k_d) + \beta_{u,d,o}(k_d)\cdot \sum_{i=0}^{\delta_{u,d}(k_d-1)-\delta_{u,d}(k_d)+1} \tilde{B}_{u,d,i}(k_d)\alpha_{u,d}^{\text{enter,l}}(k_d - \delta_{u,d}(k_d-1) - 2 + i) - \alpha_{u,d,o}^{\text{leave,l}}(k_d)c_d, \tag{25}$$

where $\tilde{B}_{u,d,0}(k_d) = \dfrac{\gamma_{u,d}(k_d-1)}{c_d}$, $\tilde{B}_{u,d,i}(k_d) = 1$ for $i \in \{1, \dots, \delta_{u,d}(k_d - 1) - \delta_{u,d}(k_d)\}$, $\tilde{B}_{u,d,\delta_{u,d}(k_d-1)-\delta_{u,d}(k_d)+1}(k_d) = \dfrac{c_d - \gamma_{u,d}(k_d)}{c_d}$, and $\delta_{u,d}(k_d) = \left\lfloor \dfrac{\tau_{u,d}(k_d)}{c_d} \right\rfloor$, $\gamma_{u,d}(k_d) = \text{rem}\,\{\tau_{u,d}(k_d), c_d\}$. Comparing (25) with (13) and (18), we conclude that

$$\alpha_{u,d}^{\text{arrive,q}}(k_d) = \sum_{i=0}^{\delta_{u,d}(k_d-1)-\delta_{u,d}(k_d)+1} \tilde{B}_{u,d,i}\alpha_{u,d}^{\text{enter,l}}(k_d - \delta_{u,d}(k_d-1) - 2 + i). \tag{26}$$

Moreover, the delay time $\tau_{u,d}(k_d)$ at time step k_d (assumed to be constant within $[k_d c_d, (k_d+1)c_d)$) is computed as follows. Vehicles are assumed to enter the link with $v_{u,d}^{\text{free}}$. If there is an idling queue in front of them, after a while they will decelerate[1] with $a_{u,d}^{\text{dec}}$ to reach $v_{u,d}^{\text{idle}}$. During this period, the distance between the upstream intersection and the tail of the waiting queue will be traveled by these vehicles (note that in the original S-model [17], it is assumed that this distance is traveled with a constant speed $v_{u,d}^{\text{free}}$). Therefore,

$$\frac{\left(C_{u,d} - q_{u,d}^{\text{ave}}(k_d)\right) l^{\text{veh}}}{N_{u,d}^{\text{lane}}} = v_{u,d}^{\text{free}} \cdot T_{u,d}^{\text{free}}(k_d) + \frac{1}{2}a_{u,d}^{\text{dec}}\left(\frac{v_{u,d}^{\text{idle}} - v_{u,d}^{\text{free}}}{a_{u,d}^{\text{dec}}}\right)^2 + v_{u,d}^{\text{free}}\left(\frac{v_{u,d}^{\text{idle}} - v_{u,d}^{\text{free}}}{a_{u,d}^{\text{dec}}}\right), \tag{27}$$

where $C_{u,d}$ is the storage capacity of link (u,d), $q_{u,d}^{\text{ave}}(k_d)$ is the average queue length on link (u,d) within $[k_d c_d, (k_d+1)c_d)$, l^{veh} is the average length of the vehicles, $N_{u,d}^{\text{lane}}$ is the number of lanes in link (u,d), $v_{u,d}^{\text{free}}$ and $v_{u,d}^{\text{idle}}$ are the free-flow and idling speed on link (u,d), and $a_{u,d}^{\text{dec}}$ is the deceleration. Note that $\dfrac{v_{u,d}^{\text{idle}} - v_{u,d}^{\text{free}}}{a_{u,d}^{\text{dec}}}$ is the time needed for vehicles to reach $v_{u,d}^{\text{idle}}$ from $v_{u,d}^{\text{free}}$ by the constant deceleration $a_{u,d}^{\text{dec}}$. Then the delay time $\tau_{u,d}(k_d)$ of the vehicles is obtained by

$$\tau_{u,d}(k_d) = T_{u,d}^{\text{free}}(k_d) + \frac{v_{u,d}^{\text{idle}} - v_{u,d}^{\text{free}}}{a_{u,d}^{\text{dec}}}. \tag{28}$$

[1]The typical values of the speeds and acceleration can be determined via identifying the S-model's parameters w.r.t. real-life data or data from a traffic microsimulator. The difference between the output produced by the model and the data is then minimized by solving an optimization problem offline [29].

where the value of $T_{u,d}^{\text{free}}(k_d)$ is computed from (27). Hence, the delay time is

$$\tau_{u,d}(k_d) = \frac{\left(C_{u,d} - q_{u,d}^{\text{ave}}(k_d)\right) l^{\text{veh}}}{N_{u,d}^{\text{lane}} v_{u,d}^{\text{free}}} - \frac{\left(v_{u,d}^{\text{idle}} - v_{u,d}^{\text{free}}\right)^2}{2a_{u,d}^{\text{dec}} v_{u,d}^{\text{free}}}. \quad (29)$$

In order to compute the average queue length $q_{u,d}^{\text{ave}}(k_d)$, we propose to substitute $q_{u,d}^{\text{ave}}(k_d)$ with the average of the queue lengths at k_d and at $k_d + 1$, where, to estimate $q_{u,d}(k_d + 1)$ at k_d, we can use extrapolation, i.e.,

$$q_{u,d}(k_d + 1) = q_{u,d}(k_d) + q_{u,d}(k_d) - q_{u,d}(k_d - 1).$$

Then we have

$$q_{u,d}^{\text{ave}}(k_d) = \frac{3}{2} q_{u,d}(k_d) - \frac{1}{2} q_{u,d}(k_d - 1). \quad (30)$$

V. Integrated flow and emission model for urban traffic networks

In this section, we briefly discuss a general mesoscopic framework for integrating macroscopic urban traffic flow models with microscopic emission models. The resulting integrated flow and emission model can be used with an MPC controller to reduce congestion and emissions in urban traffic networks. The mesoscopic framework has been introduced by Jamshidnejad et al. in [30], and considers three different urban traffic scenarios: under-saturated, saturated, and over-saturated. For each scenario, groups of vehicles with a similar traffic behavior are distinguished, where the general behavior of the group is represented by a single time-speed representative curve. Next, we present the proposed framework for the under-saturated scenario. For equations and details regarding the other two scenarios, see [30].

A. Under-saturated traffic

The under-saturated traffic scenario occurs when the demand on a link is less than the number of vehicles that can be discharged by the saturated average leaving flow rate within one cycle. The time-speed curves for different traffic behaviors in under-saturated traffic are illustrated in Figure 5. Correspondingly, the vehicles can be divided into four groups:

Group 1. Composed of those vehicles that are already in a queue on the link at the beginning of the current cycle (see the solid curve in Figure 5). At t_{green}, when the traffic light turns green, these vehicles start to accelerate and leave the link by the saturated leaving flow rate $\mu_{u,d}$.

Group 2. Composed of those vehicles that arrive at the tail of the queue during the current cycle (see the dashed curve in Figure 5). These vehicles decelerate from $v_{u,d}^{\text{free}}$ to reach $v_{u,d}^{\text{idle}}$ as they approach the tail of the queue, and idle for a while. When the traffic light turns green, each vehicle waits for the vehicles in front to accelerate. Then it also accelerates until its speed reaches $v_{u,d}^{\text{free}}$. The leaving flow rate for the vehicles in group 2 is also $\mu_{u,d}$.

Group 3. Composed of those vehicles that arrive at the tail of the queue (which is formed by the vehicles in group 1 and group 2, and the pioneers of its own group), when this queue has already started to move forward. The dashed-dotted curve in Figure 5 illustrates the average behavior for the representative vehicle of group 3. These vehicles do not need to decrease their speed to $v_{u,d}^{\text{idle}}$, since when they approach the tail of the queue, it has a higher speed $v_{u,d}^{\text{middle}}$ than the idling speed $v_{u,d}^{\text{idle}}$, where $v_{u,d}^{\text{middle}} = 0.5\left(v_{u,d}^{\text{free}} + v_{u,d}^{\text{idle}}\right)$. Therefore, these vehicles will only decelerate until they also reach $v_{u,d}^{\text{middle}}$. As soon as these vehicles are at the end of the moving queue, they mimic the time-speed curve of the front queue and move forward with it, i.e., they accelerate until their speed reaches $v_{u,d}^{\text{free}}$, and after a while they leave the link.

Group 4. Composed of those vehicles that enter the link when they do not need to decelerate at all, because there is enough space between them and the groups of vehicles that move in front of them. Therefore, the vehicles in group 4 move forward with $v_{u,d}^{\text{free}}$ until they leave the link. The average behavior of the vehicles in group 4 is illustrated by the dotted curve in Figure 5.

We first determine the number of vehicles in each of the groups. From the definitions given above, we can write

$$n_{u,d}^{\text{G}_1}(k_d) = q_{u,d}(k_d), \quad (31)$$

$$n_{u,d}^{\text{G}_2}(k_d) = \alpha_{u,d}^{\text{arrive,q}}(k_d) \cdot T_{u,d}^{\text{arrive,G}_2}(k_d), \quad (32)$$

$$n_{u,d}^{\text{G}_3}(k_d) = \alpha_{u,d}^{\text{arrive,q}}(k_d) \cdot T_{u,d}^{\text{arrive,G}_3}(k_d), \quad (33)$$

$$n_{u,d}^{\text{G}_4}(k_d) = n_{u,d}(k_d) - \textstyle\sum_{i=1}^{3} n_{u,d}^{\text{G}_i}(k_d), \quad (34)$$

where

$$T_{u,d}^{\text{arrive,G}_2}(k_d) = \mu_{u,d} \Big/ \left(\mu_{u,d} - \alpha_{u,d}^{\text{arrive,q}}(k_d)\right) \cdot \left(c_d - g_{u,d}(k_d) + n_{u,d}^{\text{G}_1}(k_d) \Big/ \mu_{u,d} - \tau_{u,d}^{\text{G}_2}(k_d)\right), \quad (35)$$

$$T_{u,d}^{\text{arrive,G}_3}(k_d) = \tau_{u,d}^{\text{G}_2}(k_d) + \left(v_{u,d}^{\text{free}} - v_{u,d}^{\text{idle}}\right) / a_{u,d}^{\text{acc}}, \quad (36)$$

with $T_{u,d}^{\text{arrive,G}_2}(k_d)$ and $T_{u,d}^{\text{arrive,G}_3}(k_d)$ the overall arrival time for all vehicles in group 2 and in group 3 (where the overall arrival time is the temporal distance between the starting points of the time-speed curves of the first and the last vehicle in a group), respectively, and $\tau_{u,d}^{\text{G}_2}(k_d)$ the average delay time for the vehicles in group 2, i.e., the average time a vehicle in group 2 needs to reach the tail of the waiting vehicles. This delay time is computed by (29) for $q_{u,d}^{\text{G}_2,\text{ave}}(k_d) = n_{u,d}^{\text{G}_1}(k_d)$.

We next explain how (35) and (36) are obtained, considering the time-speed curves of the first and the last vehicle of group 2 and group 3. The difference between the time spent by these two vehicles during the current cycle depends on the difference between the arriving flow rate $\alpha_{u,d}^{\text{arrive,q}}(k_d)$ and the leaving flow rate $\mu_{u,d}$ for the vehicles in the group. The reason is that the temporal distance between the starting points and the endpoints of the time-speed curves of the vehicles is determined by these flow rates. For the vehicles in group 2, the temporal distance between the starting points of the two consecutive time-speed curves equals $\left(\alpha_{u,d}^{\text{arrive,q}}(k_d)\right)^{-1}$, while it is less, i.e., $(\mu_{u,d})^{-1}$ for the endpoints of these curves (note that $\mu_{u,d} > \alpha_{u,d}^{\text{arrive,q}}(k_d)$). Since the number of vehicles in group 2 is $\alpha_{u,d}^{\text{arrive,q}}(k_d) T_{u,d}^{\text{arrive,G}_2}(k_d)$, the overall temporal distance between the starting points of the time-speed curves of the first and the last vehicle of group 2

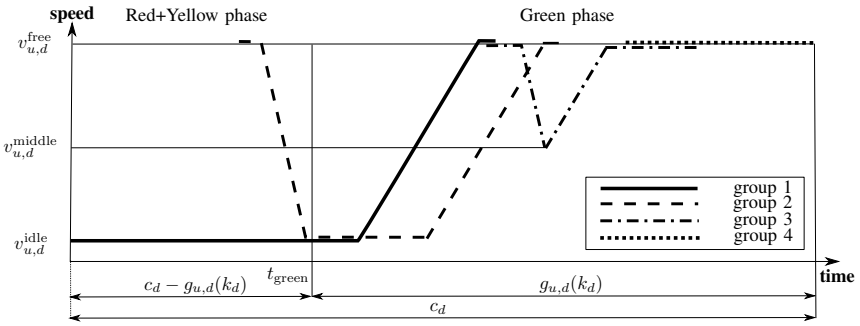

Figure 5. Traffic behaviors on link (u, d) within one cycle in an **under-saturated** traffic scenario for urban traffic networks.

is $\left(\alpha_{u,d}^{\text{arrive,q}}(k_d)\right)^{-1} \alpha_{u,d}^{\text{arrive,q}}(k_d) T_{u,d}^{\text{arrive,G}_2}(k_d)$, while the overall temporal distance between the endpoints of their time-speed curves is $(\mu_{u,d}(k_d))^{-1} \alpha_{u,d}^{\text{arrive,q}}(k_d) T_{u,d}^{\text{arrive,G}_2}(k_d)$. Hence, the difference between the time spent by the first and the last vehicle of group 2 on the link during the current cycle is $\frac{\mu_{u,d} - \alpha_{u,d}^{\text{arrive,q}}(k_d)}{\mu_{u,d}} T_{u,d}^{\text{arrive,G}_2}(k_d)$. Moreover, the time needed for all vehicles in group 2 to leave the network after they start to accelerate is almost the same assuming that they all travel the same distance, while the time they spend before accelerating might be different (due to different idling times for different vehicles). This time for the first vehicle of group 2 includes the duration of the red+yellow phase, i.e., $c_d - g_{u,d}(k_d)$, in addition to the time $n_{u,d}^{\text{G}_1}(k_d)/\mu_{u,d}$ the first vehicle of group 2 should wait for the vehicles in front of it to accelerate. The time spent by the last vehicle of group 2 before it accelerates is the time it needs to reach the tail of the front queue, i.e., the average delay time $\tau_{u,d}^{\text{G}_2}(k_d)$ of the vehicles in group 2. Finally, from the explanations given, we can obtain (35) (for more details see [30]).

For (36), since in the limit, the last vehicle of group 3 will behave similarly to the first vehicle of group 4 (purely free-flow behavior), we can assume that the starting point of the time-speed curve of the last vehicle in group 3 almost coincides with the starting point of the free-flow behavior of the first vehicle in this group (see [30] for extra details). Hence, the overall arrival time of the vehicles in group 3 $T_{u,d}^{\text{arrive,G}_3}(k_d)$, i.e., the temporal distance between the starting points of the time-speed curves of the first and the last vehicle of group 3, equals the time spent by the first vehicle of group 3 before it reaches $v_{u,d}^{\text{free}}$ at the end of its travel on the link. The first vehicle of group 3 enters the link with $v_{u,d}^{\text{free}}$ and approaches the tail of the queue formed by the vehicles in groups 1 and 2 when this queue has just started to accelerate and move forward. Hence, the first vehicle of group 3 decelerates to the current speed of the queue, i.e., $v_{u,d}^{\text{idle}}$. The time needed by this vehicle to reach the tail of the moving queue is, in the limit, the same as the average delay time $\tau_{u,d}^{\text{G}_2}(k_d)$ of the vehicles in group 2 (see the first term of (36)). Then the first vehicle of group 3 accelerates to move forward with the front queue and reach $v_{u,d}^{\text{free}}$ before it leaves the link (see the second term of (36)).

The different traffic behaviors can be divided into four types: free-flow, idling, decelerating, and accelerating. A microscopic emission model, such as VT-micro [31], can compute the instantaneous emissions $e(v, a)$ for a given vehicle from the acceleration a and speed v of the vehicle. Suppose that $E_{u,d}$ denotes the total emissions on link (u, d) in an under-saturated traffic case. Then for each of the behaviors mentioned, we can write

$$E_{u,d}^{\text{free}} = \sum_i \left(n_{u,d}^{\text{free},i}(k_d) T_{u,d}^{\text{free},i}(k_d) \right) e(v_{u,d}^{\text{free}}, 0), \tag{37}$$

$$E_{u,d}^{\text{idle}} = \sum_i \left(n_{u,d}^{\text{idle},i}(k_d) T_{u,d}^{\text{idle},i}(k_d) \right) e(v_{u,d}^{\text{idle}}, 0), \tag{38}$$

$$E_{u,d}^{\text{dec}} = \sum_i \left(\frac{n_{u,d}^{\text{dec},i}(k_d)}{a_{u,d}^{\text{dec}}} \int_{v_{u,d}^{\text{h},i,b}}^{v_{u,d}^{\text{l},i,b}} e\left(v, a_{u,d}^{\text{dec}}\right) \cdot \mathrm{d}v \right), \tag{39}$$

$$E_{u,d}^{\text{acc}} = \sum_i \left(\frac{n_{u,d}^{\text{acc},i}(k_d)}{a_{u,d}^{\text{acc}}} \int_{v_{u,d}^{\text{l},i,b}}^{v_{u,d}^{\text{h},i,b}} e\left(v, a_{u,d}^{\text{acc}}\right) \cdot \mathrm{d}v \right), \tag{40}$$

for $i \in \{\text{G}_1, \text{G}_2, \text{G}_3, \text{G}_4\}$, and where the superscripts l and h indicate the lower and higher speeds for different traffic behaviors b. From the given discussions for different groups of vehicles, in the under-saturated case, the lower speed may be $v_{u,d}^{\text{idle}}$ (see group 1 and group 2 in Figure 5) or $v_{u,d}^{\text{middle}}$ (see group 3 in Figure 5) and the higher speed is $v_{u,d}^{\text{free}}$ (see groups 1, 2, and 3 in Figure 5). Next we determine the number of vehicles from each group in the under-saturated case that participate in a specific traffic behavior, and also the total time spent by the representative vehicle for the free-flow and the idling behavior.

From Figure 5, all groups show the free-flow behavior during the current cycle. Hence,

$$\begin{aligned} n_{u,d}^{\text{free,G}_1}(k_d) &= n_{u,d}^{\text{G}_1}(k_d), \\ n_{u,d}^{\text{free,G}_2}(k_d) &= n_{u,d}^{\text{G}_2}(k_d), \\ n_{u,d}^{\text{free,G}_3}(k_d) &= n_{u,d}^{\text{G}_3}(k_d), \\ n_{u,d}^{\text{free,G}_4}(k_d) &= n_{u,d}^{\text{G}_4}(k_d), \end{aligned} \tag{41}$$

where the average time that each group spends on the free-flow

behavior is obtained by

$$
\begin{aligned}
T_{u,d}^{\mathrm{free,G_1}}(k_d) &= \frac{0.5n_{u,d}^{\mathrm{G_1}}(k_d)}{N_{u,d}^{\mathrm{lane}}\cdot v_{u,d}^{\mathrm{free}}}l^{\mathrm{veh}} - \frac{\left(v_{u,d}^{\mathrm{free}}\right)^2-\left(v_{u,d}^{\mathrm{idle}}\right)^2}{2a_{u,d}^{\mathrm{acc}}\cdot v_{u,d}^{\mathrm{free}}},\\
T_{u,d}^{\mathrm{free,G_2}}(k_d) &= \tau_{u,d}^{\mathrm{G_2}}(k_d) - \frac{v_{u,d}^{\mathrm{idle}}-v_{u,d}^{\mathrm{free}}}{a_{u,d}^{\mathrm{dec}}}\\
&\quad + \frac{n_{u,d}^{\mathrm{G_1}}(k_d)+0.5n_{u,d}^{\mathrm{G_2}}(k_d)}{N_{u,d}^{\mathrm{lane}}\cdot v_{u,d}^{\mathrm{free}}}l^{\mathrm{veh}} - \frac{\left(v_{u,d}^{\mathrm{free}}\right)^2-\left(v_{u,d}^{\mathrm{idle}}\right)^2}{2a_{u,d}^{\mathrm{acc}}\cdot v_{u,d}^{\mathrm{free}}},\\
T_{u,d}^{\mathrm{free,G_3}}(k_d) &= C_{u,d}l^{\mathrm{veh}}\Big/N_{u,d}^{\mathrm{lane}}v_{u,a}^{\mathrm{free}}+\\
&\qquad 0.5\left(v_{u,d}^{\mathrm{free}}-v_{u,d}^{\mathrm{idle}}\right)\left(1/a_{u,d}^{\mathrm{dec}}-1/a_{u,d}^{\mathrm{dec}}\right),\\
T_{u,d}^{\mathrm{free,G_4}}(k_d) &= n_{u,d}^{\mathrm{G_4}}(k_d)\Big/\alpha_{u,d}^{\mathrm{arrive,q}}(k_d).
\end{aligned}
\tag{42}
$$

Now we prove and motivate the equations above. For the vehicles in group 1, the free-flow behavior is observed at the end of the time-speed curve. The average distance traveled by the representative vehicle of group 1 after the traffic light turns green is $0.5n_{u,d}^{\mathrm{G_1}}(k_d)l^{\mathrm{veh}}/N_{u,d}^{\mathrm{lane}}$. This is because for the representative vehicle we can assume that the total distance that should be traveled to the end of the link is half of the length of the queue formed by the vehicles in group 1, i.e., half of the length $n_{u,d}^{\mathrm{G_1}}(k_d)l^{\mathrm{veh}}/N_{u,d}^{\mathrm{lane}}$, assuming that the vehicles are equally distributed along the $N_{u,d}^{\mathrm{lane}}$ lanes of the link. Knowing that the representative vehicle should travel this distance partly by the constant acceleration $a_{u,d}^{\mathrm{acc}}$ and partly by the constant speed $v_{u,d}^{\mathrm{free}}$, from the kinematics knowledge[2], we can easily obtain the first equation of (42). A similar reasoning holds for the free-flow behavior of the representative vehicle of group 2 at the end of its trip on the link, i.e., we assume the distance traveled by the representative vehicle of group 2 is the average of the distances traveled by the first and the last vehicle of group 2 (i.e., the average of $n_{u,d}^{\mathrm{G_1}}(k_d)l^{\mathrm{veh}}/N_{u,d}^{\mathrm{lane}}$ and $\left(n_{u,d}^{\mathrm{G_1}}(k_d)+n_{u,d}^{\mathrm{G_2}}(k_d)\right)l^{\mathrm{veh}}/N_{u,d}^{\mathrm{lane}}$). The last two terms of the second equation of (42) are obtained this way. Additionally, free-flow behavior is also observed at the beginning of the trip of the representative vehicle of group 2 (note that the distance between the entrance of the link and the tail of the waiting queue is traveled partly by the constant speed $v_{u,d}^{\mathrm{free}}$, and partly by the constant acceleration $a_{u,d}^{\mathrm{acc}}$). The overall time for the free-flow and the accelerating behaviors at the beginning of the time-speed curve of the representative vehicle the average delay time of the vehicles in group 2. Hence, the first two terms of (42) are obtained (see [30] for more details).

The idling behavior is observed for the vehicles in group 1 and group 2, while the vehicles in the other groups do not show any idling behavior. Thus,

$$
\begin{aligned}
n_{u,d}^{\mathrm{idle,G_1}}(k_d) &= n_{u,d}^{\mathrm{G_1}}(k_d),\\
n_{u,d}^{\mathrm{idle,G_2}}(k_d) &= n_{u,d}^{\mathrm{G_2}}(k_d),\\
n_{u,d}^{\mathrm{idle,G_3}}(k_d) &= 0,\\
n_{u,d}^{\mathrm{idle,G_4}}(k_d) &= 0,
\end{aligned}
\tag{43}
$$

[2]The displacement Δd of a vehicle that starts its motion with speed v_0 and moves with the constant acceleration a, during a time period T is obtained via $\Delta d = \frac{1}{2}aT^2+v_0T$, while the displacement of the vehicle when it moves with the constant speed v is $\Delta d = vT$.

where the time spent by the vehicles in each group for the idling behavior is

$$
\begin{aligned}
T_{u,d}^{\mathrm{idle,G_1}}(k_d) &= c_d - g_{u,d}(k_d) + \left(0.5n_{u,d}^{\mathrm{G_1}}(k_d)\right)\Big/\mu_{u,d},\\
T_{u,d}^{\mathrm{idle,G_2}}(k_d) &= T_{u,d}^{\mathrm{idle,G_1}}(k_d) - \tau_{u,d}^{\mathrm{G_2}}(k_d),\\
T_{u,d}^{\mathrm{idle,G_3}}(k_d) &= 0,\\
T_{u,d}^{\mathrm{idle,G_4}}(k_d) &= 0.
\end{aligned}
\tag{44}
$$

Note that the first vehicle of group 1 idles during the red+yellow phase, i.e., for $c_d - g_{u,d}(k_d)$ time units, while the last vehicle of the group should first wait for the rest of the vehicles in the group to leave the link. Hence, the idling time of the last vehicle of group 1 is $n_{u,d}^{\mathrm{G_1}}(k_d)/\mu_{u,d}$ time units larger than that of the first vehicle of the group. The average of the idling times of the first and the last vehicle is the idling time of the representative vehicle of this group (see the first equation of (44)). The idling time of the representative vehicle of group 2 is similar to that of the representative vehicle of group 1, except that both the first and the last vehicle idle for $\tau_{u,d}^{\mathrm{G_2}}(k_d)$ time units less than the first and the last vehicle of group 1. This is because the first and the last vehicle of group 2 enter the link with $v_{u,d}^{\mathrm{free}}$, and hence they should decelerate before they start their idling behavior. This explains how the second equation of (44)) is obtained.

The decelerating behavior occurs for the vehicles in group 2 and group 3, i.e.,

$$
\begin{aligned}
n_{u,d}^{\mathrm{dec,G_1}}(k_d) &= 0,\\
n_{u,d}^{\mathrm{dec,G_2}}(k_d) &= n_{u,d}^{\mathrm{G_2}}(k_d),\\
n_{u,d}^{\mathrm{dec,G_3}}(k_d) &= n_{u,d}^{\mathrm{G_3}}(k_d),\\
n_{u,d}^{\mathrm{dec,G_4}}(k_d) &= 0.
\end{aligned}
\tag{45}
$$

Finally, for the accelerating behavior, which is observed for the first, second, and third group, we have

$$
\begin{aligned}
n_{u,d}^{\mathrm{acc,G_1}}(k_d) &= n_{u,d}^{\mathrm{G_2}}(k_d),\\
n_{u,d}^{\mathrm{acc,G_2}}(k_d) &= n_{u,d}^{\mathrm{G_2}}(k_d),\\
n_{u,d}^{\mathrm{acc,G_3}}(k_d) &= n_{u,d}^{\mathrm{G_3}}(k_d),\\
n_{u,d}^{\mathrm{acc,G_4}}(k_d) &= 0.
\end{aligned}
\tag{46}
$$

VI. Model-predictive control for urban traffic networks

Model-predictive control (MPC) [32] is an optimization-based control approach that leads the controlled system to operate close to a defined optimal performance. In MPC, a performance index is minimized over a finite-length prediction window spanning N_{p} control time steps. Next, only a part of the obtained suboptimal control trajectory between the current and the next control time step is applied to the system. The starting point of the prediction horizon is then shifted and the optimization problem is solved using the updated state measurements. Compared to the optimal control strategies with an entirely open-loop scheme, where the optimization problem is solved offline over the entire simulation period, the suboptimal solution of the MPC is more robust towards unexpected disturbances [33].

Taking into account the positive characteristics of an MPC controller (i.e., it performs a feedback-based and to some extent robust control approach), which fit well the requirements that are usually expected for management of traffic networks with highly dynamic behaviors, we also consider the MPC approach for urban traffic control. The aim of the controller is to find a balanced trade-off between prevention/reduction of congestion and reduction of emissions. Therefore, for the prediction model of the MPC controller, which estimates the future states of the system along the prediction window, the integrated flow and emission framework introduced in Section V including the extended S-model (see Section IV-B) and VT-micro [34] is used.

A. Formulation of the optimization problem

$$\min_{\tilde{\boldsymbol{u}}(k_\mathrm{c})} \mathcal{J}(k_\mathrm{c}) = \min_{\boldsymbol{u}(k_\mathrm{c})} \left(\mathcal{J}^\mathrm{t}(k_\mathrm{c}) + \mathcal{J}^{\mathrm{s,cum}}(k_\mathrm{c})\right),$$

s.t.

(49) and (50),
Integrated flow and emission model (12)-(46) [30],
$\mathcal{U}(\tilde{\boldsymbol{u}}(k_\mathrm{c})) = 0$ (e.g., see (48)),
$\boldsymbol{u}_{\min} \leq \tilde{\boldsymbol{u}}(k_\mathrm{c}) \leq \boldsymbol{u}_{\max}$,

(47)

where $\mathcal{J}(k_\mathrm{c})$ denotes the summation of the terminal objective $\mathcal{J}^\mathrm{t}(k_\mathrm{c})$ and the cumulative stage objective function $\mathcal{J}^{\mathrm{s,cum}}(k_\mathrm{c})$ that is computed within one prediction window starting at control time step k_c, i.e., within $[k_\mathrm{c}T_\mathrm{c}, (k_\mathrm{c}+N_\mathrm{p})\,T_\mathrm{c})$ with T_c the control sampling cycle. Furthermore, the optimization variable $\tilde{\boldsymbol{u}}(k_\mathrm{c})$ is a vector that includes $\boldsymbol{u}^\top(k_\mathrm{c}), \ldots, \boldsymbol{u}^\top(k_\mathrm{c}+N_\mathrm{p}-1)$, $\boldsymbol{u}(k_\mathrm{c})$ is a vector that includes all control inputs of the system (the green time lengths for an urban traffic network along the prediction window) at control time step k_c, $\boldsymbol{u}_{\min}$ and $\boldsymbol{u}_{\max}$ are vectors of the same size as $\boldsymbol{u}(k_\mathrm{c})$ that include element-wise the minimum and maximum allowed values for the control inputs within $\boldsymbol{u}(k_\mathrm{c})$, and $\mathcal{U}(\boldsymbol{u}(k_\mathrm{c})) = 0$ indicates the equality constraints on the control vector. For example, $\mathcal{U}(\boldsymbol{u}(k_\mathrm{c})) = 0$ may indicate that the summation of the green and yellow times for each intersection equals the cycle time for that intersection. More specifically, suppose that $u_{d,i}(l)$ for all $d \in \mathcal{I}^{\mathrm{ctrl}}$ (with $\mathcal{I}^{\mathrm{ctrl}}$ the set of all intersections of the urban traffic network that are controlled by traffic signals) and $i \in \{1, \ldots, n_d^{\mathrm{green}}\}$ (with n_d^{green} the number of green signals[3] of intersection d) indicates the i^{th} traffic signal at intersection d. Then for all $d \in \mathcal{I}^{\mathrm{ctrl}}$, $\mathcal{U}(\boldsymbol{u}(k_\mathrm{c})) = 0$ is a relationship of the form

$$\sum_{i=1}^{n_d^{\mathrm{green}}} u_{d,i}(l) = c_d - y_d,\ l \in \{k_\mathrm{c}, \ldots, k_\mathrm{c}+N_\mathrm{p}-1\}, \quad (48)$$

where y_d indicates the total yellow or all-red time within one cycle of the intersection corresponding to node d.

The main aim of the control system is to find a balanced trade-off between reducing the congestion level, reducing the total emissions, and preventing high fluctuations in time for the control signals. Hence, the stage objective function of the MPC controller is formulated as a weighted combination of the total time spent (which quantifies the congestion level), the total emissions, and the absolute difference of two temporally successive control vectors. We have

$$\mathcal{J}^\mathrm{s}(k) = w_T \frac{T(k)}{T^\mathrm{t}} + \sum_{p\in\mathbb{P}} w_{E_p} \frac{E_p(k)}{E_p^\mathrm{t}} + w_v \frac{\mathcal{V}(\boldsymbol{u}(k))}{\mathcal{V}^\mathrm{n}}, \quad (49)$$

with $T(k)$ and $E_p(k)$ the total time spent and emissions of $p \in \mathbb{P}$ (where $\mathbb{P}$ is a set of pollutants, e.g., $\mathbb{P} = \{\mathrm{CO},\ \mathrm{HC},\ \mathrm{NO_x}\}$) within one control sampling time, T^t and E_p^t typical values of $T(\cdot)$ and $E_p(\cdot)$ within one prediction window, $\mathcal{V}(\boldsymbol{u}(k)) = \|\boldsymbol{u}(k) - \boldsymbol{u}(k-1)\|$ for some norm function $\|\cdot\|$, and $\mathcal{V}^\mathrm{n}$ a nominal value for $\mathcal{V}(\boldsymbol{u}(k))$ in one control sampling time that may be computed by $\|\boldsymbol{u}_{\max} - \boldsymbol{u}_{\min}\|$.

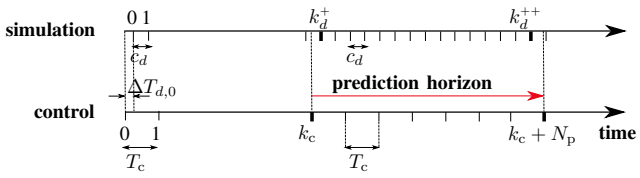

Figure 6. Simulation and control time step counters.

B. Computation of the objective function

In (49), $T(\cdot)$ and $E_p(\cdot)$ are the summation of the total time spent and total emissions for all links in the traffic network. In order to compute these quantities on each link, we should first determine the ongoing traffic state (see Section V and [30]). For the total time spent by the vehicles on link (u,d) at time step k_c, we have

$$\begin{aligned} T_{u,d}(k_\mathrm{c}) = {} & \textstyle\sum_{i=1}^{N_\mathrm{G}^s} \sum_{b\in\mathbb{B}} \Big(n_{u,d}^{b,i,s} \left(k_d^+(k_\mathrm{c}) - 1\right) \cdot \\ & \min\left\{ T_{u,d}^{b,i,s} \left(k_d^+(k_\mathrm{c}) - 1\right), k_d^+(k_\mathrm{c}) c_d - k_\mathrm{c} T_\mathrm{c} \right\} \Big) \\ & + \textstyle\sum_{j=k_d^+(k_\mathrm{c})}^{k_d^{++}(k_\mathrm{c})} \sum_{i=1}^{N_\mathrm{G}^s} \sum_{b\in\mathbb{B}} n_{u,d}^{b,i,s}(j) T_{u,d}^{b,i,s}(j) + \\ & \textstyle\sum_{i=1}^{N_\mathrm{G}^s} \sum_{b\in\mathbb{B}} \Big(n_{u,d}^{b,\mathrm{G}_i,s} \left(k_d^{++}(k_\mathrm{c}) + 1\right) \cdot \\ & \min\Big\{ T_{u,d}^{b,i,s} \left(k_d^{++}(k_\mathrm{c}) + 1\right), \\ & \qquad (k_\mathrm{c}+N_\mathrm{p}) T_\mathrm{c} - k_d^{++}(k_\mathrm{c}) c_d \Big\} \Big), \end{aligned} \quad (50)$$

where s adopts under-saturated, saturated, and over-saturated based on the ongoing traffic scenario on the link, $\mathbb{B}$={free, idling, dec, acc}, N_G^s shows the number of different groups for the ongoing traffic state, and $T_{u,d}^{b,\mathrm{G}_i,s}(k_d)$ is computed by the equations given in Sections V-A. The total emissions E_p of p for each link of the network can be computed via (37)-(40).

In general, the system's sampling time and the control sampling time might not be equal and therefore, the time instants k_d (i.e., the simulation time step counter for the link (u,d)) and k_c (i.e., the control time step counter of the network) may not coincide, hence we should find a relationship between k_d and k_c. The first upcoming simulation time step $k_d^+(k_\mathrm{c})$ (see Figure 6) for link (u,d) at time instant $k_\mathrm{c}T_\mathrm{c}$ is computed by

$$k_d^+(k_\mathrm{c}) = \lceil (k_\mathrm{c}T_\mathrm{c} - \Delta T_{d,0})\,/c_d \rceil, \quad (51)$$

[3]Note that at intersection d, for each right-of-way, we consider a traffic signal.

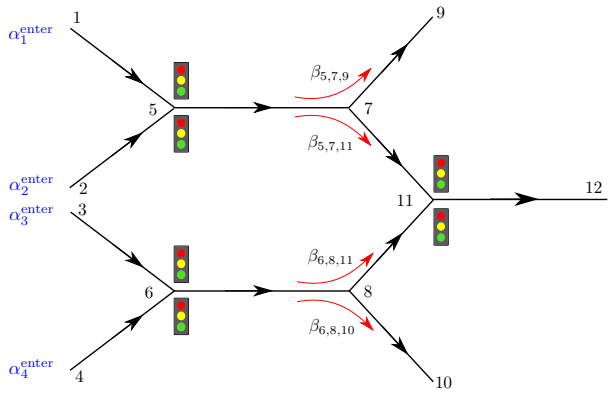

Figure 7. The urban traffic network used for the case study.

Table I
NETWORK PARAMETERS

N^{lane} [-]	ℓ^{link} [m]	μ [veh/s]	c [s]	$\beta_{5,7,9}$	$\beta_{5,7,11}$	$\beta_{6,8,11}$	$\beta_{6,8,10}$
1	500	0.8	60	0.6	0.4	0.6	0.4

Table II
MODEL PARAMETERS

v^{free} [m/s]	v^{idle} [m/s]	a^{acc} [m/s^2]	a^{dec} [m/s^2]	l^{veh} [m]
14	0.4	2	-2	7

where $\Delta T_{d,0}$ is the offset between the first simulation time step of link (u,d) ($k_d = 0$), and the first control time step ($k_{\text{c}} = 0$). For $k_d^{++}(k_{\text{c}})$, i.e., the last simulation time step that occurs during the current prediction time interval (see Figure 6), we can write

$$k_d^{++}(k_{\text{c}}) = \lfloor ((k_{\text{c}} + N_{\text{p}})T_{\text{c}} - \Delta T_{d,0})\,/c_d \rfloor. \qquad (52)$$

VII. CASE STUDY

In this section, we consider a case study to evaluate the designed smooth MPC controller. We focus on both the performance and the computation speed of the proposed control approach. We compare the values of total time spent, total emissions, and the value of the objective function computed by (49) for the proposed smooth MPC controller with those of the no-control case, a simple state-feedback controller, an optimal fixed-time controller that has been precomputed offline, and an MPC controller with pattern search as the optimization solver. Next, we also evaluate the controller from the point of computational efficiency (i.e., the CPU time).

A. Setup

The urban traffic network we use for the case study is presented in Figure 7. The network consists of 11 links, where all links have the same characteristics, i.e., number of lanes N^{lane}, length ℓ^{link}, and saturated leaving flow rate μ. Moreover, the traffic lights of all the three intersections that are controlled (see Figure 7) have the same cycle time c. The parameters $\beta_{5,7,9}$, $\beta_{5,7,11}$, $\beta_{6,8,11}$, and $\beta_{6,8,10}$ are the turning rates (i.e., the percentage of vehicles on a link that turn to a specific downstream link) of the vehicles at the corresponding intersection. We use the proposed integrated flow and emission model explained in Section V with the traffic parameters listed in Tables I and II to simulate the traffic in this network.

We run the simulations for three different demand profiles shown in Figure 8, where α_1^{enter}, α_2^{enter}, α_3^{enter}, and α_4^{enter} are the demands (i.e., the entering flow rates) of, respectively, origin 1, 2, 3, and 4 in Figure 7. These profiles have been selected in such a way that they highlight specific features, and such that they result in various traffic scenarios (e.g., under-saturated, saturated, and over-saturated) on different links of the traffic network. More specifically, 'Demand profile 1' corresponds to a relatively balanced case for the four demands $\alpha_1^{\text{enter}}, \ldots, \alpha_4^{\text{enter}}$. 'Demand profile 2' is a case where the demands at origin 2 (α_2^{enter}) and 3 (α_3^{enter}) are medium. For 'Demand profile 3', the demand at origin 3 is very high, while at the other origins the demand is low to medium. Comparing 'Demand profile 2' and 'Demand profile 3', we see a more irregular pattern for the latter case. Note that for each of the three demand profiles, at some periods during the simulation congestion occurs, more specifically on links (1,5), (2,5), (3,6), (4,6), (7,11), and (8,11). Moreover, on the intermediate links (5,7) and (6,8), a moderately congested traffic is sometimes observed. Each simulation run covers an entire hour. For each demand profile, we repeat the simulations 10 times and compute the CPU time and the realized values of the objective function, the total time spent, and the total emissions of CO, HC, and NO_{x} for each run. We will consider and compare the performance of different controllers that are described in Section VII-B with the no-control case, which corresponds to a case where there are no traffic lights at all at the intersections (i.e., all links are always open to the vehicles, unless the downstream road is blocked).

B. Controllers

State-feedback controller: At every control time step k, the feedback controller divides the green times $g_{u_1,d}$ and $g_{u_2,d}$ between two incoming links (u_1,d) and (u_2,d) at an intersection d considering the total number of vehicles $n_{u_1,d}$ and $n_{u_2,d}$ on the links, and the number of vehicles in the queues $q_{u_1,d}$ and $q_{u_2,d}$, i.e., based on the ratios $\frac{n_{u_1,d}(k)+\rho q_{u_1,d}(k)}{n_{u_1,d}(k)+n_{u_2,d}(k)+\rho\left(q_{u_1,d}(k)+q_{u_2,d}(k)\right)}$ and $\frac{n_{u_2,d}(k)+\rho q_{u_2,d}(k)}{n_{u_1,d}(k)+n_{u_2,d}(k)+\rho\left(q_{u_1,d}(k)+q_{u_2,d}(k)\right)}$, respectively. Note that ρ is a parameter that can be tuned.

Optimal fixed-time controller: This controller has constant signal settings that have been optimized offline for each demand class (Demand profiles 1, 2, 3) separately. For this aim, the objective function (i.e., the weighted sum of the total time spent and total emissions) was minimized for each demand class in a 1-h simulation using brute force with a grid size of 0.1, and a fixed traffic signal setting was obtained.

MPC controller with RProp: This controller uses a gradient-based optimization approach based on the resilient back-propagation algorithm (see Appendices B and C), and is designed as explained in Section VI, with the smooth and extended S-model (see Sections II and IV-B) as the prediction flow model and VT-micro [34] as the prediction emission model of the controller. These two models are integrated using the mesoscopic framework proposed in Section V. The control sampling time is equal to the cycle time of the traffic lights

in the traffic network (60 s), and the length of the prediction horizon is 7 time steps (note that according to the tuning rules proposed by Hegyi et al. [35], the horizon length is selected such that it is longer than the time needed for vehicles to cross the network).

MPC controller with pattern search: For evaluation of the CPU time, we compare the smooth MPC controller with a nonsmooth MPC controller. In general, several nonsmooth optimization algorithms exist, among which pattern search and genetic algorithm are the most frequently used algorithms. We noticed from several experiments that the performance and the computation speed for pattern search are more satisfactory than those of the genetic algorithm. Hence, we used pattern search for the case study that is implemented in the Global Optimization Toolbox of MATLAB, version R2015B.

The experiments were run on a PC with an Intel Xeon Quad core E5-1620 V3 CPU with a clock speed of 3.5 GHz.

C. Results and discussions

Since the MPC optimization problem may in general be a nonconvex one, in order to prevent the gradient-based approach from giving a solution that is only locally optimal and that may give a much worse performance than the global optimal value, we first ran a set of offline experiments. The aim was to determine how starting control points may play a role in getting a good approximation for the global optimum, and that what is the best choice for the selection of the starting points. These points have been selected in a structured way, i.e., we have three sets of deterministic starting points and two sets of random ones. The deterministic starting points include the shifted suboptimal solution of the previous control time step, the average of the shifted suboptimal solutions determined in the two previous control time steps, and the average of all the shifted suboptimal solutions found in the previous control time steps. The random starting points are feasible points with a uniform distribution. The results of the offline optimization did not show a noticeable difference in the overall performance of the controlled system for different choices of the starting points. Hence, for the online application of the optimization-based control approaches in this paper, we used one set of starting control points at every control time step, in particular, the shifted suboptimal solution of the previous control time step. For pattern search, we used a similar approach.

Tables III-VII show the resulting value of the objective function, the total time spent, and the total emissions of CO, HC, NO_x for the no-control case, and for the state-feedback, optimal fixed-time, and smooth and nonsmooth MPC controllers applied to the urban traffic network illustrated in Figure 7. We have considered the following values for the parameters given in (49): $w_T = 0.3$, $w_{E_{\mathrm{CO}}} = w_{E_{\mathrm{HC}}} = w_{E_{\mathrm{NO_x}}} = 0.2$, $T^{\mathrm{t}} = 10^5$ [s], $E^{\mathrm{t}}_{\mathrm{CO}} = E^{\mathrm{t}}_{\mathrm{HC}} = E^{\mathrm{t}}_{\mathrm{NO_x}} = 1$ [kg], $w_v = 0$. These results show that compared with the no-control case, the overall performance of the system is improved significantly with the state-feedback and optimal feedback controllers. The reason that the fixed-time traffic signal setting performs much worse than the state-feedback controller for Demand profile 3, is the irregular pattern of this demand profile

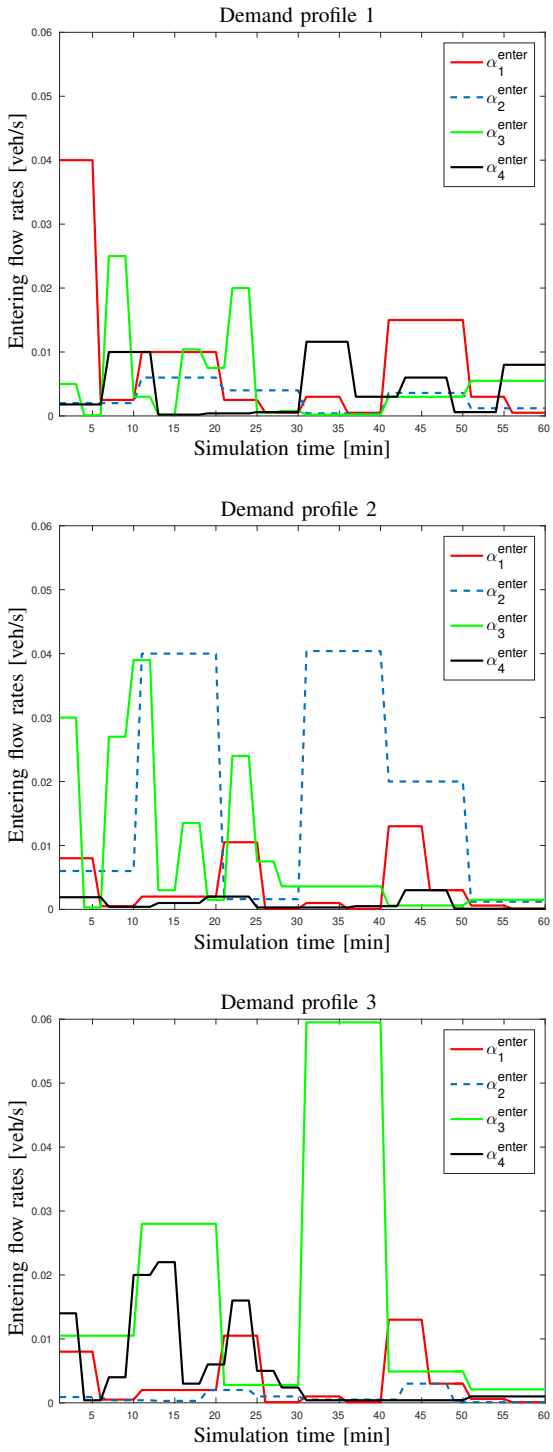

Figure 8. Different demand profiles used for 3 simulations cases.

compared with the other two profiles (see Figure 8). A constant setting is of course not always expected to be the best choice for both a very high and a very low demand, which occurs in Demand profile 3 for origin 3. However, a state-feedback controller that considers the queue lengths can adapt its traffic signal setting w.r.t. the current traffic state, and hence performs better than the fixed-time controller.

From Table VI, the improvement of the system's performance for the smooth MPC controller is higher than for the

Table III
VALUE OF THE OBJECTIVE FUNCTION, TOTAL TIME SPENT, AND TOTAL EMISSIONS OVER THE ENTIRE SIMULATION PERIOD FOR THE NO-CONTROL CASE

	Objective function	TTS [veh.s]	TE_{CO} [kg]	TE_{HC} [kg]	TE_{NO_x} [kg]
Demand profile 1	214.2	$2.9016 \cdot 10^7$	540.2	34.2	61.4
Demand profile 2	91.0	$1.2279 \cdot 10^7$	230.3	14.6	26.0
Demand profile 3	61.8	$8.3106 \cdot 10^6$	156.8	9.9	17.7

Table IV
VALUE OF THE OBJECTIVE FUNCTION, TOTAL TIME SPENT, AND TOTAL EMISSIONS OVER THE ENTIRE SIMULATION PERIOD FOR THE STATE-FEEDBACK CONTROLLER

	Objective function	TTS [veh.s]	TE_{CO} [kg]	TE_{HC} [kg]	TE_{NO_x} [kg]
Demand profile 1	4.7	$6.2404 \cdot 10^5$	12.0	0.8	1.2
Demand profile 2	5.3	$7.0179 \cdot 10^5$	13.6	0.9	1.4
Demand profile 3	6.2	$8.1954 \cdot 10^5$	16.0	1.0	1.7

Table V
VALUE OF THE OBJECTIVE FUNCTION, TOTAL TIME SPENT, AND TOTAL EMISSIONS OVER THE ENTIRE SIMULATION PERIOD FOR THE OPTIMAL FIXED-TIME CONTROLLER

	Objective function	TTS [veh.s]	TE_{CO} [kg]	TE_{HC} [kg]	TE_{NO_x} [kg]
Demand profile 1	4.9	$6.2580 \cdot 10^5$	12.9	0.8	1.3
Demand profile 2	1.2	$1.2192 \cdot 10^5$	3.6	0.3	0.3
Demand profile 3	54.9	$7.4177 \cdot 10^5$	138.8	8.8	15.7

Table VI
VALUE OF THE AVERAGE CPU TIME, OBJECTIVE FUNCTION, TOTAL TIME SPENT, AND TOTAL EMISSIONS OVER THE ENTIRE SIMULATION PERIOD FOR THE SMOOTH MPC CONTROLLER WITH RPROP

	Average CPU time per control time step [s]	Objective function	TTS [veh.s]	TE_{CO} [kg]	TE_{HC} [kg]	TE_{NO_x} [kg]
Demand profile 1	1697.2	4.4	$6.4377 \cdot 10^5$	10.7	0.7	1.1
Demand profile 2	5466.4	0.8	$1.0326 \cdot 10^5$	2.0	0.1	0.2
Demand profile 3	13241.1	3.3	$4.4603 \cdot 10^5$	8.4	0.5	0.9

Table VII
VALUE OF THE AVERAGE CPU TIME, OBJECTIVE FUNCTION, TOTAL TIME SPENT, AND TOTAL EMISSIONS OVER THE ENTIRE SIMULATION PERIOD FOR THE NONSMOOTH MPC CONTROLLER WITH PATTERN SEARCH

	Average CPU time per control time step [s]	Objective function	TTS [veh.s]	TE_{CO} [kg]	TE_{HC} [kg]	TE_{NO_x} [kg]
Demand profile 1	76479	14.0	$1.8726 \cdot 10^6$	35.8	2.3	3.9
Demand profile 2	152631	7.0	$9.4632 \cdot 10^5$	17.9	1.2	1.9
Demand profile 3	151867	28.2	$3.7954 \cdot 10^6$	71.6	4.5	8.0

Table VIII
PERCENTAGE OF IMPROVEMENT OF THE OBJECTIVE VALUE W.R.T. THE STATE-FEEDBACK CONTROLLER: $\frac{J_{\text{state-feedback}} - J_{\text{MPC}}}{J_{\text{state-feedback}}}$ (+: IMPROVED, -: BECAME WORSE)

	MPC with RProp	MPC with pattern search
Demand profile 1	+6%	-66%
Demand profile 2	+84%	-32%
Demand profile 3	+47%	-354%

Table IX
PERCENTAGE OF IMPROVEMENT OF THE OBJECTIVE VALUE W.R.T. THE OPTIMAL FIXED-TIME CONTROLLER: $\frac{J_{\text{fixed-time}} - J_{\text{MPC}}}{J_{\text{fixed-time}}}$ (+: IMPROVED, -: BECAME WORSE)

	MPC with RProp	MPC with pattern search
Demand profile 1	+10%	-185%
Demand profile 2	+34%	-483%
Demand profile 3	+94%	+48.6%

other controllers w.r.t. the no-control case. Therefore, we can conclude that the online smooth MPC approach for the given case study is highly beneficial compared with the other given controllers.

Next, we compare both the computation time and the performance of the smooth MPC controller that uses RProp with those of a nonsmooth MPC controller that uses pattern search. The corresponding results showing the performance of pattern search are given in Table VII. We see that for all the given demand profiles, the CPU time for the MPC controller with RProp is 12-45 times less than the CPU time for pattern search. This indicates that the gradient-based optimization approach performs significantly better than pattern search considering the computation speed. Moreover, the realized value of the objective function for a 1-hour simulation for the gradient-based optimization approach is almost 31.5% of the realized value of the objective function for pattern search for 'Demand profile 1', 11% for 'Demand profile 2', and 11.5% for 'Demand profile 3' (compare Tables VI and VII). In addition to that, by looking at the realized values of the total time spent and total emissions of CO, HC, and NO_x individually, we see that all these quantities are prominently smaller for the gradient-based optimization compared with pattern search. Note that for pattern search the optimization procedure takes long and hence, sometimes the maximum number of iterations is reached before the optimum values are found.

We have listed the percentages of improvement for the MPC controllers w.r.t. the state-feedback and optimal fixed-time controllers in Tables VIII and IX. These results show that for the traffic network shown in Figure 7, the proposed smooth MPC controller that uses RProp is the most efficient controller among the other given controllers. Additionally,

from the smooth and nonsmooth MPC controllers, both the performance and the CPU time of the smooth one are much better.

VIII. Conclusions and future work

We have proposed a highly efficient smooth model-predictive controller for urban traffic networks, with the aim of finding a balanced trade-off between reduction of the total time spent by the vehicles and the total emissions. We have applied a gradient-based optimization approach based on the resilient back-propagation (RProp) to find the suboptimal solution of the MPC controller. To make the proposed gradient-based optimization approach applicable to different physical systems, we have introduced general smoothening methods for nonsmooth mathematical models. We have also introduced a general formulation for transforming a time-delayed differential equation in the continuous-time domain into an equivalent discrete-time difference equation.

The simulation results have shown that the smooth MPC controller improves the performance of the network significantly w.r.t. the no-control case, and state-feedback, optimal fixed-time, and nonsmooth MPC-based controlled cases. Moreover, the smooth (gradient-based) optimization method is much faster than the nonsmooth one (the CPU time of the smooth method is 12-45 times less than the CPU time of the nonsmooth method). Note that although the resulting CPU time for the smooth optimization-based controller shows a significant decrease compared to that of the nonsmooth optimization-based controller, it is currently not yet suited for real-time control. Therefore, further improvements in computation speed of the smooth optimization-based controller should be obtained. This improvement may be achieved by using dedicated software and hardware, distributed MPC, fast MPC techniques, parameterized control approaches, etc.

Topics for future work include implementing the smooth optimization approach to a large-scale network considering a multi-level and/or multi-agent control architecture. Additionally, to make the MPC controller faster, we can consider parameterized control laws. We also suggest an extensive validation of the proposed control approach for various networks with real-life datasets.

Appendix A
Proof of Lemma III.1

Suppose that we want to find the effective inflow $\bar{u}_h(t)$ during $[t, t+h)$ for a delayed differential equation with a time-varying delay function, i.e.,

$$\bar{u}_h(t) = \frac{1}{h}\int_{t-\tau(t)}^{t+h-\tau(t+h)} u(t)\mathrm{d}t. \tag{53}$$

Assuming a piece-wise constant inflow function $u(t)$ in $[st, st+h)$, $s \in \mathbb{Z}$, (see Figure 9), and $\tau(t) = \delta(t)h + \gamma(t)$, $\forall t \in [t, t+h)$, there are two cases:

Case 1. Suppose that we have $\delta(k) \leq \delta(k-1)+1$; then by (53) can be extended as

$$\bar{u}_h(t) = \frac{1}{h}\int_{t-\delta(t)h-\gamma(t)}^{t-\delta(t)h} u(\theta)\mathrm{d}\theta + \ldots$$

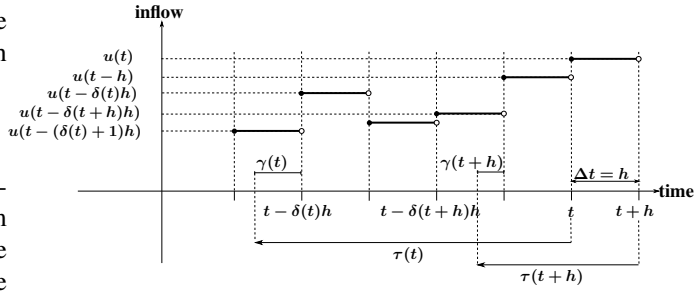

Figure 9. The effective inflow during $[t, t+h)$ for a delayed-time differential equation with a time-varying delay.

$$\begin{aligned}
&+\frac{1}{h}\int_{t-\delta(t+h)h}^{t+h-\delta(t+h)h-\gamma(t+h)} u(\theta)\mathrm{d}\theta\\
&=\frac{1}{h}\gamma(t)u(t-(\delta(t)+1)h)\\
&+\frac{1}{h}\left(h\sum_{i=1}^{\delta(t)-\delta(t+h)} u(t-(\delta(t)-i+1)\,h)\right)\\
&+\frac{h-\gamma(t+h)}{h}u(t-\delta(t+h)h).
\end{aligned}$$

In order to represent the above expression in discrete-time domain, we substitute $t+h$ by kh, and t by $(k-1)h$. This way (11) will be obtained.

Case 2. For $\delta(k) > \delta(k-1)+1$, in a similar way as for Case 1 (11) can be obtained.

Appendix B
Pontryagin's minimum principle

Our problem includes finding a suboptimal control strategy solving a smooth optimization problem for a discrete-time nonlinear system. We first explain how to solve such a problem using Pontryagin's minimum principle [16]. Consider the discrete-time nonlinear system:

$$\boldsymbol{x}(k+1) = \boldsymbol{f}(k, \boldsymbol{x}(k), \boldsymbol{u}(k)), \tag{54}$$

where $\boldsymbol{f}(\cdot)$ can in general be a nonlinear smooth function. Define a performance index for (54) during the prediction time interval $[k_\mathrm{c}T_\mathrm{c}, (k_\mathrm{c}+N_\mathrm{p})T_\mathrm{c})$ by

$$\mathcal{J}(k_\mathrm{c}) = \mathcal{J}^\mathrm{t}(k_\mathrm{c}) + \sum_{k=k_\mathrm{c}}^{k_\mathrm{c}+N_\mathrm{p}-1} \mathcal{J}^\mathrm{s}(k), \tag{55}$$

with $\mathcal{J}^\mathrm{s}(\cdot)$ the stage objective function, which at control time step $k \in \{k_\mathrm{c}, \ldots, k_\mathrm{c}+N_\mathrm{p}-1\}$ can be given by an expression of the control time step k, the state vector $\boldsymbol{x}(k)$, and the control input $\boldsymbol{u}(k)$, i.e., $\mathcal{J}^\mathrm{s}(k) = J^\mathrm{s}(k, \boldsymbol{x}(k), \boldsymbol{u}(k))$, and with $\mathcal{J}^\mathrm{t}(\cdot)$ the terminal objective function, which at control time step k_c can be given by an expression of the terminal control time step $k_\mathrm{c}+N_\mathrm{p}$ and the terminal state vector $\boldsymbol{x}(k_\mathrm{c}+N_\mathrm{p})$, i.e., $\mathcal{J}^\mathrm{t}(k_\mathrm{c}) = J^\mathrm{t}(k_\mathrm{c}+N_\mathrm{p}, \boldsymbol{x}(k_\mathrm{c}+N_\mathrm{p}))$. In our specific problem, the stage objective is obtained by summing up the total time spent (50) and the total emissions (37)-(40) over all links $(u,d) \in \mathcal{L}$. The Hamiltonian function [16] for minimizing (55) w.r.t. (54) at control time step $k \in \{k_\mathrm{c}, \ldots, k_\mathrm{c}+N_\mathrm{p}-1\}$ is defined by

$$\begin{aligned}
&H(k, \boldsymbol{\lambda}(k+1), \boldsymbol{x}(k), \boldsymbol{u}(k)) =\\
&\quad J^\mathrm{s}(k, \boldsymbol{x}(k), \boldsymbol{u}(k)) + \boldsymbol{\lambda}^\top(k+1)\cdot \boldsymbol{f}(k, \boldsymbol{x}(k), \boldsymbol{u}(k)),
\end{aligned} \tag{56}$$

where $\boldsymbol{\lambda}(\cdot)$ is called the costate. Pontryagin's minimum principle [15] states that for an input function $\bar{\boldsymbol{u}}(\cdot)$ to make the performance index (55) optimal, the following should hold for all $k \in \{k_\mathrm{c}, \ldots, k_\mathrm{c} + N_\mathrm{p} - 1\}$ at the same time [16]:

$$\boldsymbol{x}(k+1) = \frac{\partial H(k, \boldsymbol{\lambda}(k+1), \boldsymbol{x}(k), \bar{\boldsymbol{u}}(k))}{\partial \boldsymbol{\lambda}(k+1)}, \tag{57}$$

$$\boldsymbol{\lambda}(k) = \frac{\partial H(k, \boldsymbol{\lambda}(k+1), \boldsymbol{x}(k), \bar{\boldsymbol{u}}(k))}{\partial \boldsymbol{x}(k)}, \tag{58}$$

$$\boldsymbol{G}(k) = \frac{\partial H(k, \boldsymbol{\lambda}(k+1), \boldsymbol{x}(k), \bar{\boldsymbol{u}}(k))}{\partial \boldsymbol{u}(k)} = 0, \tag{59}$$

where $\boldsymbol{G}(\cdot)$ is called the reduced gradient [20]. In order to numerically solve (57), we can start from the initial state of the system, $\boldsymbol{x}(k_\mathrm{c})$. To solve (58) via backward integration, we start from $\boldsymbol{\lambda}(k_\mathrm{c} + N_\mathrm{p})$, which is given by

$$\boldsymbol{\lambda}(k_\mathrm{c} + N_\mathrm{p}) = \frac{\partial J^\mathrm{t}\left(k_\mathrm{c} + N_\mathrm{p}, \boldsymbol{x}(k_\mathrm{c} + N_\mathrm{p})\right)}{\partial \boldsymbol{x}(k_\mathrm{c} + N_\mathrm{p})}. \tag{60}$$

In our problem, we considered $\mathcal{J}^\mathrm{t}(k_\mathrm{c} + N_\mathrm{p}) = \alpha \left\| \boldsymbol{x}(k_\mathrm{c} + N_\mathrm{p}) \right\|$ for the terminal objective (with α a positive constant) to reduce the final queue lengths on the links. Applying an iterative algorithm involving an adaptive optimization method known as the resilient back-propagation (RProp) algorithm, which was initially introduced in [22], and was more recently extended in [21], we can find the optimal input function $\bar{\boldsymbol{u}}(\cdot)$.

APPENDIX C
RESILIENT BACK-PROPAGATION ALGORITHM

Using RProp [22], we can find the increment $\Delta\bar{\boldsymbol{u}}_\ell$ of the control vector at iteration ℓ based on the values of the reduced gradients in the current and in the previous iterations, i.e., $\boldsymbol{G}_{(\ell)}$ and $\boldsymbol{G}_{(\ell-1)}$. The elements of $\boldsymbol{G}$ and $\Delta\bar{\boldsymbol{u}}$ are indicated by, respectively, G_i and $\Delta\bar{u}_i$ for $i = 1, \ldots, |\bar{\boldsymbol{u}}|$, where $|\cdot|$ gives the number of entries of a vector. The elements of $\Delta\bar{\boldsymbol{u}}_\ell$ at time step $k \in \{k_\mathrm{c}, \ldots, k_\mathrm{c} + N_\mathrm{p} - 1\}$ are updated by

$$\bar{u}_{i,(\ell)}(k) = \mathrm{sat}\left(\bar{u}_{i,(\ell-1)}(k) + \Delta\bar{u}_{i,(\ell)}(k)\right), \tag{61}$$

where $\mathrm{sat}(\bar{u}_i) = \begin{cases} u_{i,\max}, & \text{if } \bar{u}_i \geq u_{\max}, \\ \bar{u}_i, & \text{if } u_{\min} < \bar{u}_i < u_{\max}, \\ u_{i,\min}, & \text{if } \bar{u}_i \leq u_{\min}, \end{cases}$ with $u_{i,\max}$ and $u_{i,\min}$ the upper and the lower bound for the element $\bar{u}_i$. Finally, for $i = 1, \ldots, |\bar{\boldsymbol{u}}|$ RProp (for $0 < \eta^- < 1$ and $\eta^+ > 1$) gives

$$\Delta u_{i,(\ell)}(k) = \begin{cases} -\mathrm{sign}\left(G_{i,(\ell)}(k)\right)\eta^+ |\Delta u_{i,(\ell-1)}(k)| \\ \qquad\qquad \text{for } G_{i,(\ell-1)}(k) G_{i,(\ell)}(k) > 0, \\ \Delta u_{i,(\ell)}(k) = -\eta^- \Delta u_{i,(\ell-1)}(k) \qquad \text{otherwise.} \end{cases}$$

ACKNOWLEDGMENTS

Research supported by the European COST Action TU1102, the NWO-NSFC project "Multi-level predictive traffic control for large-scale urban networks" (629.001.011), and the European Union's Seventh Framework Programme (FP/2007-2013) / FP7-ICT-2013.3.4, project LOCAL4GLOBAL (n. 611538).

REFERENCES

[1] A. Sjodin, K. Persson, K. Andreasson, B. Arlander, and B. Galle, "On-road emission factors derived from measurements in a traffic tunnel," *International Journal of Vehicle Design*, vol. 20, no. 1–4, pp. 147–158, 1998.

[2] W. P. Anderson, S. K. Pavlos, E. J. Miller, and R. Buliung, "Simulating automobile emissions in an integrated urban model," *Transportation Research Record*, vol. 1520, pp. 71–80, 1996.

[3] M. Barth and K. Boriboonsomsin, "Real-world carbon dioxide impacts of traffic congestion," *Transportation Research Record*, vol. 2058, pp. 163–171, 2008.

[4] M. Barth, F. An, J. Norbeck, and M. Ross, "Modal emissions modeling: A physical approach," *Transportation Research Record*, vol. 1520, pp. 81–88, 1996.

[5] J. M. Thomson, "Reflections on the economics of traffic congestion," *Journal of Transport Economics and Policy*, vol. 32, no. 1, pp. 93–112, 1997.

[6] C. Diakaki, M. Papageorgiou, and K. Aboudolas, "A multivariable regulator approach to traffic-responsive network-wide signal control," *Control Engineering Practice*, vol. 10, no. 2, pp. 183–195, 2002.

[7] C. Diakaki, V. Dinopoulou, K. Aboudolas, M. Papageorgiou, E. Ben-Shabat, E. Seider, and A. Leibov, "Extensions and new applications of the traffic-responsive urban control strategy: Coordinated signal control for urban networks," *Transportation Research Record*, vol. 1856, pp. 202–211, 2003.

[8] K. Aboudolas, M. Papageorgiou, A. Kouvelas, and E. Kosmatopoulos, "A rolling-horizon quadratic-programming approach to the signal control problem in large-scale congested urban road networks," *Transportation Research Part C*, vol. 18, no. 5, pp. 680–694, 2010.

[9] 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, Sep. 2007.

[10] T. Bellemans, B. De Schutter, and B. De Moor, "Model predictive control for ramp metering of motorway traffic: A case study," *Control Engineering Practice*, vol. 14, no. 7, pp. 757–767, 2006.

[11] D. C. Gazis and R. B. Potts, "Reduction of area-wide emissions using an efficient model-based traffic control strategy," in *Proceedings of the* 2^{nd} *International Symposium on Theory of Traffic Flow,*, London, UK, 1963, pp. 221–237.

[12] 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*, vol. 17, no. 2, pp. 163–174, 2009.

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

[14] A. Barisone, D. Giglio, R. Minciardi, and R. Poggi, "A macroscopic traffic model for real-time optimization of signalized urban areas," in *Proceedings of the* 41^{st} *IEEE Conference on Decision and Control*, Las Vegas, USA, 2002, pp. 900–903.

[15] H. Kagiwada, R. Kalaba, and Y. Thomas, "Exact solution of Pontryagin's equations of optimal control - part 1," *Journal of Optimization Theory and Applications*, vol. 5, no. 1, pp. 12–22, 1970.

[16] F. L. J. Lewis, *Optimal Control*. USA: John Wiley & Sons, Inc., 1986.

[17] 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.

[18] M. Papageorgiou and A. Kotsialos, "Nonlinear optimal control applied to coordinated ramp metering." *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, pp. 920–933, 2004.

[19] C. Pasquale, I. Papamichail, C. Roncoli, S. Sacone, S. Siri, and M. Papageorgiou, "Two-class freeway traffic regulation to reduce congestion and emissions via nonlinear optimal control," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 85–99, 2015.

[20] M. Papageorgiou and M. Marinaki, "A feasible direction algorithm for the numerical solution of optimal control problems." Dynamic Systems & Simulation Laboratory, Technical University of Crete, Chania, Greece, Tech. Rep., 1995.

[21] T. M. Bailey, "Convergence of Rprop and variants," *Neurocomputing*, vol. 159, pp. 90–95, 2015.

[22] M. Riedmiller and H. Braun, "A directive adaptive method for faster backpropagation learning: the RPROP algorithm," in *Proceedings of the IEEE International Conference on Neural Networks*, San Francisco, USA, 1993, pp. 586–591.

[23] C. F. Daganzo, "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transportation Research Part B: Methodological*, vol. 28, no. 4, pp. 269–287, 1994.

[24] L. Li, R. Negenborn, and B. De Schutter, "A sequential linear programming approach for flow assignment in intermodal freight transport," in *Proceedings of the* 16^{th} *International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, The Hague, The Netherlands, Oct. 6-9, 2013, pp. 1224–1230.

[25] R. M. P. Goverde, "A delay propagation algorithm for large-scale railway traffic networks," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 3, pp. 269–287, 2010.

[26] C. Chen and O. L. Mangasarian, "Smoothing methods for convex inequalities and linear complementarity problems," *Mathematical Programming*, vol. 71, no. 1, pp. 51–69, 1995.

[27] H. Kashani and G. Saridis, "Intelligent control for urban traffic systems," *Automatica*, vol. 2, no. 19, pp. 191–197, 1983.

[28] S. Lin, "Efficient model predictive control for large-scale urban traffic networks," Ph.D. dissertation, Delft University of Technology, April 2011.

[29] A. Paz, V. Molano, E. Martinez, C. Gaviria, and C. Arteaga, "Calibration of traffic flow models using a memetic algorithm," *Transportation Research Part C*, vol. 55, pp. 432–443, 2015.

[30] A. Jamshidnejad, I. Papamichail, M. Papageorgiou, and B. De Schutter, "A mesoscopic integrated urban traffic flow-emission model," *Transportation Research Part C*, vol. 75, pp. 45–83, 2017.

[31] K. Ahn, H. Rakha, A. Trani, and M. van Aerde, "Estimating vehicle fuel consumption and emissions based on instantaneous speed and acceleration levels," *Journal of Transportation Engineering*, vol. 128, no. 2, pp. 182–190, 2002.

[32] J. Maciejowski, *Predictive Control with Constraints*. London, UK: Prentice Hall, 2002.

[33] M. Morari and J. H. Lee, "Model predictive control: Past, present, and future," *Computers and Chemical Engineering*, vol. 23, no. 4–5, pp. 667–682, 1999.

[34] K. Ahn, A. Trani, H. Rakha, and M. van Aerde, "Microscopic fuel consumption and emission models," in *Proceedings of the* 78^{th} *Annual Meeting of the Transportation Research Board*, Washington DC, USA, Jun.–Jul. 1999.

[35] A. Hegyi, B. De Schutter, and H. Hellendoorn, "Model predictive control for optimal coordination of ramp metering and variable speed limits," *Transportation Research Part C*, vol. 13, no. 3, pp. 185–209, Jun. 2005.