

Technical report 15-026

Smoothing for efficient solution of model predictive control for urban traffic networks considering endpoint penalties*

A. Jamshidnejad, H. Hellendoorn, S. Lin, and B. De Schutter

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

A. Jamshidnejad, H. Hellendoorn, S. Lin, and B. De Schutter, "Smoothing for efficient solution of model predictive control for urban traffic networks considering endpoint penalties," *Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Las Palmas de Gran Canaria, Spain, pp. 2837–2842, Sept. 2015.

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/15_026.html

Smoothing for Efficient Solution of Model Predictive Control for Urban Traffic Networks Considering Endpoint Penalties

Anahita Jamshidnejad^a, Hans Hellendoorn^a, Shu Lin^b, and Bart De Schutter^a

^aDelft University of Technology,
Delft Center for Systems and Control (DCSC),
Mekelweg 2, 2628 CD Delft, The Netherlands
Email: {a.jamshidnejad, j.hellendoorn, b.deschutter}@tudelft.nl

^b School of Computer and Control Engineering,
University of Chinese Academy of Sciences,
HuaiBeiZhen 380, HuaiRou District,
Beijing, 101408, China
Email: slin@ucas.ac.cn

Abstract—Traffic congestion together with emissions has become a big problem in urban areas. Traffic-responsive control systems aim to make the best use of the existing road capacity. Here, we propose a model predictive controller for urban traffic networks, where the goal of the control is to find a balanced trade-off between reduction of congestion and emissions. The cost function is defined as a weighted combination of the total time spent, TTS, (as a criterion for evaluating the congestion level), the total emissions, TE, and the expected values of the TTS and TE caused by the vehicles that remain in the network at the end of the prediction horizon until they leave the network. We propose a method for estimation of the expected time spent and emissions by the remaining vehicles, where our method is based on a K Shortest path algorithm. For the prediction model of the MPC-based controller, we use a macroscopic integrated flow-emission model that includes the macroscopic flow S-model and the microscopic emission model, VT-micro. Since the S-model includes non-smooth functions, it does not allow us to benefit from efficient gradient-based methods to solve the optimization problem of the MPC-based controller. Therefore, in this paper we also propose smoothing methods for the S-model.

I. INTRODUCTION

Traffic congestion and emissions are considered as a major problem in modern urban areas, since they result in lost of time and energy, and in distribution of harmful substances including nitrogen oxides (NO_x), hydrocarbon (HC), carbon monoxide (CO), carbon dioxide (CO_2), and particulate matter in the environment. Real-time traffic-responsive control systems include efficient approaches that target the problem of traffic congestion and emissions by capacity management [1,2]. Among traffic-responsive control approaches, optimization-based control methods, especially model predictive control (MPC), have proven their efficiency for management of traffic in both freeways and in urban traffic areas [3,4,5].

The focus of this paper is on designing an MPC-based controller for urban traffic networks. MPC uses an internal

prediction model to estimate the future states of the system and to solve the optimization problem along a finite prediction horizon. The optimal control signal is implemented for one time step, and the prediction horizon is shifted for one step. Then the entire procedure is repeated (see [6] for more details).

In a majority of the available literature on traffic model predictive control for urban areas, prevention of congestion is considered as the main objective of the control system. Aboudolas et al. in [3] propose a rolling horizon approach that solves an optimization problem using quadratic programming, with the aim to minimize the risk of over-saturation and spill-back of the queues. While reduction of emissions has been considered in some literature [7,8,9,10], the number of these works is still limited.

Here, we propose an MPC-based real-time traffic-responsive controller that optimizes a multi-objective cost function including both emissions and congestion. We use a macroscopic integrated flow-emission model, called the VT-S model, that integrates a macroscopic flow model, i.e., the S-model [11] and a microscopic emission model, i.e., the VT-micro [12]. The S-model includes non-smooth functions; consequently, we cannot use efficient gradient-based methods to solve the optimization problem of the MPC. Therefore, we propose smoothing methods for the S-model so that we can finally obtain a smooth flow model. We also present an approach based on a K Shortest path algorithm that estimates the expected time spent and emissions by the vehicles that remain in the network after the prediction time interval until they all leave the network.

The paper is organized as follows; in Section II we give an introduction about MPC. Section III presents the flow and emission models used as the prediction models of the MPC controller, and formulates the optimization problem defining the objective function as a weighted combination of the total time spent (TTS) and total emissions (TE) and their expected

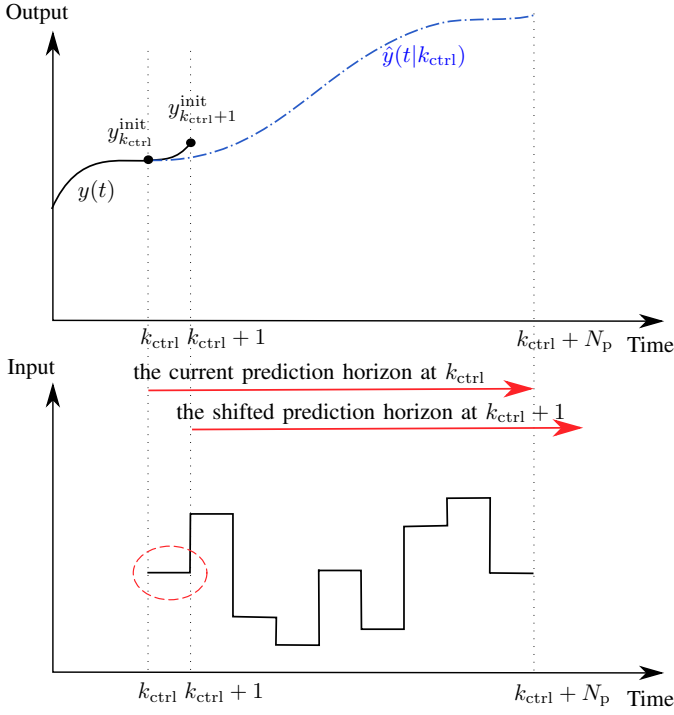


Fig. 1. Illustration of the main idea used in a model predictive controller (note that $y_{k_{ctrl}}^{init}$ is the measured value of the output signal at k_{ctrl} that is used to estimate the corresponding state, which will be the initial condition for the computations of the next time step $[(k_{ctrl} + 1)T_{ctrl}, (k_{ctrl} + 2)T_{ctrl})$)

values after the prediction interval. In Section IV we present methods for making the flow model smooth. Finally, Section V concludes the paper.

II. MODEL PREDICTIVE CONTROL (MPC)

Model predictive control (MPC) is an optimization-based control approach that makes use of the measured state received from the system at every control time step. MPC is mostly used in real-time control applications. Unlike the classical optimal control approaches, which solve the optimization problem along an infinite horizon, MPC benefits from a finite (rolling) horizon, i.e., the prediction horizon. The controller uses an internal model, which estimates the future states of the system along the prediction horizon (consisting of N_p control time steps) based on the latest measurements. Then the controller solves an optimization problem to minimize a cost function.

Figure 1 illustrates the basic idea of MPC, where $y(t)$ denotes the trajectory of the measured output signal of the controlled system, and $\hat{y}(t|k_{ctrl})$ shows the predicted value of the output signal at t by the internal model of the MPC controller supposed that the controller knows the measured value of the output at the current time step k_{ctrl} .

The output of the optimization problem is a sequence of optimal control signals, from which the first value will be injected as input to the system for a time interval of one control time step (see Figure 1). At the next control time step, new measurements will be received by the controller (i.e., $y_{k_{ctrl}+1}^{init}$ in Figure 1), the prediction horizon will be rolled forward for one time step, and the entire optimization procedure will be repeated in the same way.

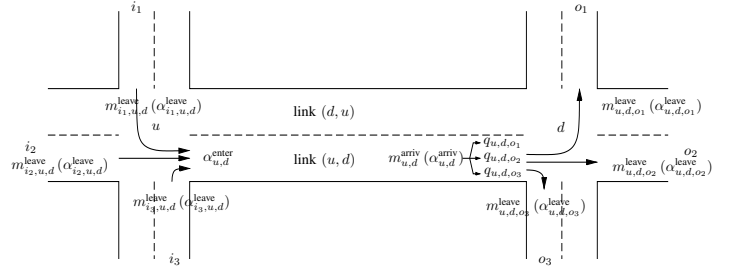


Fig. 2. Representation of a link between two intersections [11]

III. FORMULATION OF AN MPC-BASED CONTROLLER FOR URBAN TRAFFIC

Our proposed approach for a real-time traffic-responsive controller involves designing an MPC-based controller. The aim of the control system is to find a balanced trade-off between reduction of the total time spent (TTS) by the vehicles and the total emissions (TE). Therefore, the internal model of the MPC controller should be able to estimate both the states of the system and the emissions caused by the current traffic. Since the optimization problem of the MPC should be solved online, the internal model should also provide a balance between accuracy and simplicity.

We use a macroscopic integrated flow-emission model as the prediction model of the MPC-based controller. The integrated model includes a macroscopic flow model, called the S-model, and a microscopic emission model, called the VT-micro. Since the S-model includes non-smooth functions, it does not allow us to benefit from efficient gradient-based methods to solve the optimization problem of the MPC-based controller. Hence, we propose methods that help to make the S-model smooth. We also present an approach that estimates the expected time spent and emissions by the vehicles that remain in the network after the prediction time interval, where this approach is based on a K Shortest path algorithm.

A. Internal model of the MPC-based controller

1) *Urban flow model:* The behavior of traffic in urban areas is highly nonlinear; therefore, we prefer a nonlinear flow model. Lin et al. in [11] propose a simplified macroscopic urban traffic flow model called the S-model, which is able to consider different cycle times for the intersections within the urban network and also the time delay required by the vehicles to travel the distance between the upstream point of the link and the tail of the awaiting downstream queue. The S-model can represent both free and congested traffic scenarios.

In the S-model, a road is represented as a link (u, d) that is extended between the upstream and downstream nodes (i.e., intersections) u and d . The set of all links is denoted by L , and J is the set of all intersections within the network. The state variables of the S-model are $n_{u,d}(k_d)$ and $q_{u,d}(k_d)$, which are, respectively, the number of the vehicles and the queue length (counting in numbers) in link (u, d) at time step k_d . These state variables are updated at every simulation time step of link (u, d) using the following relationships:

$$n_{u,d}(k_d + 1) = n_{u,d}(k_d) + (\alpha_{u,d}^{enter}(k_d) - \alpha_{u,d}^{leave}(k_d)) c_d \quad (1)$$

$$q_{u,d}(k_d) = \sum_{o \in O_{u,d}} q_{u,d,o}(k_d) \quad (2)$$

with

$$q_{u,d,o}(k_d + 1) = q_{u,d,o}(k_d) + (\alpha_{u,d,o}^{\text{arriv}}(k_d) - \alpha_{u,d,o}^{\text{leave}}(k_d)) c_d \quad (3)$$

where k_d is the time step counter (that is taken from the downstream node d), c_d is the cycle length of the downstream intersection d (note that we consider the simulation time length for link (u, d) to be equal to c_d), $\alpha_{u,d}^{\text{enter}}$ and $\alpha_{u,d}^{\text{leave}}$ are the entering and exiting average (within one cycle) flow rates according to link (u, d) , $\alpha_{u,d,o}^{\text{arriv}}$ and $\alpha_{u,d,o}^{\text{leave}}$ are the arriving and the leaving average flow rates of the sub-stream moving towards o , and $q_{u,d,o}$ is the queue length of the vehicles in link (u, d) that intend to move towards the node o . Note that although $n_{u,d}$ and $q_{u,d}$ are integer values in real situations, they are real non-negative values in the S-model. Moreover, we have:

$$\alpha_{u,d}^{\text{leave}}(k_d) = \sum_{o \in O_{u,d}} \alpha_{u,d,o}^{\text{leave}}(k_d) \quad (4)$$

$$\alpha_{u,d}^{\text{enter}}(k_d) = \sum_{i \in I_{u,d}} \alpha_{i,u,d}^{\text{leave}}(k_d) \quad (5)$$

with $O_{u,d}$ and $I_{u,d}$ the sets of output and input nodes of link (u, d) , and

$$\alpha_{u,d,o}^{\text{arriv}}(k_d) = \beta_{u,d,o}(k_d) \cdot \alpha_{u,d}^{\text{arrive}}(k_d) \quad (6)$$

$$\alpha_{u,d}^{\text{arrive}}(k_d) = \frac{c_d - \gamma(k_d)}{c_d} \cdot \alpha_{u,d}^{\text{enter}}(k_d - \delta(k_d)) + \frac{\gamma(k_d)}{c_d} \alpha_{u,d}^{\text{enter}}(k_d - \delta(k_d) - 1) \quad (7)$$

in which $\beta_{u,d,o}$ denotes the relative fraction of the vehicles within link (u, d) that turn to o , and $\alpha_{u,d}^{\text{arrive}}(k_d)$ is the average flow rate arriving at the end of the queue within link (u, d) at time step k_d , and the delays $\delta(k_d)$ and $\gamma(k_d)$ are derived by:

$$\delta(k_d) = \left\lfloor \frac{\tau(k_d)}{c_d} \right\rfloor \quad (8)$$

$$\gamma(k_d) = \text{rem}\{\tau(k_d), c_d\} \quad (9)$$

where $\lfloor x \rfloor$ and $\text{rem}\{x, y\}$ are the floor (the largest integer not greater than x) and the remainder (remainder of the division of x by y) functions, $\tau(k_d)$ is the average delay time for the vehicles that have entered the link (u, d) until they arrive at the tail of the downstream awaiting queue at k_d . Additionally,

$$\alpha_{u,d,o}^{\text{leave}}(k_d) = \min \left(\frac{\beta_{u,d,o}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o}(k_d)}{c_d}, \frac{q_{u,d,o}(k_d)}{c_d} + \alpha_{u,d,o}^{\text{arriv}}(k_d), \frac{\beta_{u,d,o}(k_d)(C_{d,o} - n_{d,o}(k_d))}{\sum_{u \in I_{d,o}} \beta_{u,d,o}(k_d) \cdot c_d} \right) \quad (10)$$

where $\mu_{u,d}$ is the saturated flow rate of link (u, d) , $g_{u,d,o}$ is the green time length for the traffic stream that leaves link (u, d) through o , $C_{d,o}$ is the storage capacity of link (d, o) . For extra details regarding the S-model see [11].

2) *Emission model*: VT-micro [12] is a microscopic model that estimates the instantaneous emissions for individual vehicles as a function of their instantaneous speed and acceleration. Suppose that vehicle i is moving with speed $v_i(k_d)$ and acceleration $a_i(k_d)$; Then the VT-micro model yields the estimated emission, $E_{\theta,i}$, of any emission type θ where $\theta \in \{\text{CO}, \text{NO}_x, \text{HC}\}$ produced by the vehicle i at time step k_d as:

$$E_{\theta,i}(v_i(k_d), a_i(k_d)) = \exp(\tilde{v}_i^\top(k_d) P_\theta \tilde{a}_i(k_d)) \quad (11)$$

where P_θ is a pre-calibrated matrix [12], and

$$\tilde{v}_i(k_d) = [1 \quad v_i(k_d) \quad v_i^2(k_d) \quad v_i^3(k_d)]^\top, \\ \tilde{a}_i(k_d) = [1 \quad a_i(k_d) \quad a_i^2(k_d) \quad a_i^3(k_d)]^\top$$

As we see the S-model (explained in Section III-A1) is a *macroscopic* flow model, while VT-micro includes a *microscopic* emission model. Shu et al. [10] integrate the S-model with the VT-micro model that results in a macroscopic integrated flow-emission model called the VT-S model. First, Shu et al. [10] define a set of behaviors for the vehicles that travel within the network, assuming that each behavior is uniform within sub-intervals of the total time spent by each vehicle in the network. Therefore, in these sub-intervals fixed values for acceleration of the vehicles could be considered. Then different traffic states are considered for the vehicles and based on these states the VT-S model makes an estimate of the instantaneous emissions (see [10] for more details).

3) *MPC formulation*: As we explained before, the optimization problem of MPC is solved online every control time step over the prediction horizon. For the controller design, we assume that for all intersections, the control time interval T_{ctrl} is the same. For all intersections $d \in J$, we assume that there is an integer N_d such that $T_{\text{ctrl}} = N_d \cdot c_d$, and thus $k_d = N_d \cdot k_{\text{ctrl}}$.

Then the MPC optimization problem at control step k_{ctrl} is formulated as follows:

$$\min_{\mathbf{g}(k_{\text{ctrl}})} \mathcal{J}(k_{\text{ctrl}}), \\ \text{s.t.} \\ \text{integrated VT-micro \& S-model,} \\ \mathcal{G}(\mathbf{g}(k_{\text{ctrl}})) = 0, \\ \mathbf{g}_{\min} \leq \mathbf{g}(k_{\text{ctrl}}) \leq \mathbf{g}_{\max} \quad (12)$$

where $\mathcal{J}(k_{\text{ctrl}})$ is the objective (cost) function, $\mathbf{g}(k_{\text{ctrl}})$ is a vector composed of all green phases to be optimized, and $\mathcal{G}(\mathbf{g}(k_{\text{ctrl}}))$ represents the equality constraints on the input vector, e.g., the summation of the green time lengths of different stages should adapt the cycle time of that intersection.

Here, we aim to find a balanced trade-off between reduction of the congestion (i.e., reduction of the total time spent (TTS) by the vehicles), and reduction of the total emissions (TE). Therefore, we have a multi-objective optimization problem, for which we define the objective function as a linear weighted

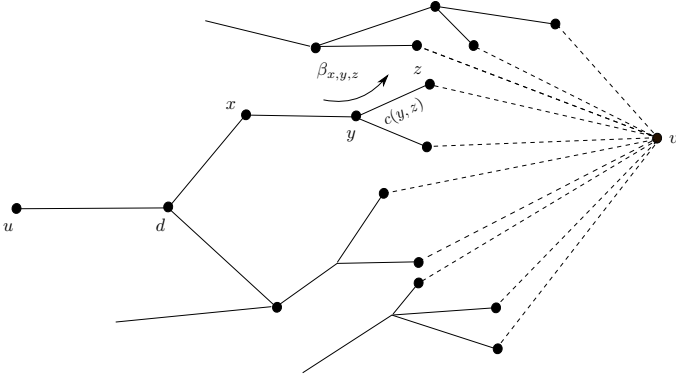


Fig. 3. Recasting the problem of finding the $K_{u,d}$ most likely used routes into a point-to-point K shortest path problem

combination of different objectives:

$$\begin{aligned} \mathcal{J}(k_{\text{ctrl}}) = & w_1 \frac{\text{TTS}(k_{\text{ctrl}})}{\text{TTS}_{\text{nominal}}} + \sum_{\theta \in \Theta} w_{2,\theta} \frac{\text{TE}_{\theta}(k_{\text{ctrl}})}{\text{TE}_{\theta,\text{nominal}}} + \\ & w_3 \frac{\text{TTS}_{\text{endpoint}}(k_{\text{ctrl}})}{\text{TTS}_{\text{endpoint,nominal}}} + \sum_{\theta \in \Theta} w_{4,\theta} \frac{\text{TE}_{\theta,\text{endpoint}}(k_{\text{ctrl}})}{\text{TE}_{\theta,\text{endpoint,nominal}}} + \\ & w_5 \frac{\text{var}(\mathbf{g}(k_{\text{ctrl}}))}{\text{var}_{\text{nominal}}} \end{aligned} \quad (13)$$

where $\text{var}(\mathbf{g}(k_{\text{ctrl}}))$ takes into account the variations in the control signal $\mathbf{g}(k_{\text{ctrl}})$, i.e.,

$$\text{var}(\mathbf{g}(k_{\text{ctrl}})) = \sum_{i=k_{\text{ctrl}}+1}^{k_{\text{ctrl}}+N_p} (\mathbf{g}(i) - \mathbf{g}(i-1))^2$$

Note that this term is added to the objective function to suppress oscillations of the control signal. Moreover, TTS and TE_{θ} denote the total time spent and the total estimated emission of $\theta \in \Theta = \{\text{CO}, \text{NO}_x, \text{HC}\}$ in the network during the prediction interval $[k_{\text{ctrl}}T_{\text{ctrl}}, (k_{\text{ctrl}} + N_p - 1)T_{\text{ctrl}})$, i.e.,

$$\begin{aligned} \text{TTS}(k_{\text{ctrl}}) &= \sum_{i=k_{\text{ctrl}}}^{k_{\text{ctrl}}+N_p-1} \text{TTS}(i), \\ \text{TE}_{\theta}(k_{\text{ctrl}}) &= \sum_{i=k_{\text{ctrl}}}^{k_{\text{ctrl}}+N_p-1} \text{TE}_{\theta}(i) \end{aligned}$$

with $\text{TTS}(i)$ and $\text{TE}_{\theta}(i)$ the total time spent and the total emissions during $[iT_{\text{ctrl}}, (i+1)T_{\text{ctrl}})$, and $\text{TTS}_{\text{nominal}}$ and $\text{TE}_{\theta,\text{nominal}}$ the nominal performances for TTS and TE_{θ} within the prediction time interval. The 3rd and the 4th terms in (13) account for the expected time spent and emissions resulted by the vehicles that have entered the network within the prediction interval and that are still in the network at the end of the prediction time interval, until they leave the network. Next we explain how these terms are computed.

4) Estimation of the expected time spent and emissions:

Suppose that we have a destination independent model, and the traffic situation is fixed after the prediction interval, i.e., all parameters are fixed at their values at $(k_{\text{ctrl}} + N_p)T_{\text{ctrl}}$. Now for a given link (u, d) , if we consider all the possible routes to the endpoints of the network that are reachable via (u, d) , for some

networks such as grid-shaped networks vehicles might move within cyclic paths. Then the number of the possible routes will become infinity. To prevent this situation, we determine a limited number $K_{u,d}$ of the most likely used routes from the link (u, d) to the endpoints of the network. The aim is to use an existing shortest path algorithm, for example the Yen's K shortest path routing algorithm [13].

A K shortest path algorithm seeks for the shortest paths that are extended between a pair of points a and b . Therefore, we first transform the problem into a point-to-point problem by connecting all the endpoints of the network to a single virtual endpoint "v" (see Figure 3), so that the problem reduces to finding the $K_{u,d}$ shortest routes that connect d and v . A route R_j between d and v is defined as:

$$R_j(d) = \{(d, d_{j,1}), (d_{j,1}, d_{j,2}), \dots, (d_{j,n_j-1}, v)\} \quad (14)$$

with n_j the number of links in route $R_j(d)$. For every pair of links $(x, y), (y, z) \in L$, with L the set of all links in the network, we define the endpoint turning rates as:

$$\beta_{\text{endpoint},x,y,z} = \beta_{x,y,z}(k_{\text{ctrl}} + N_p - 1) \quad (15)$$

In our problem, we look for the $K_{u,d}$ most likely used routes from (u, d) to v , i.e., the routes with the largest value of

$$\prod_{(x,y),(y,z) \in (R_j(d) \cup \{(u,d)\})} \beta_{\text{endpoint},x,y,z}$$

for $j \in \{1, 2, \dots, N_j(d)\}$, where $N_j(d)$ is the number of all possible (without cyclic paths) routes from d to v , and this is equivalent to finding the largest value of

$$\log \left(\prod_{(x,y),(y,z) \in (R_j(d) \cup \{(u,d)\})} \beta_{\text{endpoint},x,y,z} \right)$$

As Yen's algorithm [13] seeks for the minimized summation of the costs, hence our problem could be reformulated as finding the least $\sum_{(x,y),(y,z) \in (R_j(d) \cup \{(u,d)\})} (-\log \beta_{x,y,z})$, i.e., we look for the $K_{u,d}$ shortest routes where the costs $C(y, z)$ of the links are redefined as (see Figure 3):

$$C(y, z) = -\log \beta_{x,y,z} \quad (16)$$

Note that (16) is a legitimate definition for the cost, as we have $0 \leq \beta_{x,y,z} \leq 1$ and hence $C(y, z) \geq 0$.

After the $K_{u,d}$ shortest routes from the current link (u, d) to v are found, we put them in a set called $\mathcal{R}_{u,d,K_{u,d}}$. Moreover, the summation of the turning rates towards all the selected routes from node d should be unity. Indeed, by considering only $K_{u,d}$ routes, we assume that the turning rates towards the other routes are zero. Therefore, we need to define $\gamma_{u,d,r}$ for $r \in \{1, 2, \dots, K_{u,d}\}$, where $\gamma_{u,d,r}$ indicates the percentage of the vehicles that are within link (u, d) at the end of the prediction interval and that tend to travel the r^{th} route. Then $\gamma_{u,d,r}$ for $r \in \{1, 2, \dots, K_{u,d}\}$ is defined as:

$$\gamma_{u,d,r} = \frac{\prod_{(x,y),(y,z) \in (R_r(d) \cup \{(u,d)\})} \beta_{\text{endpoint},x,y,z}}{\sum_{l=1}^{K_{u,d}} \left(\prod_{(x,y),(y,z) \in (R_l(d) \cup \{(u,d)\})} \beta_{\text{endpoint},x,y,z} \right)} \quad (17)$$

Now that the least costly routes are determined, we can compute $\text{TTS}_{\text{endpoint}}$ and $\text{TE}_{\theta, \text{endpoint}}$ in (13):

$$\text{TTS}_{\text{endpoint}}(k_{\text{ctrl}}) = \sum_{(u,d) \in L} \left(n_{\text{endpoint}, u, d} \sum_{r=1}^{K_{u,d}} \gamma_{u,d,r} \text{TTS}_{u,d,r} \right) \quad (18)$$

$$\text{TE}_{\theta, \text{endpoint}}(k_{\text{ctrl}}) = \sum_{(u,d) \in L} \left(n_{\text{endpoint}, u, d} \sum_{r=1}^{K_{u,d}} \left(\gamma_{u,d,r} \sum_{\theta \in \Theta} \text{TE}_{\theta, u, d, r} \right) \right) \quad (19)$$

where $\text{TTS}_{u,d,r}$ and $\text{TE}_{\theta, u, d, r}$ are the total time spent and the total emissions of θ by each vehicle remaining on link (u, d) at the end of the prediction interval along route r , and

$$n_{\text{endpoint}, u, d} = n_{u,d}(k_{\text{ctrl}} + N_p - 1)$$

IV. SMOOTHING METHODS FOR THE S-MODEL

The simplified S-model explained in Section III-A1 includes two equations with non-smooth functions, i.e., the *floor* and the *min* functions in (8) and (10). A non-smooth model does not allow to use available efficient gradient-based optimization methods for MPC. Hence, it would be useful to have a nonlinear *smooth* model, expressed by means of functions that are differentiable everywhere in their domain. In this section, we propose a method to make these functions smooth.

A. Smooth form of the “min” Function

In (10), we should deal with:

$$\alpha = \min\{x, y, z\} \quad (20)$$

Therefore, we look for the smooth form of the following function in general:

$$x_{\min} = \min\{x_1, x_2, \dots, x_n\} \quad (21)$$

Our smoothing approach is inspired by the method proposed in [14], where the following smooth form of the plus function, i.e., $x^+ = \max\{0, x\}$, is proposed:

$$x^+ = \max\{0, x\} \approx x + \frac{1}{\alpha} \log(1 + e^{-\alpha x}) \quad \text{for } \alpha \gg 1 \quad (22)$$

We start by considering $\max\{x_1, x_2\}$. We have:

$$\begin{aligned} \max\{x_1, x_2\} &= \frac{1}{2} (\max\{0, x_2 - x_1\} + x_1 \\ &\quad + \max\{0, x_1 - x_2\} + x_2) \\ &\approx \frac{1}{2} \left(x_2 + \frac{1}{\alpha} \log(1 + e^{-\alpha(x_2 - x_1)}) \right. \\ &\quad \left. + x_1 + \frac{1}{\alpha} \log(1 + e^{-\alpha(x_1 - x_2)}) \right) \\ &\approx \frac{1}{2\alpha} \log \left(e^{\alpha x_2} (1 + e^{-\alpha(x_2 - x_1)}) \right. \\ &\quad \left. e^{\alpha x_1} (1 + e^{-\alpha(x_1 - x_2)}) \right) \\ &\approx \frac{1}{2\alpha} \log \left((e^{\alpha x_2} + e^{\alpha x_1}) \cdot \right. \\ &\quad \left. (e^{\alpha x_1} + e^{\alpha x_2}) \right) \\ &\approx \frac{1}{\alpha} \log(e^{\alpha x_1} + e^{\alpha x_2}) \end{aligned} \quad (23)$$

In general, we can write:

$$\max\{x_1, \dots, x_n\} = \frac{1}{\alpha} \log \left(\sum_{i=1}^n e^{\alpha x_i} \right) \quad (24)$$

Note that (22) is a specific case for $n = 2$ and $x_1 = 0$ and $x_2 = x$ in (24), where we just need to substitute x on the right-hand side of (22) with $\frac{1}{\alpha} \log e^{\alpha x}$.

Now, in order to obtain a general form for the *min* function given by (21), we write:

$$\begin{aligned} \min\{x_1, \dots, x_n\} &= -\max\{-x_1, \dots, -x_n\} \\ &= -\frac{1}{\alpha} \log \left(\sum_{i=1}^n e^{-\alpha x_i} \right) \end{aligned} \quad (25)$$

B. Smooth form of the floor function

Eq. (8) in Section III-A1 involves the floor function, which is non-smooth. The floor function has a piece-wise step shape (see the dashed plot in Figure 4). In [14] the step function is approximated by a sigmoid function:

$$S(x, \alpha) = \frac{1}{1 + e^{-\alpha x}} \quad (26)$$

Here we can use a similar idea by considering a piece of the transformed sigmoid function in each interval $[i, i + 1)$, where $i \in \mathbb{Z}$, as the smooth approximation of the floor function in this interval (see the solid plot in Figure 4).

We are looking for a smooth function by connecting the right endpoint of the sigmoid function (Figure 4) to the left endpoint point of a piece of a transformed sigmoid function that starts at the end point of the initial sigmoid piece. Therefore, we need to make sure that the slopes are equal at the end point and at the beginning point (the points that should be attached) of the two pieces. The slope of the sigmoid function at any positive real value x_0 equals the slope at $-x_0$, because:

$$\left. \frac{d}{dx} (1 + e^{-\alpha x})^{-1} \right|_{x=x_0} = \frac{\alpha e^{-\alpha x_0}}{(1 + e^{-\alpha x_0})^2} \quad (27)$$

$$\left. \frac{d}{dx} (1 + e^{-\alpha x})^{-1} \right|_{x=-x_0} = \frac{\alpha e^{\alpha x_0}}{(1 + e^{\alpha x_0})^2} \quad (28)$$

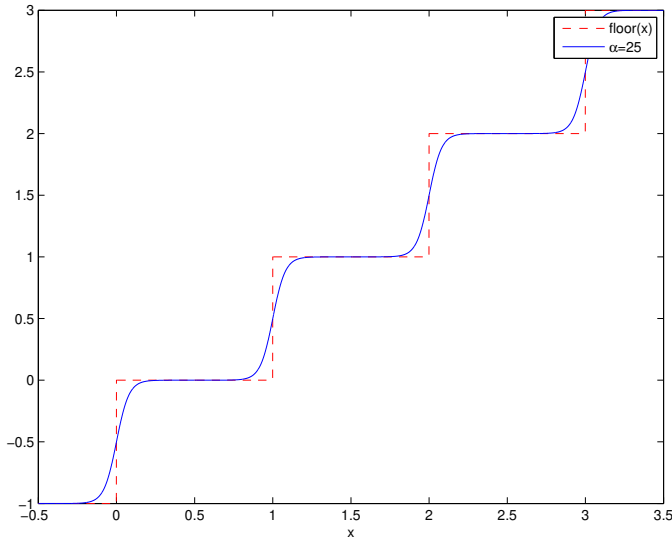


Fig. 4. The floor function (dashed) and its smooth form (solid) using the sigmoid function $S(x, \alpha) = \frac{1}{1+e^{-\alpha x}}$ with $\alpha = 25$

and if we multiply the numerator and the denominator of (27) by $(e^{\alpha x_0})^2$, (28) is obtained. Finally, the resulting smooth function for the floor function is mathematically expressed by:

$$\lfloor x \rfloor \approx \sum_{k \in \mathbb{Z}} \frac{1}{1 + e^{-\alpha(x-k)}}, \quad (29)$$

Now we can use the available efficient gradient-based algorithms to solve the MPC optimization problem. We can either solve the optimization problem given by (12) directly, or we can use the Pontryagin's principle [15], which translates the optimization problem into a two-point boundary-value problems (TPBVP) to be solved.

V. CONCLUSIONS AND FUTURE WORK

We have proposed a model predictive controller that aims to find a balanced trade-off between reduction of congestion and emissions in urban traffic networks. A macroscopic integrated flow-emission model known as the VT-S model has been used as the prediction model of the MPC. The integrated model involves a macroscopic flow model, i.e., the S-model, and a microscopic emission model, i.e., VT-micro. Since the S-model includes non-smooth functions, we cannot use the available efficient gradient-based methods to solve the optimization problem of the MPC-based controller. Therefore, in this paper we have proposed smoothing methods to make the flow model smooth.

The proposed MPC-based controller solves an optimization problem that minimizes a weighted combination of the total time spent and emissions by the vehicles, and the estimated values of the time spent and emissions caused by the vehicles that will stay in the network at the end of the prediction horizon (i.e., endpoint penalties). We have proposed an approach based on a K Shortest path algorithm that approximates the expected time spent and emissions by the remaining vehicles.

For future work, the MPC controller proposed in this paper will be applied to an urban traffic subnetwork, which will be

modeled by microsimulation, and the effect of implementing the designed MPC controller on reduction of congestion and emissions will be evaluated.

ACKNOWLEDGMENT

This research has been supported by the NWO-NSFC project "Multi-level predictive traffic control for large-scale urban networks" (629.001.011), which is partly financed by the Netherlands Organization for Scientific Research (NWO).

REFERENCES

- [1] C. Diakaki, M. Papageorgiou, and K. Aboudolas, "A multi-variable regulator approach to traffic-responsive network-wide signal control," *Control Engineering Practice*, vol. 10, no. 2, pp. 183–195, 2002.
- [2] 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.
- [3] 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.
- [4] M. van den Berg, A. Hegyi, B. De Schutter, and J. Hellendoorn, "Integrated traffic control for mixed urban and freeway networks: A model predictive control approach," *European Journal of Transport and Infrastructure Research*, vol. 7, no. 3, pp. 223–250, 2007.
- [5] T. Bellekens, 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.
- [6] J. Maciejowski, *Predictive Control with Constraints*. Prentice Hall, London, UK, 2002.
- [7] A. Stevanovic, J. Stevanovic, K. Zhang, and S. Batterman, "Optimizing traffic control to reduce fuel consumption and vehicular emissions," *Transportation Research Record*, vol. 2128, pp. 105–113, 2009.
- [8] S. Zegeye, B. De Schutter, J. Hellendoorn, and E. Breunese, "Reduction of area-wide emissions using an efficient model-based traffic control strategy," in *Proceedings of the 2011 IEEE Forum on Integrated and Sustainable Transportation Systems (FISTS 2011)*, (Vienna, Austria), pp. 307–312, June–July 2011.
- [9] T. Y. Liao and R. Machemehl, "Optimal traffic signal strategy for fuel consumption and emissions control at signalised intersections," in *Proceedings of the 24th European Transport Forum*, (London), Sept. 1996.
- [10] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn, "Integrated urban traffic control for the reduction of travel delays and emissions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 1609–1619, 2013.
- [11] 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.
- [12] 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.
- [13] J. Yen, "Finding the K shortest loopless path in a network," *Management Science*, vol. 17, no. 11, pp. 712–716, 1971.
- [14] 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.
- [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.