Delft University of Technology

Delft Center for Systems and Control

Technical report 16-021

Gradient-based model-predictive control for green urban mobility in traffic networks*

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

If you want to cite this report, please use the following reference instead:
A. Jamshidnejad, I. Papamichail, H. Hellendoorn, M. Papageorgiou, and B. De Schutter, "Gradient-based model-predictive control for green urban mobility in traffic networks," *Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Brazil, pp. 1077–1082, Nov. 2016. doi:10.1109/ITSC.2016.7795690

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

Gradient-based model-predictive control for green urban mobility in traffic networks

Anahita Jamshidnejad[a], Ioannis Papamichail[b], Hans Hellendoorn[a], Markos Papageorgiou[b], Bart De Schutter[a]

[a]Delft 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] Technical University of Crete,
Dynamic Systems & Simulation Laboratory,
73100 Chania, Greece
Email: {ipapa,markos}@dssl.tuc.gr

***Abstract*—To deal with the traffic congestion and emissions, traffic-responsive control approaches can be used. The main aim of the control is then to use the existing capacity of the network efficiently, and to reduce the harmful economical and environmental effects of heavy traffic. In this paper, we design a highly efficient model-predictive control system that uses a gradient-based approach to solve the optimization problem, which has been reformulated as a two-point boundary value problem. A gradient-based approach computes the derivatives to find the optimal value. Therefore, the optimization problem should involve only smooth functions. Hence, for nonsmooth functions that may appear in the internal model of the MPC controller, we propose smoothening approaches. The controller then uses an integrated smooth flow and emission model, where the control objective is to reduce a weighted combination of the total time spent and total emissions of the vehicles. We perform simulations to compare the efficiency and the CPU time of the smooth and nonsmooth optimization approaches. The simulation results show that the smooth approach significantly outperforms the nonsmooth one both in the CPU time and in the optimal objective value.**

I. Introduction

In urban areas, especially in capital cities, we should deal with economical, environmental, and health problems resulted by heavy traffic. In order to solve these issues, various control approaches have been proposed in the literature (see [1]–[5]), where the focus of the control engineers is on developing methods that use the existing capacity of the network in a more efficient way.

Model-predictive control or MPC [6] is an optimization-based control approach that has proven to be efficient in handling input and state constraints for various applications. MPC minimizes a cost function along a finite horizon and finds a suboptimal control strategy along the horizon (see Figure 1). To reduce the computation time, a control horizon of a smaller length than the prediction horizon may be considered. This imposes an additional constraint on the MPC optimization problem, i.e., the control signal remains constant from the end of the control horizon to the end of the prediction horizon (see Figure 1). MPC uses a prediction model to estimate the future states of the system and to use them for solving the optimization problem. The suboptimal control signal is implemented for one time step, and then the prediction window is shifted forward for one time step (see Figure 1). The controller receives the measured states from the system and uses them as the initial states of the optimization problem to solve the problem again.

In this paper a highly efficient MPC controller for urban traffic networks is designed. The controller finds a balanced trade-off between reduction of the total time spent and emissions. An integrated urban flow-emission model is used that combines the smooth modified S-model (see [7] and [8]) and VERSIT+ [9]. We apply a gradient-based method based on the minimum principle of Pontryagin [10] to solve the MPC optimization problem. In particular, to find the suboptimal solution of the MPC problem, we apply the resilient back-propagation (RProp) proposed by Riedmiller et al. [11].

In Section II, we formulate the optimization problem of the MPC. Section III introduces the internal model of the controller including the S-model and VERSIT+. We then show how to render these models smooth to apply a gradient-based optimization approach. Section IV explains the gradient-based optimization method, which uses RProp. In Section V, the simulation results are given for a case study. Finally, Section VI concludes the work and topics for future work are discussed.

II. Formulation of the urban traffic MPC controller

In this section, we represent the formulation of the optimization problem of the MPC controller and the selected objective function. The designed MPC controller performs in the discrete-time domain, with $k_{\rm ctrl}$ denoting the control time step, $T_{\rm ctrl}$ the control sampling time, k_d the simulation time step for intersection d, and T_d the simulation sampling time for intersection d. Note that, in general, T_d and $T_{\rm ctrl}$ may not be equal. Hence, the internal model of the MPC controller estimates the states corresponding to the downstream intersection d at time steps k_d, which might be different from the time steps $k_{\rm ctrl}$, at which the controller should make a decision about the best

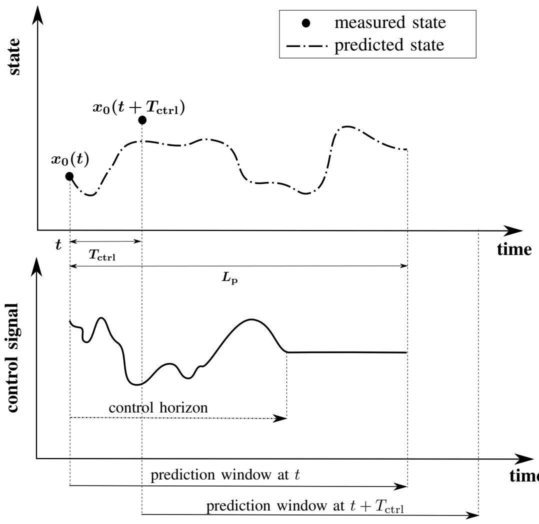


Figure 1. MPC in the continuous-time domain.

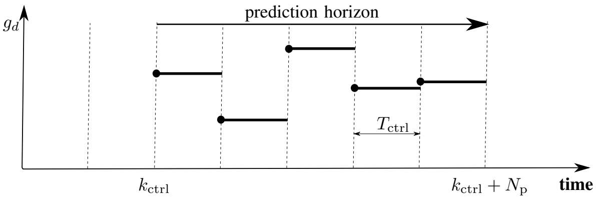


Figure 2. Piecewise constant suboptimal control signal within one prediction interval.

control strategy. There may also be an offset between k_d and k_{ctrl} (this will be discussed in more detail later in Section II-A).

Suppose that at time step k_{ctrl} the controller should make a decision about the control signals of the intersections within the network. Then the controller should minimize an objective function $\mathcal{J}(\cdot)$ and find the suboptimal control sequence $\boldsymbol{g}$ for the entire prediction horizon. Assume that the prediction horizon is of length $L_{\mathrm{p}} = N_{\mathrm{p}} T_{\mathrm{ctrl}}$, i.e., it includes N_{p} control time steps. Since the MPC controller performs in the discrete-time domain, the optimization problem computes the optimal control signal of each intersection d at discrete time steps k_{ctrl}, $k_{\mathrm{ctrl}}+1$, ..., $k_{\mathrm{ctrl}}+N_{\mathrm{p}}-1$ along the prediction horizon (see Figure 2). We consider to have a piecewise constant suboptimal control signal within $[k_{\mathrm{ctrl}} T_{\mathrm{ctrl}}, (k_{\mathrm{ctrl}}+N_{\mathrm{p}})T_{\mathrm{ctrl}})$, i.e. the control signal remains the same within $[kT_{\mathrm{ctrl}}, (k+1)T_{\mathrm{ctrl}})$ for $k \in \{k_{\mathrm{ctrl}}, \ldots, (k_{\mathrm{ctrl}}+N_{\mathrm{p}}-1)T_{\mathrm{ctrl}}\}$.

Therefore, for intersection d, a vector $\boldsymbol{g}_d$ including all variables that should be optimized, i.e., $[g_d(k_{\mathrm{ctrl}}) \quad \ldots \quad g_d(k_{\mathrm{ctrl}}+N_{\mathrm{p}}-1)]^\top$ is considered. Since we should solve the optimization problem for the entire network, the optimization variable $\boldsymbol{g}$ will be a vector that includes the entries of all vectors $\boldsymbol{g}_d$ for all intersections d within the traffic network. Then we can formulate the optimization problem of the MPC controller at time step k_{ctrl} as

$$
\begin{aligned}
&\min_{\boldsymbol{g}(k_{\mathrm{ctrl}})} \mathcal{J}(k_{\mathrm{ctrl}}),\\
&\quad\text{such that}\\
&\quad\textbf{C1.}\ \text{Integrated flow-emission model estimates}\\
&\qquad\ \text{dynamics of the urban network,}\\
&\quad\textbf{C2.}\ \mathcal{G}_{\mathrm{eq}}(\boldsymbol{g}(k_{\mathrm{ctrl}})) = 0,\\
&\quad\textbf{C3.}\ \mathcal{G}_{\mathrm{ineq}}(\boldsymbol{g}(k_{\mathrm{ctrl}})) < 0,
\end{aligned}
\tag{1}
$$

where $\mathcal{G}_{\mathrm{eq}}(\cdot)$ and $\mathcal{G}_{\mathrm{ineq}}(\cdot)$ are functions that operate on the optimization variable and expressing the equality and inequality constraints of the optimization problem.

As we mentioned before, the aim of the designed MPC controller is to find a balanced trade-off between reduction of the total time spent by the vehicles and reduction of the total emissions. Therefore, we define the objective function as a weighted combination of the total time spent (divided by its typical order of the magnitude) and the total emissions of CO, HC, and $\mathrm{NO_x}$ (divided by their typical order of the magnitudes). Note that we consider CO, HC, and $\mathrm{NO_x}$ as an example for the pollutant types we want to reduce. We can write

$$
\begin{aligned}
\mathcal{J}(k_{\mathrm{ctrl}}) = {} & w_{\mathrm{T}} \sum_{k\in\mathbb{T}} \frac{\mathrm{TTS}(k)}{\mathrm{TTS_t}} + w_{\mathrm{CO}} \sum_{k\in\mathbb{T}} \frac{\mathrm{TE_{CO}}(k)}{\mathrm{TE_{CO,t}}}\\
& + w_{\mathrm{HC}} \sum_{k\in\mathbb{T}} \frac{\mathrm{TE_{HC}}(k)}{\mathrm{TE_{HC,t}}} + w_{\mathrm{NO_x}} \sum_{k\in\mathbb{T}} \frac{\mathrm{TE_{NO_x}}(k)}{\mathrm{TE_{NO_x,t}}},
\end{aligned}
\tag{2}
$$

where $\mathbb{T} = \{k_{\mathrm{ctrl}}, k_{\mathrm{ctrl}}+1, \ldots, k_{\mathrm{ctrl}}+N_{\mathrm{p}}-1\}$, and $\mathrm{TTS}(\cdot)$ and $\mathrm{TE}_e(\cdot)$ are functions that give the total time spent by the vehicles and the total emissions of the pollutant e. The subscript "t" indicates the typical order of the magnitude for the given function.

A. Synchronization of the simulation model and the MPC controller

In order to compute the total time spent and the total emissions of the vehicles in (2), we need the internal model of the MPC controller to give us an estimate of the network's state (including the number of the vehicles within the network), and the corresponding driving behaviors during the prediction interval. The integrated flow and emission model can provide us with this information. An extensive discussion regarding the integrated flow and emission model used in this paper is given in [8], where the model first detects the governing traffic scenario (under-saturated, saturated, over-saturated) within each link of the network, and then computes the number of vehicles that move with different traffic behaviors (e.g., uniform speed, accelerating, idling, etc.).

The internal model of the MPC controller estimates the network's state at every simulation time step (e.g., at k_d for intersection d). As we explained in Section II, since the simulation sampling time and the control sampling time are not necessarily equal, then k_d and k_{ctrl} also indicate different time instants. However, since the total time spent and the total emissions should be computed by the MPC controller,

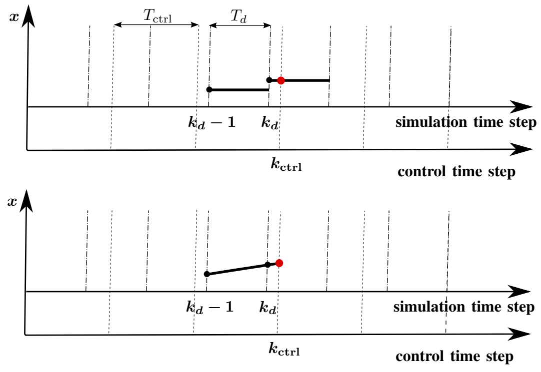


Figure 3. Two options for computing the states at control time steps based on the values at simulation time steps.

it needs to know the estimated state values of the network at k_{ctrl}. Therefore, we propose two options to synchronize the simulation model and the MPC controller, i.e.,

1) Assume that $\boldsymbol{x}(\cdot)$ is piecewise constant within the interval $[k_d T_d, (k_d+1)T_d)$ (see the top plot in Figure 3). Hence, from the following relation between k_d and k_{ctrl},

$$k_d(k_{\text{ctrl}}) = \left\lfloor \frac{k_{\text{ctrl}} T_{\text{ctrl}}}{T_d} \right\rfloor, \tag{3}$$

we can substitute $\boldsymbol{x}(k_{\text{ctrl}})$ by $\boldsymbol{x}(k_d(k_{\text{ctrl}}))$.

2) The second option is to consider a linear extrapolation approximation for the states based on the current and the previous values of the state estimated by the internal flow model of the MPC controller (see the bottom plot in Figure 3), i.e.,

$$\boldsymbol{x}(k_{\text{ctrl}}) = \boldsymbol{x}(k_d(k_{\text{ctrl}})) + \left(k_{\text{ctrl}} \frac{T_{\text{ctrl}}}{T_d} - k_d(k_{\text{ctrl}})\right) \cdot (\boldsymbol{x}(k_d(k_{\text{ctrl}})) - \boldsymbol{x}(k_d(k_{\text{ctrl}}) - 1)) \tag{4}$$

where $k_d(k_{\text{ctrl}})$ is found by (3).

III. Flow and emission models

For the internal flow and emission models of the MPC controller, we use, respectively, the S-model [7] and the VERSIT+ [9]. Next, we briefly discuss the original models, and we also show how to render these models smooth.

A. Nonsmooth S-model

The S-model is a nonlinear and nonsmooth discrete-time urban traffic flow model that was introduced by Lin et al. in [7]. The simulation sampling time of the S-model might differ for various links, and it is considered to be equal to the cycle time T_d of the downstream intersection d of a link (u,d) (with u the upstream intersection of that link). The state vector $\boldsymbol{x}_{u,d}(k_d)$ of link (u,d) at time step k_d, includes $x^{\text{n}}_{u,d}(k_d)$, which is the total number of the vehicles observed on link (u,d) during the time interval $[k_d T_d, (k_d+1)T_d)$, and $x^{\text{q}_o}_{u,d}(k_d)$ (for all[1] $o \in \mathcal{O}_{u,d}$), which is the total number of the vehicles that are idling in a queue on link (u,d) within the mentioned time interval, and that intend to move towards the subsequent intersection o. Then, the state vector of link (u,d) is updated at every simulation time step by

$$\boldsymbol{x}_{u,d}(k_d+1) = \boldsymbol{x}_{u,d}(k_d) + \boldsymbol{a} T_d, \tag{5}$$

with

$$\boldsymbol{a} = \begin{bmatrix} \alpha^{\text{enter}}_{u,d}(k_d) - \alpha^{\text{leave}}_{u,d}(k_d) \\ \alpha^{\text{arrive},\text{q}_{o_1}}_{u,d}(k_d) - \alpha^{\text{leave, q}_{o_1}}_{u,d}(k_d) \\ \vdots \\ \alpha^{\text{arrive},\text{q}_{o_N}}_{u,d}(k_d) - \alpha^{\text{leave, q}_{o_N}}_{u,d}(k_d) \end{bmatrix},$$

with N the number of entries within $\mathcal{O}_{u,d}$, $\alpha^{\text{enter}}_{u,d}(k_d)$ and $\alpha^{\text{leave}}_{u,d}(k_d)$ the total entering and leaving flow rates of link (u,d) during the time interval $[k_d T_d, k_d T_d + 1)$, and $\alpha^{\text{arrive},\text{q}_{o_i}}_{u,d}(k_d)$ and $\alpha^{\text{leave, q}_{o_i}}_{u,d}(k_d)$ the arriving and the leaving flow rates within the mentioned time interval for the queue of vehicles that intend to move towards o_i.

B. Smooth S-model

The S-model includes minimum function, which is nonsmooth. Hence, the resulting optimization problem is also nonsmooth, and prevents us from implementing gradient-based optimization approaches for solving it. We propose the following smooth form:

$$\min\{x_1, x_2, x_3\} \approx \frac{1}{4}\left(x_1 + x_2 + 2x_3 - \sqrt{(x_1 - x_2)^2 + \alpha}\right) - \sqrt{\frac{1}{4}\left(x_1 + x_2 - 2x_3 - \sqrt{(x_1 - x_2)^2 + \alpha}\right)^2 + \alpha}. \tag{6}$$

Here is the reason of proposing (6); suppose that we have $x_1 \leq x_2$, then

$$x_1 - x_2 = -|x_1 - x_2|,$$

and also

$$\min\{x_1, x_2\} = x_1 = \frac{1}{2}\left(x_1 + x_2 + (x_1 - x_2)\right),$$

which results in

$$\min\{x_1, x_2\} = \frac{1}{2}\left(x_1 + x_2 - |x_1 - x_2|\right). \tag{7}$$

We now approximate the nonsmooth function $|\cdot|$ by a smooth expression. We propose to use

$$|x| \approx \sqrt{x^2 + \alpha},$$

with α the smoothening parameter. Note that when $\alpha \to 0$, the right-hand side of the equation becomes close to $|x|$, and the sharpness of the corresponding graph increases, i.e., the level of smoothness decreases. From (7), we have

$$\min\{x_1, x_2\} \approx \frac{1}{2}\left(x_1 + x_2 - \sqrt{(x_1 - x_2)^2 + \alpha}\right). \tag{8}$$

Finally, by implementing (8) twice, we obtain (6).

[1] $\mathcal{O}_{u,d}$ is a set that includes all the subsequent intersections that are connected to link (u,d) via intersection d.

C. Emission model VERSIT+

VERSIT+ [9] is a microscopic emission model that gives the instantaneous emissions, in gram per second, for different categories of the speed range that is typical for the Netherlands. The instantaneous emissions E_e of the pollutant e for each individual vehicle are computed as follows (note that the units for speed and acceleration are $\mathrm{m/s}$ and $\mathrm{m/s^2}$):

- for $v < 1.4$ and $a < 0.5$ (idling),
$$E_e = E_{0,e}; \tag{9}$$
- for $v < 14$ (no idling),
$$E_e = E_{1,e} + E_{2,e} \max\{D, 0\} + E_{3,e} \max\{D-1, 0\}; \tag{10}$$
- for $14 \leq v < 22.2$,
$$E_e = E_{4,e} + E_{5,e} \max\{D, 0\} + E_{6,e} \max\{D-1, 0\}; \tag{11}$$
- for $v > 22.2$,
$$E_e = E_{7,e} + E_{8,e} \max\{D-0.5, 0\} + E_{9,e} \max\{D-1.5, 0\}, \tag{12}$$

with $D = a + 0.004v$, v and a denote, respectively, the speed and the acceleration of the vehicle, and $E_{i,e},\ i = 1\ldots,9$ are fixed parameters for different pollutants e.

Note that from (8), we have
$$\min\{x, 0\} \approx \frac{1}{2}\left(x - \sqrt{x^2 + \alpha}\right). \tag{13}$$
We also know that
$$\max\{x, 0\} = -\min\{-x, 0\}. \tag{14}$$
Therefore, in the given equations for VERSIT+, we can substitute
$$\max\{D, 0\} \approx \frac{1}{2}\left(D + \sqrt{D^2 + \alpha}\right), \tag{15}$$
and obtain a smooth emission model as well.

IV. Solving the smooth optimization problem

To solve the smooth form of the optimization problem, we use the Pontryagin's minimum principle [12], which states that a necessary condition for a solution to be an optimal solution of the optimization problem is that it should minimize the corresponding Hamiltonian of the problem. The Hamiltonian of the optimization problem (1) with the objective function given by (2) is defined as
$$\begin{aligned} &\boldsymbol{H}(k, \boldsymbol{\lambda}(k+1), \boldsymbol{x}(k), \boldsymbol{g}(k)) = \\ &\quad \mathcal{J}(k) + \boldsymbol{\lambda}^\top(k+1) \cdot \boldsymbol{f}(k, \boldsymbol{x}(k), \boldsymbol{g}(k)), \end{aligned} \tag{16}$$
with $\boldsymbol{\lambda}(\cdot)$ the costate and $\boldsymbol{f}(\cdot)$ the model expression that computes the states of the system.

In order to find the control signal that minimizes (16), we use the resilient back-propagation (RProp) [11] approach. In this approach, the gradient of the Hamiltonian is computed at every optimization iteration, and is compared with its value from the previous iteration. If the gradient has not changed sign, it means that we are still on the right track to find a local optimum. Therefore, RProp decreases the absolute value of the increment of the optimization variable, while keeping its previous sign (see Figure IV).

However, if the gradient has changed sign, it means that we have jumped over a local optimum. Therefore, RProp increases the absolute value of the increment of the optimization variable, and it changes the sign of the increment w.r.t. the previous iteration (see Figure 4(b)).

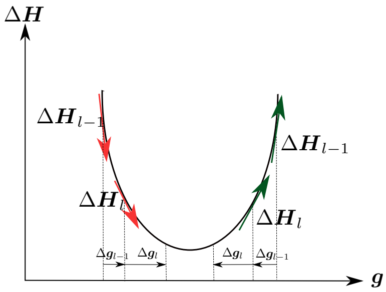
V. Results

In this section, we compare the CPU time and the efficiency (i.e., the value of the objective function) for implementing the gradient-based RProp approach for solving the smooth formulation of the MPC problem, and for implementing a genetic algorithm approach for solving the nonsmooth formulation of the problem.

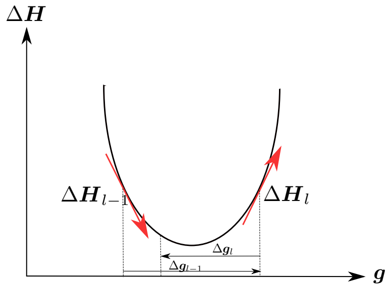
We consider the urban traffic network shown in Figure 5 for our case study. The network consists of four entrances, for which the corresponding entering flow rates are shown by $\alpha_1^{\text{enter}}, \ldots, \alpha_4^{\text{enter}}$. There are eleven one-way links within the network, where each of these links has only one lane and a length of 0.5 [km]. The parameters $\beta_1, \ldots, \beta_4$ indicate the turning rates of the vehicles at junctions, where $\beta_1 = \beta_3 = 60\%$ and $\beta_2 = \beta_4 = 40\%$.

There are three traffic lights in the network, where all these lights have the same cycle time of 60 [s]. The free-flow and the idling speed of the vehicles are respectively 14 [m/s] and 0.4 [m/s]. The average length of the vehicles in the network is 7 [m], and the acceleration and deceleration of the vehicles are 2 [m/s^2] and -2 [m/s^2]. The simulations will be implemented for three different demand profiles (the demand profiles are illustrated in Figure 6). Each simulation is run for 30 [min].

For implementing the gradient-based approach, we benefit from the minimum principle of Pontryagin (see Section IV, and for more details see [10], [12], [13]), and by defining the Hamiltonian of the optimization problem and setting the partial derivative of the Hamiltonian w.r.t. the optimization variable equal to zero. To find the control signals that satisfy this condition, we use RProp. The results of the simulations for both the smooth and the nonsmooth optimization are shown in Tables I (for RProp) and II (for genetic algorithm). From these results we see that the gradient-based (smooth) approach for solving the optimization problem significantly outperforms the genetic algorithm (nonsmooth) approach. The CPU time can decrease up to a factor of 50 using the gradient-based approach, and the optimal value of the objective function can be up to 3.5 times less for the gradient-based approach w.r.t. the nonsmooth approach. Therefore, based on these simulation results, we see that when the optimization problem of the MPC is formulated as a smooth problem, and is solved via a gradient-based approach, it is clearly more efficient than a nonsmooth approach.



(a) RProp when the optimization variable for the previous and current iterations are on the same side of the optimum.



(b) RProp when the optimization variable for the previous and current iterations are on different sides of the optimum.

Figure 4. The resilient back-propagation (RProp) algorithm (note that l denotes the iteration number).

Table I. SIMULATION RESULTS FOR THE GRADIENT-BASED (SMOOTH OPTIMIZATION) APPROACH

	CPU time [s]	Objective function	TTS [s]	$\mathrm{TE_{CO}}$ [kg]	$\mathrm{TE_{HC}}$ [kg]	$\mathrm{TE_{NO_x}}$ [kg]
Demand 1	6690.31	16.90	2.6165×10^5	4.735	0.306	0.528
Demand 2	1386.4	9.96	2.9032×10^5	5.307	0.341	0.594
Demand 3	1209.3	8.52	4.0235×10^5	3.035	0.323	0.367

Table II. SIMULATION RESULTS FOR THE GENETIC ALGORITHM (NONSMOOTH OPTIMIZATION) APPROACH

	CPU time [s]	Objective function	TTS [s]	$\mathrm{TE_{CO}}$ [kg]	$\mathrm{TE_{HC}}$ [kg]	$\mathrm{TE_{NO_x}}$ [kg]
Demand 1	57711.5	40.91	1.1894×10^6	22.247	1.432	2.456
Demand 3	10115.5	43.18	1.2592×10^6	22.968	1.479	2.565
Demand 2	9797.9	29.19	8.5275×10^5	15.364	1.003	1.691

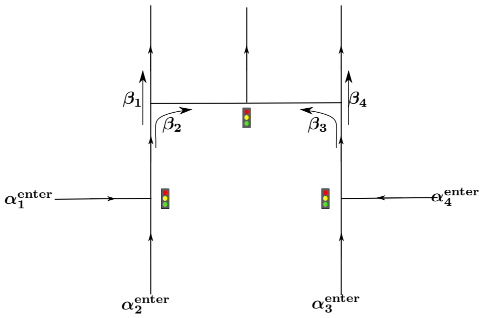


Figure 5. Urban traffic network with 4 entrances.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have designed an efficient model-predictive control system, where the control aim is to reduce the total time spent and the emissions in and urban traffic network. The controller uses a smooth integrated flow and emission model to estimate the future states and emissions of the network and to solve its optimization problem via a gradient-based approach that uses the resilient back-propagation (RProp) approach. In the paper, we have also presented the smooth form of the urban flow S-model.

Simulations have been implemented for an urban traffic network with four entrances and eleven links. For the simulations, we have used both the smooth MPC optimization formulation solved by the gradient-based RProp approach and the nonsmooth formulation of the MPC problem that has been solved via a genetic algorithm. The simulation results show a significant improvement in both the CPU time and the value of the objective function, when the optimization problem is solved via RProp compared with when the nonsmooth optimization formulation is considered.

Topics for future work include an extensive case study for different optimization approaches (smooth and nonsmooth) and comparing the corresponding CPU time and efficiency (i.e., the optimal value of the objective function). Moreover, we can use other urban flow and emission models (e.g., cell transmission model [14] and VT-micro [15]) for both the smooth and the nonsmooth formulation of the MPC optimization problem.

ACKNOWLEDGMENT

This research has been supported by the European COST Action TU1102 and 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). For the second and the fourth author, the research leading to these results has received funding from the European Commission under the European Union’s Seventh Framework Programme (FP/2007-2013) / FP7-ICT-2013.3.4, project LOCAL4GLOBAL (n. 611538)

REFERENCES

[1] 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.

[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.

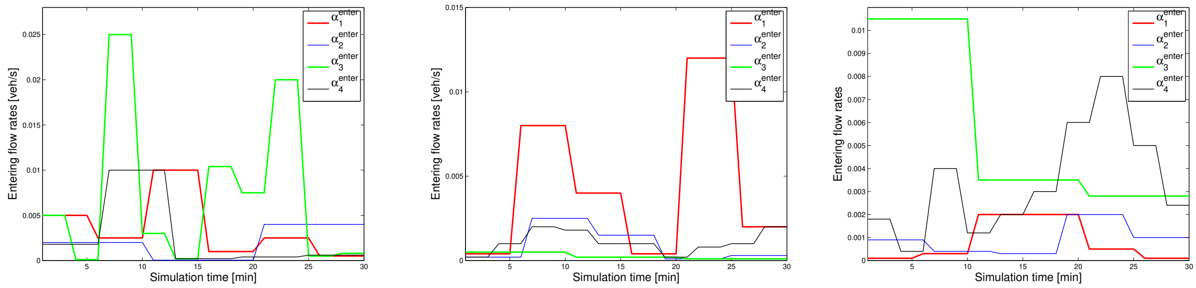


Figure 6. Demand profiles used for simulations.

[3] 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.

[4] 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.

[5] B. De Schutter, “Model predictive traffic control for green mobility,” in *Proceedings of the 2014 European Control Conference*, Strasbourg, France, June 2014, pp. 2260–2263.

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

[7] 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.

[8] A. Jamshidnejad, I. Papamichail, M. Papageorgiou, and B. De Schutter, “Model-predictive control for green mobility in urban traffic networks: Efficient solution based on general smoothening methods – addendum,” Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands, Tech. Rep. 15-033a, Dec. 2015.

[9] N. E. Ligterink, R. D. Lange, and E. Schoen, “Refined vehicle and driving-behaviour dependencies in the VERSIT+ emission model,” in *Proceedings of the ETAPP Symposium*, Toulouse, France, June 2009, pp. 177–186.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

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