

Technical report 16-017

A model-predictive urban traffic control approach with a modified flow model and endpoint penalties*

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, "A model-predictive urban traffic control approach with a modified flow model and endpoint penalties," *Proceedings of the 14th IFAC Symposium on Control in Transportation Systems (CTS 2016)*, Istanbul, Turkey, pp. 147–152, May 2016.

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.51.19 (secretary)
fax: +31-15-278.66.79
URL: <http://www.dcsc.tudelft.nl>

*This report can also be downloaded via http://pub.deschutter.info/abs/16_017.html

A Model-Predictive Urban Traffic Control Approach with a Modified Flow Model and Endpoint Penalties^{*}

Anahita Jamshidnejad^{*} Ioannis Papamichail^{**}
Markos Papageorgiou^{**} Bart De Schutter^{*}

^{*} *Delft Center for Systems and Control (DCSC), Delft University of Technology, the Netherlands*

^{**} *Dynamic Systems and Simulation Laboratory, Technical University of Crete, Greece*

Abstract: Nowadays, congestion caused by traffic in urban areas is considered as a major problem. In order to make the best use of the existing road capacity, traffic-responsive control systems, including model-predictive controllers, are excellent choices. A model-predictive controller can minimize a cost function along a given time horizon. We propose a model-predictive control system that aims to reduce the congestion, and uses an internal flow model, which is our proposed modified version of the S-model. In the formulation of the objective function for the controller, we take into account the effect of those vehicles that remain in the network at the end of the prediction horizon until the network is completely evacuated. We formulate this effect as endpoint penalties for the MPC optimization problem. Finally, we will apply the designed controller to an urban traffic network and compare two scenarios, i.e., the fixed-time control case and the model-predictive control approach with the endpoint penalties proposed in this paper. The results prove the excellent performance of the model-predictive controller compared with the fixed-time controller.

Keywords: Traffic control, Model-predictive control, Urban traffic flow model, Endpoint penalties.

1. INTRODUCTION

Traffic in urban areas is significantly time and energy consuming, especially when roads become highly congested. This waste of time and energy, and the inefficient consumption of fuel by vehicles while idling in congested queues, can be the source of huge economical and environmental costs. These negative effects have been investigated and assessed by Center for Economics and Business Research (2014) for the US, Germany, France, and UK. Based on their findings, it is expected that the economical costs of traffic congestion in the above-mentioned four countries will show an increase of 46% in 2030 compared with 2013. This indicates that it is very important, both from the economical and the environmental point-of-view, to take actions aimed at reducing congestion in urban traffic areas.

The main reason of congestion in urban traffic networks is inefficient use of the available roads (Thomson, 1997). Expanding the current road capacity requires long-term planning and can be highly costly. However, one solution that can be achieved in a relatively short time period is to make the best use of the existing road capacity by introducing more efficient traffic management and control systems.

^{*} This research has been supported by COST ARTS and by the NWO-NFSC 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).

Real-time traffic-responsive control systems are widely known for their high efficiency compared with other traffic management systems, and for their adaptability to dynamic changes in the system (Diakaki et al., 2002, 2003). One of the real-time control categories that has been implemented extensively for different systems, is the optimization-based control methods.

Model-predictive control (MPC), which is the main focus of this paper, is an optimization-based control approach. We choose MPC, since it is a suboptimal control approach that fits well our aim of finding a control strategy that manages traffic systems close to their optimal performance; in addition to that, MPC is indeed a dynamic optimization approach and its characteristics (such as handling constraints on the states and on the inputs, being an anticipative control method, making use of the measured states/output as feedback, etc.) fit the requirements of traffic networks with highly dynamic behaviors (Maciejowski, 2002).

Moreover, compared with optimal control strategies that use dynamic programming within an entirely open-loop scheme, the suboptimal solution computed by MPC is more robust towards external disturbances as MPC uses the measured states of the system at every time step.

In order to solve the optimization problem for MPC, a prediction horizon with a finite number of time steps is considered. A suboptimal control strategy is computed

along this prediction horizon, but only the signal corresponding to the first step of the horizon will be applied to the traffic network. Afterwards, the starting point of the horizon will be moved ahead for one time step, and the procedure will be repeated. A model-predictive controller should predict the future states of the system along the prediction horizon; therefore, a model of the system is considered by the controller. A detailed discussion on MPC can be found in (Maciejowski, 2002).

In this paper, we will focus on designing an urban traffic control system that makes use of the MPC approach and reduces the total time spent (TTS) by the vehicles in the network. We propose a new version of the S-model, which is an urban traffic flow model developed by Lin et al. (2012), where we introduce a new state, i.e., queues formed at the sources of the network. Moreover, we introduce some modifications to make the model more accurate, while it is still computationally very efficient. MPC has been used previously for both freeways and for urban traffic networks (Aboudolas et al., 2010, 2009; van den Berg et al., 2007; Burger et al., 2013; Bellemans et al., 2006; Lin et al., 2013; Hegyi et al., 2005). However, we formulate an objective function for the MPC optimization problem that takes into account the effect of those vehicles that will still be in the network by the end of the prediction interval. We also propose a computationally efficient algorithm for computing the endpoint penalties in real time.

The paper is organized as follows: in Section 2, we give a brief introduction to model-predictive control. In Section 3, we explain the formulation of the problem that we are going to solve with MPC in order to reduce the total time spent by vehicles, and consequently the congestion level, in the traffic network. Section 4 is about estimating the endpoint penalties for the objective function using shortest-path algorithms for networks with a large number of routes. We present the results of implementing the MPC-based controller with endpoint penalties in Section 5 for an urban traffic network. Finally, Section 6 concludes the paper.

2. MODEL-PREDICTIVE CONTROL (MPC)

Model-predictive control (MPC) is a real-time control approach that finds a suboptimal control signal for the controlled system through optimizing an objective function along a finite-length horizon (known as the prediction horizon). Figure 1 presents the main idea behind MPC for a discrete-time system, for which measurements are available at discrete time steps only.

MPC is a dynamic optimization approach, i.e., the controller receives the measured states of the system at every time step as feedback, and applies them as the initial conditions of its optimization problem. Therefore, MPC is a more robust control approach for systems that are exposed to external disturbances, compared with classical optimal control methods, which apply dynamic programming in an open-loop control scheme and solve the optimization problem offline.

An MPC controller receives the measured states of the system at each time step k_{ctrl} . However, the controller should estimate the optimal control strategy not only at

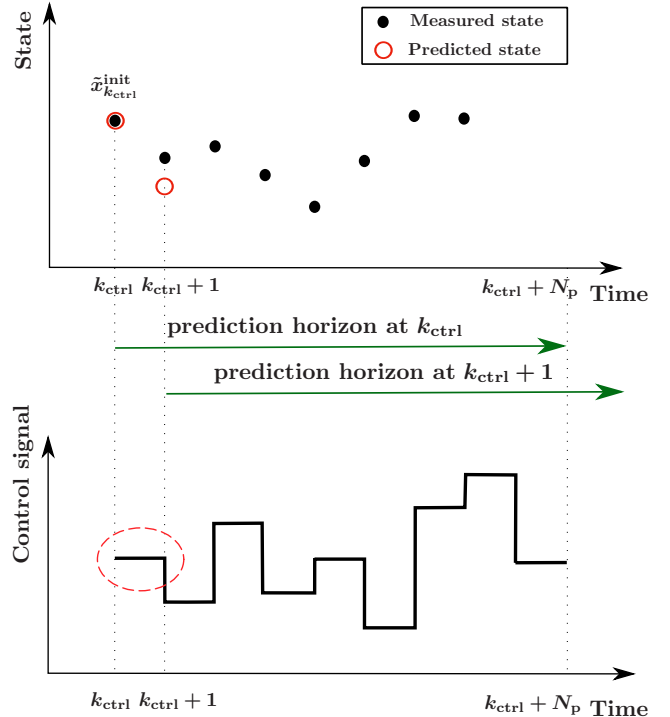


Fig. 1. Basic idea behind model-predictive control (MPC) where $\tilde{x}_{k_{\text{ctrl}}}^{\text{init}}$ shows the measured state variable at k_{ctrl} .

k_c , but also along the entire prediction horizon of length N_p . Therefore, it needs to have a model of the system that uses the initial measured values of the state to produce an estimation of the states at all time steps $k_{\text{ctrl}} + i$ for $i \in \{1, \dots, N_p\}$.

Since most model-predictive controllers are used in real time, the prediction model of the controller should provide a trade-off between accuracy and computational efficiency. When the model estimates the future states of the system, the controller minimizes a cost function that is formulated based on the current and the predicted states. The output of the MPC controller is a suboptimal control sequence within $[k_{\text{ctrl}}T_{\text{ctrl}}, (k_{\text{ctrl}} + N_p)T_{\text{ctrl}}]$, where T_{ctrl} denotes the length of the time step (see Figure 1).

The optimal control signal will be implemented for the real system only within $[k_{\text{ctrl}}T_{\text{ctrl}}, (k_{\text{ctrl}} + 1)T_{\text{ctrl}}]$. At time step $k_{\text{ctrl}} + 1$, the states of the system will be measured again and will be sent to the controller. The prediction horizon will be moved ahead along the time axis for one time step, and the procedure explained above will be repeated.

3. MPC FOR URBAN TRAFFIC

3.1 Formulating the Problem

We discuss a model-predictive control system to optimize the green time length of the intersections in an urban traffic network. Since the aim of the controller is to reduce the total time spent by the vehicles (and consequently the level of congestion) in the network, we propose a multi-objective optimization problem to be solved by the MPC controller. The objective function $\mathcal{J}(k_{\text{ctrl}})$ is formulated by:

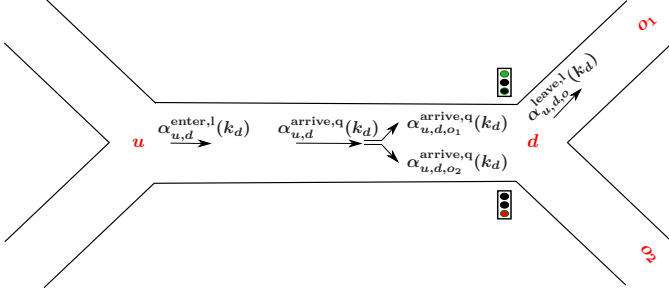


Fig. 2. Illustration of the entering, arriving, and leaving flows for the urban traffic flow model

$$\mathcal{J}(k_{\text{ctrl}}) = w_1 \frac{\text{TTS}(k_{\text{ctrl}})}{\text{TTS}^{\text{nominal}}} + w_2 \frac{\text{TTS}_{\text{endpoint}}(k_{\text{ctrl}})}{\text{TTS}_{\text{endpoint}}^{\text{nominal}}} + w_3 \frac{\text{Var}(\mathbf{g}(k_{\text{ctrl}}))}{\text{Var}^{\text{nominal}}}, \quad (1)$$

with:

- TTS(k_{ctrl}): total time spent by the vehicles in the network during the prediction interval,
- TTS^{nominal}: nominal performance for TTS(k_{ctrl}) during the prediction interval,
- TTS_{endpoint}(k_{ctrl}): expected time spent by the remaining vehicles in the network at the end of the prediction time interval until they leave the network,
- TTS_{endpoint}^{nominal}: nominal performance for TTS_{endpoint}(k_{ctrl}) during the prediction interval,
- Var(g): sum of all the consecutive time variations in the green time (this term prevents possible large oscillations of the control signal with time).

3.2 A Modified internal Flow Model for the Model-Predictive Controller

The MPC controller requires a traffic flow model. The model should be both accurate and simple, and should be able to handle all the computations for estimating the future states in real time. As a starting point, we have selected the S-model (Lin et al., 2012), which is a simplified form of the BLX model given by van den Berg et al. (2007) and by Lin and Xi (2008); we propose new modifications for the model to make it more accurate and computationally efficient.

The S-model is a nonlinear flow model for urban traffic networks that takes into account different cycle times for different intersections. Intersections are represented by nodes in the S-model, and a road between two intersections u and d is considered as a link between nodes u and d and is denoted by (u, d) ; \mathcal{L} and \mathcal{J} are the sets of links and nodes of the S-model.

The simulation time step of the S-model might vary for different links, as it is considered to be equal to the cycle time of the downstream intersection of that link. Therefore, different time step counters are also considered, i.e., for each link $(u, d) \in \mathcal{L}$ the time step counter is represented by k_d .

For the original S-model, two states are considered for each link in the network, i.e., the number of vehicles in link (u, d) indicated by $n_{u,d}(k_d)$, and the number of vehicles

within the queue in link (u, d) denoted by $q_{u,d}(k_d)$. These state variables of link $(u, d) \in \mathcal{L}$ are updated by:

$$n_{u,d}(k_d + 1) = n_{u,d}(k_d) + \left(\alpha_{u,d}^{\text{enter},l}(k_d) - \alpha_{u,d}^{\text{leave},l}(k_d) \right) c_d, \quad (2)$$

$$q_{u,d}(k_d) = \sum_{o \in \mathcal{O}_{u,d}} q_{u,d,o}(k_d), \quad (3)$$

with,

$$q_{u,d,o}(k_d + 1) = q_{u,d,o}(k_d) + \left(\alpha_{u,d,o}^{\text{arrive,q}}(k_d) - \alpha_{u,d,o}^{\text{leave,l}}(k_d) \right) c_d, \quad (4)$$

where we have:

- c_d : cycle time of the downstream intersection of link (u, d) ,
- $\mathcal{O}_{u,d}$: set of output nodes of link (u, d) ,
- $\alpha_{u,d}^{\text{enter},l}(k_d)$: average entering flow of link (u, d) during $[k_d c_d, (k_d + 1) c_d)$,
- $\alpha_{u,d}^{\text{leave},l}(k_d)$: average exiting flow of link (u, d) during $[k_d c_d, (k_d + 1) c_d)$,
- $q_{u,d,o}$: queue length in link (u, d) composed of vehicles intending to move to node o ,
- $\alpha_{u,d,o}^{\text{arrive,q}}(k_d)$: average arriving flow at the tail of the queue during $[k_d c_d, (k_d + 1) c_d)$,
- $\alpha_{u,d,o}^{\text{leave,l}}(k_d)$: average leaving flow towards node o during $[k_d c_d, (k_d + 1) c_d)$.

As the first modification to the S-model, we propose to add a new third state, i.e., the length of the queue that is formed at the sources of the traffic network. A source queue is formed when the available free space on the outgoing of the source is less than the number of vehicles that intend to enter the outgoing of the source (note that we assume w.l.o.g. that for each source there is only one outgoing link). For each source node s of the network, we can write:

$$q_{s,d_s}^{\text{source}}(k_{d_s} + 1) = q_{s,d_s}^{\text{source}}(k_{d_s}) + \alpha_{s,d_s}^{\text{enter},l}(k_{d_s}) c_{d_s} - C_{s,d_s} + n_{s,d_s}(k_{d_s}), \quad (5)$$

where $\alpha_{s,d_s}^{\text{enter},l}(k_{d_s}) c_{d_s}$ gives the total number of vehicles that intend to enter link (s, d_s) during $[k_{d_s} c_{d_s}, (k_{d_s} + 1) c_{d_s})$, and $C_{s,d_s} - n_{s,d_s}(k_{d_s})$ is the free space on link (s, d_s) at time step k_{d_s} . Note that in (5), we have supposed that the external link that ends at node s and feeds link (s, d_s) of the network, has an unlimited capacity such that all those vehicles that cannot enter link (s, d_s) , can stay in the link ending at s and become part of the source queue. This way, link (s, d_s) allows only $C_{s,d_s} - n_{s,d_s}(k_{d_s})$ number of new vehicles to enter it.

Figure 2 illustrates a link within an urban traffic network, where the entering, leaving, and arriving flows are shown. We have:

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

with $\beta_{u,d,o}(k_d)$ the fraction of vehicles in link (u, d) that intend to move towards o , and $\alpha_{u,d}^{\text{arrive,q}}(k_d)$ the average flow arriving at the tail of the waiting queue during $[k_d c_d, (k_d + 1) c_d)$.

We propose the following modified equation for the arriving flow rate:

$$\alpha_{u,d}^{\text{arrive,q}}(k_d) = \frac{c_d - \tau_{u,d}(k_d)}{c_d} \alpha_{u,d}^{\text{enter},l}(k_d) + \frac{\tau_{u,d}(k_d - 1)}{c_d} \alpha_{u,d}^{\text{enter},l}(k_d - 1), \quad (7)$$

where the time delay $\tau_{u,d}(k_d)$ is indeed the time needed by the vehicles that enter the link until they reach the end of

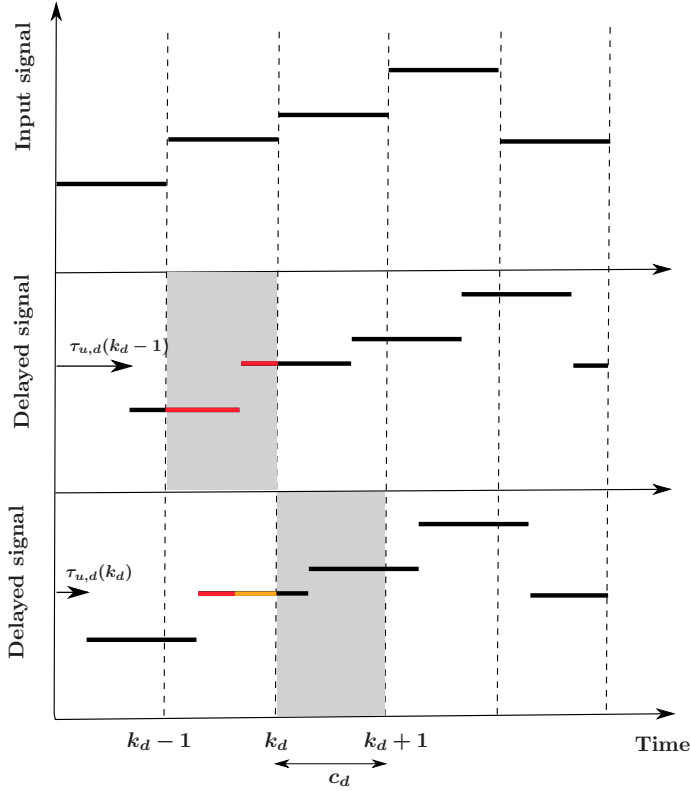


Fig. 3. Continuous form of the piecewise constant input signal, $\alpha_{u,d}^{\text{enter},1}(k_d c_d)$, and the delayed input signal, $\alpha_{u,d}^{\text{enter},1}(k_d c_d - \tau_{u,d}(k_d))$

the waiting queue. Compared with the equation given for $\alpha_{u,d}^{\text{arrive},q}(k_d)$ by Lin et al. (2012) in the original version of the S-model which considers a time-independent $\tau_{u,d}$, we here consider a time-dependent time delay as this is more consistent with the rest of the equations of the S-model where the time delay is indeed time-dependent.

Now we explain how (7) is obtained. The arriving flow at k_d is indeed the entering flow with a delay of $\tau_{u,d}(k_d)$ time steps. Therefore, if we plot these flows along the time axis, the curve representing the arriving flow is the same as the curve that represents the entering flow with a shift equal to $\tau_{u,d}(k_d)$ time units to the right. However, we should make sure that if the time delay varies over time, then in the long term we never miss or reconsider some fractions of the entering flows in computing the arriving flows.

Fig. 3, shows the shifted entering flow, i.e., the arriving flow functions where the grey areas show the current time step. From this figure, the red (dashed) sections of the input signal are considered in the computation of the arriving flow during the time interval $[(k_d - 1)c_d, k_d c_d)$. For the next step, i.e., for the interval $[k_d c_d, (k_d + 1)c_d)$, if the value of the time delay changes compared with $k_d - 1$ (e.g., $\tau_{u,d}(k_d) < \tau_{u,d}(k_d - 1)$), then the yellow (dashed-dotted) section shown in Figure 3 will be ignored in the computation of the arriving flow during the time interval $[k_d c_d, (k_d + 1)c_d)$. To compensate for this issue, we add this part of the input signal to the input considered for the interval $[k_d c_d, (k_d + 1)c_d)$ as follows:

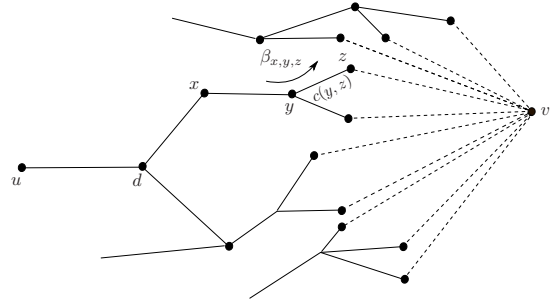


Fig. 4. Considering a point-to-point problem to find the $K_{u,d}$ most likely used routes in a traffic network

$$\alpha_{u,d}^{\text{arrive},q}(k_d) = \frac{1}{c_d} \int_{k_d c_d - (\tau_{u,d}(k_d - 1) - \tau_{u,d}(k_d))}^{k_d c_d + c_d} \alpha_{u,d}^{\text{enter},1}(t - \tau_{u,d}(t)) dt, \quad (8)$$

which gives (7) considering a piecewise constant entering function flow.

Finally, the delay time within the time interval $[k_d c_d, (k_d + 1)c_d)$ is computed by:

$$\tau_{u,d}(k_d) = \frac{(C_{u,d} - q_{u,d}^{\text{ave}}(k_d)) \cdot l_{\text{veh}}}{N_{u,d}^{\text{lane}} \cdot v_{u,d}^{\text{free}}}, \quad (9)$$

with:

- $C_{u,d}$: storage capacity of link (u, d) ,
- $q_{u,d}^{\text{ave}}(k_d)$: average queue length on the link (u, d) during $[k_d c_d, (k_d + 1)c_d)$,
- l_{veh} : average length of the vehicles,
- $N_{u,d}^{\text{lane}}$: number of lanes in link (u, d) ,
- $v_{u,d}^{\text{free}}$: free-flow speed on link (u, d) .

In order to compute the average queue length $q_{u,d}^{\text{ave}}(k_d)$, we propose the following options:

- (1) Substituting $q_{u,d}^{\text{ave}}(k_d)$ with the queue length at k_d , i.e.,

$$q_{u,d}^{\text{ave}}(k_d) = q_{u,d}(k_d),$$

- (2) Using extrapolation, i.e.,

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

$$\begin{aligned} q_{u,d}^{\text{ave}}(k_d) &= \frac{q_{u,d}(k_d) + q_{u,d}(k_d + 1)}{2} \\ &= \frac{3}{2} q_{u,d}(k_d) - \frac{1}{2} q_{u,d}(k_d - 1) \end{aligned}$$

4. ENDPOINT PENALTIES

Note that we have a destination-independent model. For determining the effect of those vehicles that will remain in the network at the end of the prediction horizon, we consider fixed traffic situations at the end of the prediction interval, $(k_{\text{ctrl}} + N_p)T_{\text{ctrl}}$, till the network is completely evacuated. We also suppose that no new vehicles will enter the network from $(k_{\text{ctrl}} + N_p)T_{\text{ctrl}}$ till the remaining vehicles leave the network. To estimate the time spent by the remaining vehicles in the network at $(k_{\text{ctrl}} + N_p)T_{\text{ctrl}}$, we should first allocate to each of these vehicles a route that leads them to one of the exit nodes of the network.

For the vehicles remaining on a given link (u, d) at $(k_d + N_p)T_{\text{ctrl}}$, we first specify a limited number $K_{u,d}$ of routes to the exit nodes that are most likely to be used by the remaining vehicles. This is necessary especially for

networks with grid-shaped parts, in which vehicles may move in cyclic paths and hence, the number of possible routes to the exits will become infinity.

most of the available shortest-path algorithms, e.g., Yen's algorithm (Yen, 1971), are developed such that they find the K shortest routes that connect two nodes (i.e., a single exit node should be considered). Therefore, we first transform the problem of finding the shortest routes in a traffic network into a point-to-point problem. We do that by connecting a virtual endpoint "v" to all end nodes of the network (see Figure 4).

We should also redefine some of the concepts for the recast problem. For a shortest-path algorithm, each route has a "cost" value, based on which routes with the least cumulative costs are investigated by the algorithm. We make use of the turning rates of the roads that are used by the S-model to define the costs. The routes that have the largest value of

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

are most likely to be selected by the vehicles in the network, where $R_j(d)$ is a route between two nodes d and v , $j \in \{1, 2, \dots, N_j(d)\}$, and $N_j(d)$ is the number of all possible (with no cyclic paths) routes from d to v , and $(x,y), (y,z) \in \mathcal{L}$. Equivalently, we can find the routes that have the largest value of

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

These routes will indeed have the least cumulative cost for the shortest-path algorithm. Hence, we can define the cost $C(y,z)$ of link (y,z) as:

$$C(y,z) = -\log \beta_{x,y,z} \quad (10)$$

Note that (10) is a valid expression for the cost value, since we have $0 \leq \beta_{x,y,z} \leq 1$ and therefore $C(y,z) \geq 0$.

Now we should adjust the values of the turning rates to the new setup. Since some of the links/routes are discarded from the network via the shortest-path algorithm, the summation of turning rates $\beta_{x,y,z}$ at some nodes might not be equal to the unity anymore. Therefore, $\gamma_{u,d,r}$, $r \in \{1, 2, \dots, K_{u,d}\}$ is redefined as the percentage of those vehicles in link (u,d) at the end of the prediction interval that intend to leave the network through the r^{th} route specified by the shortest-path algorithm. We have:

$$\gamma_{u,d,r} = \frac{\prod_{(x,y),(y,z) \in (R_r(d) \cup (u,d))} \beta_{x,y,z}}{\sum_{l=1}^{K_{u,d}} \left(\prod_{(x,y),(y,z) \in (R_l(d) \cup (u,d))} \beta_{x,y,z} \right)} \quad (11)$$

Finally, $\text{TTS}_{\text{endpoint}}(k_{\text{ctrl}})$ in (1) is computed by:

$$\text{TTS}_{\text{endpoint}}(k_{\text{ctrl}}) = \sum_{(u,d) \in \mathcal{L}} \left(n_{\text{endpoint},u,d} \sum_{r=1}^{K_{u,d}} \gamma_{u,d,r} \text{TTS}_{u,d,r} \right), \quad (12)$$

with $n_{\text{endpoint},u,d}$ the number of vehicles observed in link (u,d) at $(k_d + N_p)T_{\text{ctrl}}$, and $\text{TTS}_{u,d,r}$ the average total time spent by each vehicle that is in link (u,d) at the end of the prediction interval, until it exits the network. To

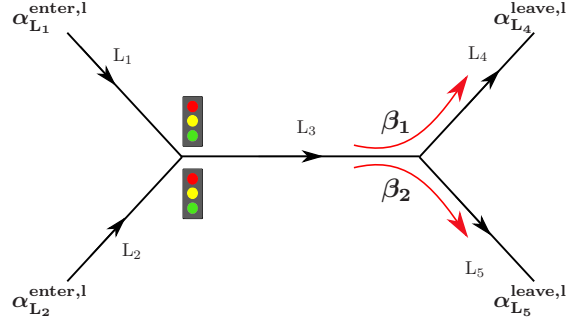


Fig. 5. Urban traffic network used for the case study

compute $\text{TTS}_{u,d,r}$, we can simply assume that all vehicles will move through the links with a fixed average speed of

$$v^{\text{ave}} = \frac{v_{u,d}^{\text{free}} + v_{u,d}^{\text{low}}}{2},$$

with $v_{u,d}^{\text{low}}$ the typical idling speed of the vehicles in the network.

5. SIMULATION AND RESULTS

Table 1. Parameters used for the case study

Parameter	value
l_{veh} [m]	7
g_{max} [s]	55
g_{min} [s]	5
C_{L_1} [veh]	55
C_{L_2} [veh]	50
C_{L_3} [veh]	75
C_{L_4} [veh]	65
C_{L_5} [veh]	50
μ_{L_i} , $i \in \{1, \dots, 5\}$ [veh/s]	1.5
$v_{L_i}^{\text{free}}$, $i \in \{1, \dots, 5\}$ [m/s]	12
c_{L_i} , $i \in \{1, \dots, 5\}$ [s]	60
β_1 [-]	0.7
β_2 [-]	0.3

In this section, we consider a case study involving an urban traffic network that is composed of two entrances, two exits, and one intermediate link (see Figure 5), and the links are represented by L_i , $i \in \{1, \dots, 5\}$. Two traffic lights are located at the intersecting point of the two entering links. The parameters used for the case study are listed in Table 1, where g_{max} and g_{min} denote the maximum and minimum possible values of the green time length and β_1 and β_2 are the turning rates for the vehicles on L_3 to L_4 and L_5 .

Table 2. Initial entering flow rates $\left[\frac{\text{veh}}{\text{s}} \right]$

	$\alpha_{L_1}^{\text{enter},1}(k_0)$	$\alpha_{L_2}^{\text{enter},1}(k_0)$
Case 1	1.2	0.5
Case 2	1.0	1.0
Case 3	0.2	1.3

Table 3. Total time spent [min] for the 3 cases

	Simulation 1	Simulation 2
Case 1	1771	1744
Case 2	2073	2031
Case 3	1556	1542

The entering flow rates from links L_1 and L_2 are initially (at $k_0 = 0$) set to the values shown in Table 2 for 3

different experiments. We consider a sine function with an amplitude equal to $\alpha_{L_i}^{\text{enter},1}(k_0)$ and a frequency of $\frac{\pi}{10}$ to model the entering flow rates through time. For each of these three cases, two controllers are considered, i.e.,

- fixed-time controller;
- model predictive controller for which both the TTS and the endpoint penalties are considered in formulation of the objective function.

The total simulation period for each of the two scenarios applied to Cases 1-3 is considered to be 75 min. We use `fmincon` in MATLAB to solve the minimization problem. The corresponding total time spent by the vehicles during the simulation period is computed for each scenario. The values of the total time spent are represented in Table 3. From this table we see that the controlled system using a model predictive controller always shows a better performance (i.e., a smaller total time spent by the vehicles) compared with the fixed-time control scenario. Implementing the model-predictive controller can reduce the total time spent by the vehicles up to 2%.

6. CONCLUSIONS AND FUTURE WORK

We have proposed a model-predictive urban traffic control system that aims to reduce the total time spent by the vehicles in the network and hence, to decrease the level of congestion. We have introduced a new version of an urban traffic flow model (i.e., the S-model), which is nonlinear. Our modifications can make the model more accurate, while the modified model is still computationally very efficient. Moreover, we have introduced a new third state for the S-model, i.e., the length of the queues that are formed at the sources of an urban traffic network. We have also proposed some modifications for the equations used in the S-model to make the model more consistent by considering a time-dependent delay time (i.e., the time needed for the vehicles that enter the network until they reach the tail of the waiting queue) in all equations.

The objective function of the MPC optimization problem is formulated as a weighted combination of the total time spent by the vehicles, the estimated value of the time spent by the vehicles that will remain in the network at the end of the prediction interval (i.e., endpoint penalties), and the sum of variations for consecutive green time signals (to prevent large oscillations in the control signal). We have developed an approach based on a shortest-path algorithm in order to approximate the endpoint penalties.

For future work, we propose to also consider the total emissions caused by the vehicles and the endpoint penalties corresponding to the expected emissions by the vehicles that will remain in the network at the end of the prediction interval in formulating the objective function of the MPC optimization problem. To solve the optimization problem, we propose to use an efficient method such as RPROP. The computation time and complexity can be compared for different optimization approaches.

ACKNOWLEDGEMENTS

This research has been supported by COST ARTS and by the NWO-NFSC 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

- Aboudolas, K., Papageorgiou, M., and Kosmatopoulos, E. (2009). Store-and-forward based methods for the signal control problem in large-scale congested urban road networks. *Transportation Research Part C*, 17(2), 163–174.
- Aboudolas, K., Papageorgiou, M., Kouvelas, A., and Kosmatopoulos, E. (2010). A rolling-horizon quadratic-programming approach to the signal control problem in large-scale congested urban road networks. *Transportation Research Part C*, 18(5), 680–694.
- Bellemans, T., De Schutter, B., and De Moor, B. (2006). Model predictive control for ramp metering of motorway traffic: A case study. *Control Engineering Practice*, 14(7), 757–767.
- Burger, M., van den Berg, M., Hegyi, A., De Schutter, B., and Hellendoorn, J. (2013). Considerations for model-based traffic control. *Transportation Research Part C*, 35, 1–19.
- Center for Economics and Business Research (2014). The future economic and environmental costs of gridlock in 2030. Technical report, INRIX.
- Diakaki, C., Dinopoulou, V., Aboudolas, K., Papageorgiou, M., Ben-Shabat, E., Seider, E., and Leibov, A. (2003). Extensions and new applications of the traffic-responsive urban control strategy: Coordinated signal control for urban networks. *Transportation Research Record*, 1856, 202–211.
- Diakaki, C., Papageorgiou, M., and Aboudolas, K. (2002). A multi-variable regulator approach to traffic-responsive network-wide signal control. *Control Engineering Practice*, 10(2), 183–195.
- Hegyi, A., De Schutter, B., and Hellendoorn, H. (2005). Model predictive control for optimal coordination of ramp metering and variable speed limits. *Transportation Research Part C*, 13(3), 185–209.
- Lin, S., De Schutter, B., Xi, Y., and Hellendoorn, H. (2012). Efficient network-wide model-based predictive control for urban traffic networks. *Transportation Research Part C*, 24, 122–140.
- Lin, S., De Schutter, B., Xi, Y., and Hellendoorn, H. (2013). Integrated urban traffic control for the reduction of travel delays and emissions. *IEEE Transactions on Intelligent Transportation Systems*, 1(4), 1609–1619.
- Lin, S. and Xi, Y. (2008). An efficient model for urban traffic network control. In *Proceedings of the 17th World Congress the International Federation of Automatic Control*, 14066–14071. Seoul, Korea.
- Maciejowski, J. (2002). *Predictive Control with Constraints*. Prentice Hall, London, UK.
- Thomson, J.M. (1997). Reflections on the economics of traffic congestion. *Journal of Transport Economics and Policy*, 32, 93–112.
- van den Berg, M., Hegyi, A., De Schutter, B., and Hellendoorn, J. (2007). Integrated traffic control for mixed urban and freeway networks: A model predictive control approach. *European Journal of Transport and Infrastructure Research*, 7(3), 223–250.
- Yen, J. (1971). Finding the K shortest loopless path in a network. *Management Science*, 17(11), 712–716.