Delft University of Technology

Delft Center for Systems and Control

Technical report 11-015

Model-based predictive traffic control: A piecewise-affine approach based on METANET*

N. Groot, B. De Schutter, S. Zegeye, and H. Hellendoorn

If you want to cite this report, please use the following reference instead:
N. Groot, B. De Schutter, S. Zegeye, and H. Hellendoorn, "Model-based predictive traffic control: A piecewise-affine approach based on METANET," *Proceedings of the 18th IFAC World Congress*, Milan, Italy, pp. 10709–10714, Aug.–Sept. 2011. doi:10.3182/20110828-6-IT-1002.03181

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

Model-Based Predictive Traffic Control: A Piecewise-Affine Approach Based on METANET

Noortje Groot * **Bart De Schutter** * **Solomon Zegeye** *
Hans Hellendoorn *

* *Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands (e-mail: n.b.groot@tudelft.nl)*

Abstract: As an alternative to the rather intensive computations when using the nonlinear traffic flow model METANET in a model-based predictive control context, a piecewise-affine (PWA) approximation of the model is proposed. Here, several model equations amongst which the fundamental diagram are approximated by a PWA function. Some selected methods to determine this approximation are shortly discussed. In addition, we make use of model properties and physical insight to improve the PWA approximation. For the purpose of traffic control, the PWA approximation of the METANET model can be used in a model predictive control (MPC) framework. In view of the on-line optimization used by MPC and the related trade-off between accuracy and computational speed, in a case study the accuracy of various approximations is compared to the original nonlinear formulation of METANET.

Keywords: traffic control; model-based predictive control; piecewise-affine model approximation

1. INTRODUCTION

For model-based predictive control of traffic networks one needs both a model that tracks the traffic states (flow, velocity, etc.) and an optimization approach that computes e.g., variable speed limits yielding an optimal traffic throughput. In this paper the second-order macroscopic model METANET is chosen, in which traffic is perceived as a flow. Such a model is commonly used as it has shown to provide good accuracy while it does not require as much computation time as microscopic models that take individual vehicles into account (see e.g., Hoogendoorn and Bovy (2001)). Further, we use the Model Predictive Control (MPC) approach, which predicts the evolution of states and determines the corresponding optimal decision variables. After implementation of the first set of these inputs the process is repeated. MPC has been frequently adopted when solving control problems as it easily incorporates various constraints and adapts well to structural changes in the system due to the moving horizon strategy (see e.g., Maciejowski (2002); Rawlings and Mayne (2009)).

Now, given a nonlinear nonconvex model (METANET) the on-line optimization problem of MPC will be difficult to solve quickly to optimality. For this reason the choice of a piecewise-affine (PWA) approximation of the nonlinear functions is made, which results in an MPC optimization problem that can be more easily solved as a mixed linear integer program (MILP). However, since integer programs are proven NP-complete (Garey and Johnson, 1979), attention should be paid to keeping the number of binary variables small as they increase the MILP's complexity. At the same time, the fewer such variables are allowed in the approximation, the larger the discrepancy with the original function. In other words, constructing a PWA formulation of the METANET model to be used in an MPC framework is nontrivial, yet it may result in a good solution in less time than when using the original nonlinear model.

In this paper, first the original METANET traffic model will be briefly stated (Section 2), after which we point at some possible methods for PWA approximation (Section 3). The adapted model equations are given in Section 4. In Section 5 the MPC formulation in traffic control is presented and a final case study in Section 6 shows the effect of the proposed piecewise-approximated METANET equations on the total time spent (TTS) by vehicles in a particular network, where several levels of accuracy of the approximations are evaluated.

2. METANET

The original METANET model developed by Messmer and Papageorgiou (1990) is discrete in time and space: the traffic network can be seen as a graph where links represent homogeneous parts of a freeway, further divided into segments of 500-1000 m. The nodes connecting links represent changes in the freeway like on-ramps and the merging of lanes. As regards the discretization in time, typically a simulation time step T_s of about 10 s is used.

The following equations represent the evolution of traffic flow $q_{m,i}$ (veh/h), density $\rho_{m,i}$ (veh/km/lane), and space-mean speed $v_{m,i}$ (km/h) for segment i of link m for time step k:

$$q_{m,i}(k) = \lambda_m \rho_{m,i}(k) v_{m,i}(k) \tag{1}$$

$$\rho_{m,i}(k+1) = \rho_{m,i}(k) + \frac{T_\mathrm{s}}{L_m \lambda_m} \left[q_{m,i-1}(k) - q_{m,i}(k) \right] \tag{2}$$

$$v_{m,i}(k+1) = v_{m,i}(k) + \frac{T_s}{\tau}[V[\rho_{m,i}(k)] - v_{m,i}(k)] + \frac{T_s v_{m,i}(k)\,[v_{m,i-1}(k) - v_{m,i}(k)]}{L_m} - \frac{T_s \eta\,[\rho_{m,i+1}(k) - \rho_{m,i}(k)]}{\tau L_m\,(\rho_{m,i}(k) + \kappa)}, \quad (3)$$

with λ_m the number of lanes and L_m the length (m) of the segments of link m. The time constant τ, η, and κ are model parameters. Commonly used values of these and other parameters are provided in Section 6.

The desired speed is represented by

$$V[\rho_{m,i}(k)] = \min\left[v_{\text{free},m} \exp\left[-\frac{1}{a_m}\left(\frac{\rho_{m,i}(k)}{\rho_{\text{cr},m}}\right)^{a_m}\right], (1+\alpha)v_{\text{control},m,i}(k)\right], \quad (4)$$

including variable-speed control, i.e., having as decision variable the speed limit $v_{\text{control},m,i}(k)$ (km/h) (Hegyi et al., 2005a). In case of no speed control the desired-speed equation consists only of the first term. Here, $v_{\text{free},m}$ (km/h) denotes the free-flow speed and α is a non-compliance factor. Further, a_m is a model parameter and $\rho_{\text{cr},m}$ (veh/km/lane) denotes the critical density.

Mainstream origins and on-ramps are modeled as a queue where w_o (veh) represents the queue length at origin o:

$$w_o(k+1) = w_o(k) + T_s(d_o(k) - q_o(k)). \quad (5)$$

Here, d_o (veh/h) denotes the traffic demand and q_o (veh/h) the outflow of origin o:

$$q_o(k) = \min\left[d_o(k) + \frac{w_o(k)}{T_s}, r_o(k)C_o, C_o\left(\frac{\rho_{\text{jam},m} - \rho_{m,1}(k)}{\rho_{\text{jam},m} - \rho_{\text{cr},m}}\right)\right], \quad (6)$$

for a metered on-ramp with ramp-metering rate $r_o(k) \in [0,1]$. For non-metered on-ramps or mainstream origins, $r_o(k)$ is set to 1. Further, C_o (veh/h) and $\rho_{\text{jam},m}$ (veh/km/lane) represent the capacity of origin o and the maximum density of link m connected to the given origin.

For the first segment of an outgoing link of each origin, the following speed-drop factor is added to speed equation (3) with δ a model parameter:

$$-\frac{\delta T_s q_o(k) v_{m,1}(k)}{L_m \lambda_m (\rho_{m,1}(k) + \kappa)}. \quad (7)$$

The METANET model can be further complemented to take into account e.g., merges and drops of lanes and the resulting speed drops, mainstream metering, or it can be adapted to different models for dynamic speed limits (see e.g., Kotsialos et al. (1999); Hegyi et al. (2005a,b)).

3. PWA APPROXIMATION

A PWA function consists of a collection of affine functions defined on polyhedra and can be written in general:

$$f(x) = a_i^T x + b_i \quad \text{if } x \in \Omega_i,$$

where $x \in \mathbb{R}^n$ denotes the independent variable and $a_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}$ are the constant vectors for each of the N convex polyhedra Ω_i in the x-space, such that $\cup_i \Omega_i = \mathbb{R}^n$ and $\text{int}(\Omega_i) \cap \text{int}(\Omega_j) = \emptyset$ for all i, j, $i \neq j$.

In the remainder of this section several possible approximation methods are shortly discussed. Here it should be noted that there is a difference in complexity between the approximation of single-variable and bivariate (or multivariate in general) functions, which both occur in the METANET setting we consider in this paper. For more information on the various available methods for PWA function approximations, see e.g., the overviews by Ferrari-Trecate et al. (2003) and Azuma et al. (2010).

Least-squares optimization For single-variate nonlinear functions one could determine the number of regions or intervals of the PWA function and optimize both the intervals and parameters of the affine functions using least-squares optimization, i.e., minimizing the squared difference between the original function and the approximation. E.g., the following PWA problem may be solved in a least-squares manner – here given for an approximation of a function f defined on an interval $[x_{\min}, x_{\max}]$ by a continuous PWA function $f_{\text{PWA}}(x)$ with 3 intervals:

$$\min_{\alpha,\beta,\gamma,\delta,\epsilon,\zeta} \int_{x_{\min}}^{x_{\max}} (f_{\text{PWA}}(x) - f(x))^2\, dx$$

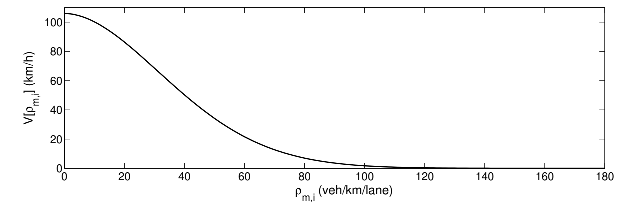
s.t.

$$f_{\text{PWA}}(x) = \begin{cases} \gamma + \dfrac{x - x_{\min}}{\alpha - x_{\min}}(\delta - \gamma) & \text{for } x_{\min} \leq x < \alpha \\ \delta + \dfrac{x - \alpha}{\beta - \alpha}(\epsilon - \delta) & \text{for } \alpha \leq x < \beta \\ \epsilon + \dfrac{x - \beta}{x_{\max} - \beta}(\zeta - \epsilon) & \text{for } \beta \leq x \leq x_{\max}. \end{cases}$$

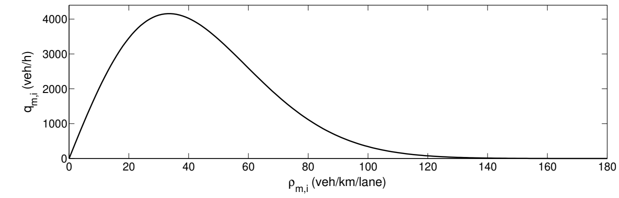
Piecewise-affine identification An alternative approach is hybrid or piecewise-affine identification. In general, the three available methods described here create local data sets after which the clustering algorithm creates local affine models by classifying the points. Similar models are again grouped into clusters, depending on the number of regions required (Ferrari-Trecate et al., 2003).

Amongst the available methods for pattern recognition the most precise for bivariate identification is the algorithm Multi-category Robust Linear Programming (MRLP) (Bredensteiner and Bennett, 1999). However, this method is computationally expensive and generally works for the identification of up to three polytopes based on up to 200 data points. This is due to the fact that one linear program is solved to find boundaries of all regions simultaneously. Alternatively, the clustering algorithms (Proximal) Support Vector Classification (P)(SVC) can be used, yet for multivariate estimation the original domain of the variables may then not be completely covered by the union of computed subregions. In contrast to MRLP the SVC approach (Vapnik, 1998) solves several quadratic programs in order to sequentially find boundaries between two regions or half-spaces at a time. The most time-efficient algorithm PSVC (Fung and Mangasarian, 2001) only requires a single system of linear equations. Compared to the non-proximal version, it assigns data points to the closest of two parallel half-planes that are maximally separated, leading to a strongly convex objective.

These algorithms are part of the Hybrid Identification Toolbox (HIT) (Ferrari-Trecate, 2005), a platform embedded within the Multi-Parametric Toolbox (MPT) for Matlab (Kvasnica et al., 2004).



(a) Speed as a function of density



(b) Flow as a function of density

Fig. 1. Fundamental diagrams of traffic flow

Partially piecewise-constant approximation A simple yet rough approximation approach for bivariate functions that uses relatively few auxiliary variables is by segmentation of the domain of one of the variables, where in each region or subdomain that variable is assigned a constant value. In general, a bivariate function $f(x, y)$ can be approximated as follows. Assume that based on the relative ranges $\frac{x_{\max}-x_{\min}}{x_{\max}}$ and $\frac{y_{\max}-y_{\min}}{y_{\max}}$ or the magnitude of the partial derivatives, variable x is selected to be taken constant in each region. Now a selection of N consecutive intervals $[x_i, x_{i+1}]$ for $i = 1, \ldots, N-1$ and with $x_1 = x_{\min}$, $x_N = x_{\max}$, we can set e.g.,

$$f(x, y) \approx f(\frac{x_i + x_{i+1}}{2}, y) \qquad \text{for } x \in [x_i, x_{i+1}].$$

Now, if f is linear in y (as will be the case for several functions appearing in the METANET model), this approach results in a PWA approximation of f. Alternatively the least-squares optimization approach discussed above can be applied for each function $f(\frac{x_i+x_{i+1}}{2}, y)$.

This approach is a more basic case of PWA approximation but it can deliver adequate approximation results for some functions. For single-variate functions this partially piecewise-constant approach may also be applied, but in general it is not very difficult to obtain a more accurate PWA formulation for this class of functions. In Section 6 the effect of each of these methods is shown; first we consider each nonlinear METANET equation separately.

4. PWA APPROXIMATION OF METANET

In this section the nonlinear METANET equations will be approximated by PWA functions, after which the new model can be incorporated into the PWA-MPC traffic control approach. When applied to this specific case it pays off to use additional information in the approximations, e.g., the fundamental diagrams of traffic flow depicted in Figure 1 (see also May (1990)) that represent the equilibrium relations between speed, flow, and density. Note further that (2), (5), and (6) do not need to be approximated as the first two functions are already linear and the latter equation is PWA.

4.1 Nonlinear flow equation (1)

Starting with the traffic flow, which is a function of density and speed (1), a simple approximation approach is to take one of the variables piecewise-constant. Having the smallest domain, we decide to do so for velocity variable $v_{m,i}(k)$ and substitute this continuous variable by the mean value of each subdomain. So (1) is transformed into

$$q_{m,i}(k) = \lambda_m \rho_{m,i}(k) \frac{v_j + v_{j+1}}{2} \quad \text{for } v_{m,i}(k) \in [v_j, v_{j+1}].$$

The intervals $[v_j, v_{j+1}]$ can be chosen by taking into consideration the shape of the function or determined in a more sophisticated way using optimization (cf. Section 3).

In the approximation of the bivariate equation (1) it is important to take into account the shape of fundamental diagram shown in Figure 1(a) and (b). To be more precise, in order to increase the accuracy of the approximation while trying to keep the number of regions or affine pieces small, one can assign different weights to data points or areas where a small error is important. Looking at the shape of the fundamental diagram it can be inferred that a situation in which the density and speed are simultaneously close to their maximum value is not likely to occur in real life; a similar argument holds for low speeds and densities. Thus, the focus should be on a good match in the regions determined by the fundamental diagram.

Note further that the approximated flow variable is used in (2), where the difference in flow between two segments is multiplied by a relatively small constant of about $9.26 \cdot 10^{-4}$ h/km/lane (taking the standard parameter values). Therefore, the effect of the approximation error made in the flow equation is relatively small when looking at the total model. However, $q_{m,i}(k)$ can take up large values of up to $6.5 \cdot 10^4$ veh/h. Also, one should realize that the iterative procedure of the optimization method causes this error to re-appear also in variables for later time steps of the prediction horizon.

The accuracy of the approximation therefore remains an important issue: for an approximation error of roughly 10% in the factor $v_{m,i}(k)\rho_{m,i}(k)$, the relative error in $q_{m,i}(k)$ will be of the same order. On the other hand, when adopting a piecewise-constant approach where only the velocity is approximated, an error of 10% in the velocity (on average 6 km/h) can bring down the impact on the relative error of $q_{m,i}(k)$ with a few percent.

4.2 Speed equations (3)–(4)

Within the speed equation (3) several issues should be dealt with, i.e.:

- the density variable arising in an exponential factor in the first term of (4),
- multiplication of speed variables,
- division of density variables by another density,
- subtraction of the term written separately in (7).

Density arising in the exponential term of equation (4) This exponential function represents the *fundamental diagram* of speed as a function of traffic density. As can be seen from the curve of Figure 1(a), approximation by only one affine equation would yield a too small accuracy due to the quick convergence of the function – w.r.t. the maximum density of 180 veh/km/lane – to zero. Instead an approximation using 2 or 3 pieces based on least-squares optimization can be considered (see Section 6).

From Figure 1(b) of the fundamental diagram, we can further deduce that the critical density ρ_{crit} of 33.5 veh/km/lane is especially important. For this purpose weighting can be added here, as well as to the area around a small density and close to the free-flow speed to improve performance in this critical region.

Multiplication of speed variables – $v_{m,i}(k)[v_{m,i-1}(k) - v_{m,i}(k)]$ Rather than choosing for a piecewise-affine approximation, here, we can choose to simply keep the first velocity variable $v_{m,i-1}(k)$ either constant at a value determined by historical data (in general) or equal to the currently measured value (in receding horizon). Note that the error caused by this method is overall relatively small due to the multiplication of the replaced velocity by the relatively small term T_s/L_m ($2.78 \cdot 10^{-3}$ s/m if the parameters are taken to be as before). For an approximation error of roughly 10% in the approximated speed variable, and given an average difference between velocity of two segments of 20 km/h, the approximation results in a relative error on $v_{m,i}(k+1)$ of less than a percent.

Alternatively, a more exact approximation could also be obtained using a similar method to that of bivariate equation (1).

Division by density – $\frac{\rho_{m,i+1}(k)-\rho_{m,i}(k)}{\rho_{m,i}(k)+\kappa}$ As in the previous step, the density term in the denominator is kept constant at a historically-based value or taken according the last measurements. However, note that the multiplication factor $\dfrac{T_s\eta}{\tau L_m}$ in the numerator is rather large (33.33 km), making the effect of the error of the approximated denominator not insignificant. On the other hand, the addition of the relatively large factor $\kappa = 40$ veh/km/lane in the denominator again reduces the error. For an approximation error of 10% in density, the relative error on $v_{m,i}(k)$ will be on average (taking the difference in density between two segments to be 30 veh/km/lane) of the same order.

Subtraction of the term in (7) Final adaptations are made to this speed-drop term by adopting the same constant approach of substituting the density in the denominator as in the paragraph above, combined with the substitution of $q_o(k)v_{m,1}(k)$ as in the piecewise-affine approximation of the flow equation (1).

It should be noted though that the impact of this term on the speed equation from which it is subtracted – and only for the first segment of an on-ramp's outgoing link – is not large: for the standard parameters we get a factor $\frac{1.1\cdot 10^{-5} q_o(k)v_{m,1}(k)}{\rho_{m,1}(k)+40}$. Thus, taken the relatively small multiplicative factor in the numerator, alternatively also speed variable $v_{m,1}(k)$ may be fixed at a constant value.

Finally, the on-ramp flow equation (6) is already PWA, finalizing the reformulation of the METANET model. However, in order to obtain a directly implementable optimization problem some further adaptations using binary auxiliary variables are needed, as will be explained next.

4.3 From PWA to MILP

To make the resulting PWA METANET model a directly solvable problem when incorporated with the MPC optimization method, the PWA model can be written as a mixed integer linear program (MILP) with some decision variables of an integer and some of a rational domain. The following statements provide a summary that covers the conversion (adapted from Bemporad and Morari (1999)).

Here, binary dummy variables (denoted by $\delta \in \{0, 1\}$) are introduced to indicate whether a certain region Ω_i of the PWA model is selected, i.e., whether a PWA function $y : X \to \mathbb{R}$ equals one of the affine pieces $f : X \to \mathbb{R}$ (defined over a bounded set $X \subset \mathbb{R}^n$), or to represent a product of binary variables. The constants m, M denote respectively a lower and upper bound of f over X. Finally, c denotes an arbitrary constant and ϵ represents the machine precision used to turn a strict inequality into a non-strict inequality (i.e., turning the equivalence statement below into an inequality implementable on digital computers).

Now we have

- $f(x) \le c \Leftrightarrow \delta = 1$ is equivalent to:
$$\begin{cases} f(x) \le c + (M-c)(1-\delta) \\ f(x) \ge c(1-\delta) + \epsilon + (m-\epsilon)\delta. \end{cases}$$
- $\delta = \delta_1\delta_2$ is equivalent to:
$$\begin{cases} -\delta_1 + \delta \le 0 \\ -\delta_2 + \delta \le 0 \\ \delta_1 + \delta_2 - \delta \le 1. \end{cases}$$
- $y = \delta f(x)$ is equivalent to:
$$\begin{cases} y \le M\delta \\ y \ge m\delta \\ y \le f(x) - m(1-\delta) \\ y \ge f(x) - M(1-\delta). \end{cases}$$

Concerning the implications of this conversion for the ease of computation of the final MPC problem, the number of regions and thus binary variables added can cause the solver to require drastically increasing computation times. However, a division using fewer regions may further increase the approximation error. Therefore an appropriate balance between speed and accuracy has to be found.

5. MPC FOR TRAFFIC CONTROL

Using MPC, based on measurements of the current state variables, at time step k future states are predicted for a prediction horizon of N_{p} steps. Optimizing the objective function over this horizon, the trajectory of optimal decision variables is determined. Implementing the first input, the procedure is repeated in a moving horizon fashion.

Amongst the possible optimization goals for traffic models are the maximization of traffic throughput, the spreading of traffic density, and the minimization of travel time or the variation in control variables. We chose as our objective the minimization of total time traffic spends (TTS) (veh.h) in the system, i.e., the time vehicles wait at on-ramps and in mainstream origin queues before joining the freeway plus the time they spend on the freeway itself:

$$J_{\mathrm{TTS}} = T_{\mathrm{s}} \sum_{k=1}^{N_{\mathrm{sim}}} \left(\sum_{(m,i)\in I_{\mathrm{all}}} L_m \lambda_m \rho_{m,i}(k) + \sum_{o\in O_{\mathrm{all}}} w_o(k) \right),$$

where N_{sim} expresses the simulation time over which we optimize, I_{all} is the set of pairs of indices (m, i) of all links and segments in the network, and O_{all} denotes the set of indices of all origins. Note that the TTS is linear in the state variables $\rho_{m,i}(k)$ and $w_o(k)$.

The objective of the MPC controller is to reduce the TTS over the prediction horizon N_{p}, i.e.,

$$J_{\mathrm{TTS}}^{\mathrm{MPC}}(k) = T_{\mathrm{s}} \sum_{j=1}^{N_{\mathrm{p}}} \left(\sum_{(m,i)\in I_{\mathrm{all}}} L_m \lambda_m \rho_{m,i}(k+j) + \sum_{o\in O_{\mathrm{all}}} w_o(k+j) \right).$$

In practice, one often adds a penalty term on input deviations and a control horizon $N_{\mathrm{c}} < N_{\mathrm{p}}$ is introduced after which the control signals are taken constant, i.e.,

$$\sum_{j=1}^{k+N_{\mathrm{c}}-1} \Big\{ a_{\mathrm{ramp}} \sum_{o\in O_{\mathrm{all}}} |r_o(k+j) - r_o(k+j-1)| +$$

$$a_{\mathrm{speed}} \sum_{(m,i)\in C_{\mathrm{all}}} |v_{\mathrm{control},m,i}(k+j) - v_{\mathrm{control},m,i}(k+j-1)| \Big\},$$

where a_{ramp} and a_{speed} are weighting coefficients and C_{all} is the set of all pairs of indices (m, i) of links and segments in which a variable speed limit is active. By introducing auxiliary variables and using the transformation properties given above, this penalty term can also be transformed into a linear function subject to a system of mixed integer equations. We thus end up with an MILP optimization problem, for which efficient solvers are available (see e.g., (Atamtürk and Savelsbergh, 2005)).

6. CASE STUDY

In this section the rewritten PWA formulation of the METANET traffic model is compared with a standard nonlinear formulation w.r.t. the total time spent (TTS). As a benchmark we use the case study of Hegyi et al. (2005a), looking at a two-lane freeway consisting of 6 segments, where segment 3 and 4 include a dynamic speed limit and an on-ramp is placed between segment 4 and 5.

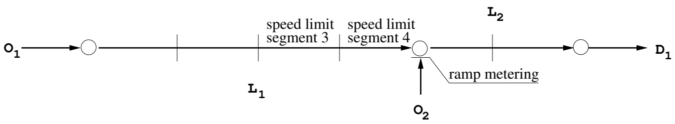


Fig. 2. Set-up of the case study

We take the standard parameter settings used by both Hegyi et al. (2005a) and Kotsialos et al. (1999): $v_{\mathrm{free}} = 102\,\mathrm{km/h}$, $T_{\mathrm{s}} = 10\,\mathrm{s}$, $\tau = 18\,\mathrm{s}$, $\kappa = 40\,\mathrm{veh/km/lane}$, $\eta = 60\,\mathrm{km^2/h}$, $\rho_{\mathrm{max}} = 180\,\mathrm{veh/km/lane}$, $\delta = 0.0122$, $a_m = 1.867$, $C_{o_1} = 4000\,\mathrm{veh/h}$ (mainstream origin), $C_{o_2} = 2000\,\mathrm{veh/h}$ (on-ramp), $\rho_{\mathrm{crit}} = 33.5\,\mathrm{veh/km/lane}$, $L_m = 1\,\mathrm{km}$, and $\alpha = 0.1$. We simulate the freeway for a simulation horizon N_{sim} corresponding to 40 min.

Coming back at the approximations made in the METANET model, Figures 3 and 4 show the most important approximations, i.e., of the nonlinear flow equation (1) and the fundamental diagram (4), obtained using PWA identification and least-squares optimization, respectively.

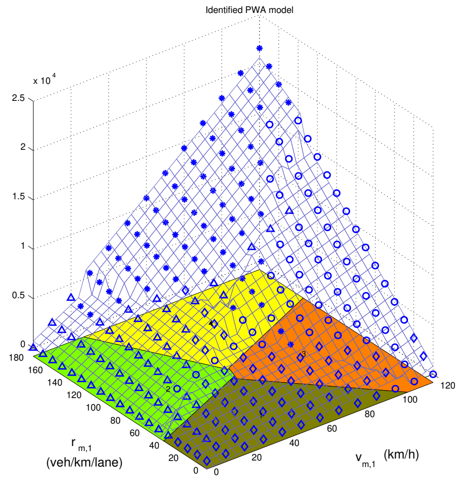


Fig. 3. Approximation of the flow equation (1)

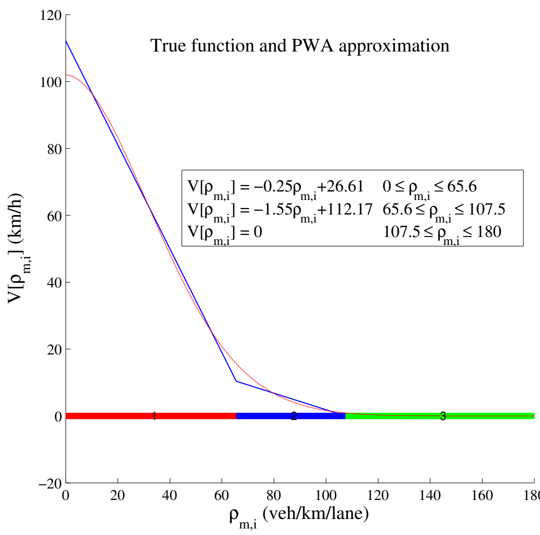


Fig. 4. Fundamental diagram – PWA approximation

Table 1 shows a selection of results for different levels of approximation. The description under ‘method’ refers to the approximation used, where ‘all approximations’ refers to the best approximation results for the fundamental diagram and flow equation, next to the substitution of the velocity and density variables by constants as discussed in Section 4. Thus, only the last row reflects a fully PWA model, as opposed to the approximation of only a subset of nonlinear equations in the other rows. Note further that the traffic scenario is run without applying control, i.e., not yet making use of variable ramp metering rates and speed limits.

It is interesting to see that taking the speed and density variables constant as explained in Section 4 can be done with only a small deterioration of the TTS w.r.t. the original nonlinear model, whereas a more accurate approximation of the fundamental diagram and flow equation results in a relatively large deviation from the original TTS. At the same time, the constant values are easy to tune: they can be adapted either separately or simultaneously to yield the results closest to the original TTS, which is

Table 1. Comparison of TTS for some levels of approximation and the relative error w.r.t. the original model

Method	TTS in veh.h (% error)
Original nonlinear	$1.46 \cdot 10^3$
$v(k)$ constant	$1.46 \cdot 10^3$ (0.01%)
$\rho(k)$ constant	$1.46 \cdot 10^3$ (0.01%)
Both $v(k), \rho(k)$ constant	$1.412 \cdot 10^3$ (3.3%)
Fundamental diagram with 2 pieces	$1.33 \cdot 10^3$ (8.84%)
Fundamental diagram with 3 pieces	$1.41 \cdot 10^3$ (3.4%)
Flow eq., partially piecewise-constant	$1.24 \cdot 10^3$ (15%)
All approximations, PWA	$1.23 \cdot 10^3$ (15%)

not possible for the PWA approximation of real functions. Overall, the deterioration due to approximation of the entire METANET model is not very small and should be improved by more elaborate tuning of the approximations.

7. CONCLUSIONS AND FURTHER RESEARCH

In the current paper, a PWA formulation of the traffic model METANET was made in order to ease the computational complexity of the original nonlinear nonconvex model-based traffic control problem. Several methods to approximate the nonlinear functions were discussed and a small case study showed the performance of the PWA formulation w.r.t. the total time spent (TTS) of traffic in the system. It turned out that, indeed, our approximations (without much tuning) resulted in a good overall performance, indicating it is fruitful to further extend the study.

It should be kept in mind that one specific traffic situation is tested; as part of future research an extensive assessment and (sensitivity) analysis of various approximations for different set-ups and scenarios could be made, e.g., incorporating vehicular emissions. Moreover, in this paper a fully controlled case is not yet run, while such an MPC optimization may reduce the effect of the approximation errors by means of adapting the control variables in a receding horizon approach, improving the final value of the objective function (TTS). At the same time a controlled simulation would give information concerning the running time under different levels of approximation and is therefore a logical next step. In addition, application-specific knowledge could be used as was done in e.g., the approximation of the flow equation based on the fundamental diagram. Also in the choice between taking a variable constant or approximating it more accurately, information from other model equations can prove useful. Here, one may benefit from a more systematic method allowing for a quick approximation while removing the need for much tuning. A bounded-error approach could be an example, including options to weigh data points that require a good match more heavily.

ACKNOWLEDGEMENTS

Research supported by the BSIK project 'Next Generation Infrastructures (NGI)', the Shell/TU Delft Sustainable Mobility program, the European 7th Framework Network of Excellence 'Highly-complex networked control systems (HYCON2)', the Delft Research Center Next Generation Infrastructures, and the Transport Research Centre Delft.

REFERENCES

Atamtürk, A. and Savelsbergh, M. (2005). Integer-programming software systems. *Annals of Operations Research*, 140(1), 67–124.

Azuma, S., Imura, J., and Sugie, T. (2010). Lebesgue piecewise affine approximation of nonlinear systems. *Nonlinear Analysis: Hybrid Systems*, 4(1), 92–102.

Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3), 407–427.

Bredensteiner, E. and Bennett, K. (1999). Multicategory classification by support vector machines. *Computational Optimizations and Applications*, 12(1–3), 53–79.

Ferrari-Trecate, G. (2005). Hybrid identification toolbox (HIT). `http://www-rocq.inria.fr/who/Giancarlo.Ferrari-Trecate/HIT_toolbox.html`.

Ferrari-Trecate, G., Muselli, M., Liberati, D., and Morari, M. (2003). A clustering technique for the identification of piecewise affine systems. *Automatica*, 39(2), 205–217.

Fung, G. and Mangasarian, O. (2001). Proximal support vector machine classifiers. In F. Provost and R. Srikant (eds.), *Proceedings KDD-2001: Knowledge Discovery and Data Mining*, 77–86. Asscociation for Computing Machinery, San Fransisco, U.S.A.

Garey, M. and Johnson, D. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, San Francisco, U.S.A.

Hegyi, A., De Schutter, B., and Hellendoorn, H. (2005a). Model predictive control for optimal coordination of ramp metering and variable speed limits. *Transportation Research Part C*, 13(3), 185–209.

Hegyi, A., De Schutter, B., and Hellendoorn, J. (2005b). Optimal coordination of variable speed limits to suppress shock waves. *IEEE Transactions on Intelligent Transportation Systems*, 6(1), 102–112.

Hoogendoorn, S. and Bovy, P. (2001). State-of-the-art of vehicular traffic flow modelling. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 215(4), 283–303.

Kotsialos, A., Papageorgiou, M., and Messmer, A. (1999). Optimal coordinated and integrated motorway network traffic control. In *Proceedings of the 14th International Symposium of Transportation and Traffic Theory (ISTTT)*, 621–644. Jerusalem, Israel.

Kvasnica, M., Grieder, P., and Baotić, M. (2004). Multi-parametric toolbox (mpt). `http://control.ee.ethz.ch/~mpt/`.

Maciejowski, J. (2002). *Predictive Control with Constraints.* Prentice Hall, Harlow, England.

May, A. (1990). *Traffic Flow Fundamentals.* Prentice-Hall, Englewood Cliffs, NJ, U.S.A.

Messmer, A. and Papageorgiou, M. (1990). METANET: A macroscopic simulation program for motorway networks. *Traffic Engineering and Control*, 31(9), 466–470.

Rawlings, J. and Mayne, D. (2009). *Model Predictive Control: Theory and Design.* Nob Hill Publishing, Madison, WI, U.S.A.

Vapnik, V. (1998). *Statistical Learning Theory.* John Wiley and Sons, New York, NY, U.S.A.