

Technical report 19-016

SPERT: A Speed Limit Strategy for Recurrent Traffic Jams*

J. R. D. Frejo and B. De Schutter

To cite this work, please refer to the published version:

J. R. D. Frejo and B. De Schutter, "SPERT: A speed limit strategy for recurrent traffic jams," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 692–703, Feb. 2019. doi:[10.1109/TITS.2018.2833628](https://doi.org/10.1109/TITS.2018.2833628)

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: <https://www.dcsc.tudelft.nl>

* This report can also be downloaded via <https://dpub.eu/19-016>

SPERT: A Speed Limit Strategy for Recurrent Traffic Jams*

José Ramón D. Frejo and Bart De Schutter, *Senior Member, IEEE*

Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands

(e-mail: j.r.dominguezfrejo@tudelft.nl; b.deschutter@tudelft.nl).

Abstract

This paper proposes and simulates a speed limit controller for recurrent traffic jams (SPERT). SPERT is a simple yet efficient variable speed limit (VSL) control strategy based on the behavior of the optimal controller without any need for online optimization. The online implementation of SPERT is a simple rule-based controller that activates and deactivates the corresponding variable speed limit when the densities of the dominant bottlenecks (which are found offline) reach predefined thresholds. These thresholds are defined in order to activate and deactivate the speed limits at the same bottleneck density at which they would be activated or deactivated in the nominal case. The simulation results show that SPERT is able to approach the optimal behavior while eliminating online computational cost, increasing robustness, and outperforming previously proposed easy-to-implement VSL control algorithms.

Index Terms: Variable speed limits (VSLs), freeway traffic control, optimal control

1 Introduction

Nowadays, traffic jams on freeways cause many social and economic problems (waste of time and fuel, a greater accident risk, and an increase in pollution). Since the construction of new freeways is not always a viable option or it is too costly, much research has been focused on solving these problems by using other solutions. One of the most promising one is the use of dynamic control signals such as ramp metering, Variable Speed Limits (VSLs), reversible lanes, and route guidance. These control measures have already been successfully implemented in practice in USA, Germany, Spain, Netherlands, and other countries [1].

In the past years, variable speed limits have emerged as a potential traffic management measure for increasing freeway efficiency [2–7], in contrast to safety-oriented applications such as [8] (a review about the use of VSLs for freeway traffic control can be found in [9]). When computing the VSL values that provide the highest efficiency increase, the use of appropriate non-local and multivariable control algorithms can considerably improve the reduction in the Total Time Spent (TTS), emissions, fuel consumption, and other traffic performance indices [2–5].

Among the available options described in the literature, Model Predictive Control (MPC) [10], which minimizes a cost function using a receding horizon approach, has shown to substantially improve the performance of the controlled traffic network in various simulation studies [2–4]. The main drawback of MPC is that the computation time quickly increases with the size of the network, making it difficult to apply centralized MPC for large traffic networks. Distributed and hybrid MPC may relieve these limitations [11–13] but, unfortunately, the obtained controllers are still too complex to be implemented in real time for large networks and, moreover, they are not robust in case of communication or measurement errors. For this and other reasons, completely centralized control of large traffic networks is still viewed by most practitioners as impractical and unrealistic. In order to overcome this practical problem, easy-to-implement control algorithms have been designed for ramp metering [14] and reversible lanes [15]. However, an easy-to-implement VSL control algorithm that approximates the performance of an MPC controller necessarily has to be a bit more complex.

*This work was supported by the European Union’s Horizon 2020 Research and Innovation Programme through the Marie Skłodowska-Curie Grant under Grant 702579.

Corresponding author: José Ramón D. Frejo

In the literature, a few practically implementable controllers designed to reduce congestion using VSLs have been proposed previously. In [16], a control algorithm (SPECIALIST) based on shock wave theory is proposed. This controller is able to solve/reduce isolated shock waves that do not necessarily always happen at the same time or that do not have the same magnitude. However, this controller does not take into account the optimal solution and, in some cases, solving a shock wave could create a new traffic jam or increase an existing one as can be seen in [17]. In [18], a local VSL controller is proposed that uses a cascade control structure with feedback of the density at the bottleneck area and the flow downstream of the VSL application area. However, similarly to SPECIALIST, this controller does not consider the optimal solution, entailing significant suboptimality in some cases as can be seen in [19]. In [20], microscopic simulations confirm that Mainstream Traffic Flow Control (MTFC) can successfully avoid the capacity drop and the onset of congestion, thereby increasing the performance of a freeway bottleneck. In [21], a reinforcement learning-based VSL control strategy is proposed. The VSL controller is trained off-line, using the Cell Transmission Model (CTM), to “learn the optimal speed limits for various traffic states to achieve a long-term goal of system optimization”. However, since the learning time increases exponentially as the number of state variables increases, the application of this controller to large networks is a challenge. In [6] a shockwave-based VSL controller is proposed that uses a heuristic switching logic-based control law with specified thresholds of prevailing traffic flow conditions.

When designing a VSL control method, it has to be taken into account in general that a simple linear or rule-based controller for VSL that can perform properly for one particular kind of congestion, is probably going to perform suboptimally for other kinds of congestion. Therefore, we propose the use of two control levels. In the upper level, a scheduling controller detects online the main kinds of congestion (recurrent congestion, shock waves, or unexpected capacity reductions) while a practically implementable controller for each kind of congestion is used in the lower level. This paper focuses on a lower-level VSL control algorithm for the first considered kind of congestion (recurrent congestion). In future work, the proposed control algorithm will be integrated into a fully developed two-level controller.

The main contribution of this paper is the proposal and simulation of a SPEEd limit controller for Recurrent Traffic jams, or SPERT. To the best of our knowledge, the proposed controller is the first VSL control algorithm that allows for an easy implementation (eliminating on-line computation costs) while, at the same time, it is based on the optimal solution (outperforming other local controllers in situations where the globally optimal solution differs substantially from the local one).

SPERT is a rule-based algorithm that increases or decreases the values of the VSL when the densities of the corresponding dominant bottlenecks reach predefined thresholds. These thresholds are defined in order to activate and deactivate the speed limits at the same bottleneck density at which they would be activated or deactivated in the nominal case.

Firstly, Section 2 introduces the macroscopic model used (METANET) and Section 3 summarizes the main aspects about computation of the optimal solution. Section 4 explains the main characteristics of SPERT, which is compared with other previously proposed controllers in Section 5. Finally, the simulation results are shown in Section 6.

2 Prediction and Simulation Model

The controller proposed in this work could be computed and tested using any macroscopic traffic flow model like METANET [22] or CTM [23]. For illustration purposes, we have chosen the second-order model METANET for both simulation and control because it provides a good trade-off between simulation speed and accuracy and it can handle control actions such as ramp metering [14], route guidance [24], reversible lanes [15], and VSLs [3].

METANET represents the traffic network as a graph where the links (indexed by m) correspond to freeway stretches, which are divided into N_m segments of length L_m with λ_m lanes. The traffic density $\rho_{m,i}(k)$ and the mean speed $v_{m,i}(k)$ dynamically characterize each segment where k is to the time step corresponding to instant $t = kT$ and T is the simulation time step. From now on, all segments will be considered to have different lengths and, therefore $N_m = 1 \forall m$, making it unnecessary to differentiate between links and segments; thus, hereafter only the index i will be used. In order to improve readability, Table 8 is included in the Appendix in order to explain, in a unified way, the variables used in this paper.

Two main equations describe the system dynamics of METANET model. The first one expresses the conservation of vehicles:

$$\rho_i(k+1) = \rho_i(k) + \frac{T}{\lambda_i L_i} (q_{i-1}(k) - q_i(k) + q_{r,i}(k) - \beta_i(k)q_{i-1}(k)) \quad (1)$$

where $q_{r,i}(k)$ is the traffic flow that enters the freeway segment i from the connected on-ramp (if any) and $\beta_i(k)$ is the split ratio of the off-ramp (i.e. the percentage of vehicles exiting the freeway through the off-ramp in segment i , if any). We set $\beta_i(k) = 0$ and $q_{r,i}(k) = 0$ for segments without an off-ramp or an on-ramp, respectively. The traffic flow in each segment $q_i(k)$ can be computed for each time step using $q_i(k) = \lambda_i \rho_i(k) v_i(k)$.

The second equation expresses the mean speed as a sum of the previous mean speed, a relaxation term, a convection term, and an anticipation term:

$$v_i(k+1) = v_i(k) + \frac{T}{\tau_i} (V(k) - v_i(k)) + \frac{T}{L_i} v_i(k) (v_{i-1}(k) - v_i(k)) - \frac{\mu_i T}{\tau_i L_i} \frac{\rho_{i+1}(k) - \rho_i(k)}{\rho_i(k) + K_i} \quad (2)$$

where K_i , τ_i , and μ_i are model parameters that have to be estimated for each segment and $V(k)$ is the desired speed for the drivers (see (3)). As proposed in [3], the model can take different values for μ_i , depending on whether the downstream density is higher than the density in the actual segment (resulting in $\mu_{H,i}$) or lower than the actual segment (resulting in $\mu_{L,i}$). The desired speed is modeled by the following equation, which includes the effect of the VSL as in [3]:

$$V(k) = \min \left(v_{f,i} e^{-\frac{1}{a_i} \left(\frac{\rho_i(k)}{\rho_{c,i}} \right)^{a_i}}, (1 + \alpha) V_{c,i}(k) \right) \quad (3)$$

where α is a model parameter, $V_{c,i}(k)$ is the value of the VSL in segment i , a_i is a model parameter, $v_{f,i}$ is the free flow speed that the cars reach in steady state, and $\rho_{c,i}$ is the critical density (the density corresponding to the maximum flow in the fundamental diagram). In [18] and other references, VSLs are included in the model by adapting the parameters of the fundamental diagram ($\rho_{c,i}$, $v_{f,i}$, and a_i). So far, to the best of our knowledge, there is not yet a model validation and/or numerical comparison between both ways of including the effects of the VSL on the literature. This will be the topic of a future work that will be done in order to justify the VSL model choice.

As proposed in [25], an extra penalization term is added to the speed equation (2) if there is an on-ramp in segment i in order to account for the speed drop caused by merging phenomena:

$$-\frac{\delta_i T q_{r,i}(k) v_i(k)}{L_i \lambda_i (\rho_i(k) + K_i)} \quad (4)$$

where δ_i is a model parameter.

In order to complete the model, the following equation defines the flow that enters from an uncontrolled on-ramp:

$$q_{r,i}(k) = \min \left(C_{r,i}, D_i(k) + \frac{w_i(k)}{T}, C_{r,i} \frac{\rho_{m,i} - \rho_i(k)}{\rho_{m,i} - \rho_{c,i}} \right) \quad (5)$$

where $\rho_{m,i}$ and $C_{r,i}$ are model parameters and $w_i(k)$ is the queue length on a ramp on segment i , the dynamics of which are defined by

$$w_i(k+1) = w_i(k) + T (D_i(k) - q_{r,i}(k)) \quad (6)$$

where $D_i(k)$ is the demand of the on-ramp connected to segment i . The mainline flow entering the first segment and the downstream density of the last segment are modeled as explained in [3].

3 VSL Optimal Solution

The computation of the VSL optimal solution has been analyzed in many previous references either using a rolling prediction horizon [3, 4] or optimizing over the entire simulation horizon [5]. In this work, we use the second choice because we will not compute any optimization on-line so the computation load is not a key factor.

The optimal solution for a given demand can be found by solving the following optimization problem with cost function $J(k)$ (see (8)), which is used to measure the performance of the system with respect to the VSL sequence [4]:

$$\min_{V_{c,t}(k)} J(k) \quad \text{with } V_{c,i}(k) \in S \text{ and } i = i_1, i_2, \dots, i_{N_{\text{VSL}}} \quad (7)$$

where S is the set of allowed values for the VSL, $V_{c,t}(k) = [V_{c,i_1}(k), V_{c,i_1}(k+1), \dots, V_{c,i_1}(k+N_s-1), V_{c,i_2}(k), V_{c,i_2}(k+1), \dots, V_{c,i_{N_{\text{VSL}}}}(k+N_s-1)]$ is the vector containing the VSL values, N_{VSL} the number of VSLs, and N_s is the number of time steps.

The main term of the cost function minimizes the Total Time Spent (TTS) of all vehicles in the network (including the waiting time experienced in the on-ramp queues). This is a natural, and most commonly used in the literature, cost function for freeway traffic systems. Moreover, under the condition that the network inflow is known or can be predicted accurately, as it is assumed in the current paper, minimizing the TTS is equivalent to maximizing the time-weighted outflow of the network (as shown in [26]).

The second term of the cost function limits (using a soft constraint in order to make optimization faster) the maximum value of the on-ramp queues. The third term penalizes VSL variations in order to avoid undesirable oscillations. The second term is equal to 0 if queue constraints are not violated and the last term is much smaller than the TTS:

$$J(k) = \sum_{\ell=1}^{N_s} \left[T \sum_{i \in O} w_i(k + \ell) + T \sum_{i \in I} (\rho_i(k + \ell) L_i \lambda_i) + \epsilon \sum_{i \in O} \Omega_i(k + \ell) + \psi \sum_{i=1}^{N_{VSL}} (V_{c,i}(k + \ell) - V_{c,i}(k + \ell - 1))^2 \right] \quad (8)$$

where $\Omega_i(k + \ell) = \max((q_{r,i}(k + \ell) - q_{r,\max}), 0)$, O is the set of all the segments with an on-ramp, ψ and ϵ are tuning parameters, and I is the set of all the segments. If it is desired to limit or reduce the frequency of speed limit switching, an additional term penalizing each switch may be included in the cost function.

In this work, optimizations have been computed continuously using the gradient-based optimization algorithm RPROP [27, 28]. Subsequently, the continuous VSL values have been discretized. Another possibility would have been to directly optimize the VSL profiles by using discrete optimization [11]. It has to be pointed out that, for most cases, it is necessary to run the optimization algorithm many times (with different initial points) in order to avoid ending upon local minima (because the problem is non-convex). Possible candidates for initial points include the minimum values for the speed limits, a set of random initial points, the VSL profiles given by other simpler controllers or a previously computed optimal solution for a similar scenario.

4 SPERT: A SPEEd Limit Controller for Recurrent Traffic Jams

The employment of optimal control techniques in order to compute online the speed limit values is not deemed sufficiently practicable for ready field implementation because of the computation times required, the need for robustness of the controller against communication or measurement errors, and the not very intuitive controller behavior (from the traffic operator point of view).

In order to try to address these issues while approaching the behavior of an optimal controller under different circumstances or kinds of congestion, we propose the use of two control levels with, in the upper level, a scheduling controller that detects online the kind of congestion that the traffic network is showing and, in the lower level, a practically implementable controller for each kind of congestion.

The higher-level controller mainly consists of a set triggering conditions that, after identifying the current state of a part of the network (uncongested, recurrent bottleneck, shock wave, incident causing unexpected congestion, ...), decide when each low-level controller has to be active and which parameters or inputs (e.g., which typical demand) have to be used. The higher-level controller will be detailed in future work.

This paper focuses on the lower-level VSL control algorithm for recurrent congestion (SPERT), a simple yet efficient VSL control strategy based on the behavior of the optimal controller without any need for on-line optimization. The online implementation of SPERT involves a simple rule-based controller that activates and deactivates the corresponding variable speed limit when the densities of the dominant bottlenecks reach predefined thresholds, given by the nominal case.

This paper extends the algorithm proposed in [29], where a first approach to SPERT is presented. However, in [29], the algorithm is still not generalized for different bottlenecks, the formulation of the controller is not general, the network used for case study is simpler and smaller, the controller is not compared with other control algorithms, and many explanations and details about SPERT are not yet provided. The results obtained for the simulation done for the conference paper [29] already show that SPERT is able to locally approach to behavior of an optimal controller for a single bottleneck. However, the improvement obtained with respect to the performance of other easy-to-implement algorithms is less significant than the one obtained in the current paper, with two active bottlenecks. This suggests that, as explained in the current paper, the advantages of SPERT will be improved if the size of the network is increased.

SPERT is composed of the following 6 steps: 1: Typical Demand Estimation, 2: Discrete Optimal Solution, 3: Traffic Jam Splitting, 4: Bottleneck Selection, 5: Definition of the Density Thresholds, and 6:

Online Implementation. The first 5 steps are computed off-line so they have to be performed only once and their computation load is not a limitation.

Step 1: Typical Demand Estimation

The typical demand is obtained by averaging the measured demands for weekdays with available measurements and without incidents. For different congestion/demand profiles depending on weather conditions or other measurable/estimable events, one typical demand should be defined for each case. If the obtained typical demand is noisy, a smoother demand should be obtained by using a filter (such as an Exponential Smoothing Filter [30]) in order to reduce the number of suboptimal local minima that may appear during the optimization process at step 2. If necessary, more advanced methods for demand estimation could be used [31].

Step 2: Discrete Optimal Solution

The discrete optimal VSL profiles for the typical demands (obtained in step 1) are computed off-line by using the procedure explained in Section 3. The optimization horizon is typically a peak period of a few hours or a day (at most). From now on, these optimal VSL profiles computed using the typical demand will be called nominal VSL profiles and denoted by $V_{c,i}^{\text{nom}}(\cdot)$.

Step 3: Traffic Jam Splitting

The different recurrent traffic jams that appear in the network for the typical demands, computed in Step 1, are now divided in time and space. The resulting set of “local traffic jams” is denoted as L . The splitting is done based on the densities of the no-control case ($\rho_i^{\text{NC}}(k)$) and the nominal VSL profiles ($V_{c,i}^{\text{nom}}(\cdot)$):

- Firstly, an initial set L_1 is considered, which is only composed by one traffic jam covering the entire simulation period.
- Subsequently, an intermediate set L_2 is obtained by spatially dividing L_1 : If segment i is uncongested during the entire no-control simulation and its corresponding nominal VSL is deactivated during the entire nominal simulation (i.e. $\rho_i^{\text{NC}}(k) < \rho_{c,i}, \forall k$ and $V_{c,i}^{\text{nom}}(k) = V_{c,i}^{\text{max}}, \forall k$, where $V_{c,i}^{\text{max}}$ is the maximum value allowed for the VSL on segment i), the traffic jams upstream and downstream of segment i at the time step k are considered to be different traffic jams.
- Lastly, the final set of “local traffic jams” $L = L_3$ is found by temporally dividing the set L_2 obtained in the previous substep: If at step time k , all the VSLs within a local traffic jam in L_2 are deactivated, the traffic jams occurring after and before time step k are considered as different traffic jams.

Step 4: Bottleneck Selection

SPERT activates and deactivates the VSLs based on the densities of the dominant bottlenecks (the bottlenecks with the highest influence on the VSLs profiles computed by the optimal controller). During this step, the algorithm finds what segments have to be used as dominant bottleneck for each VSL. In order to reach this goal, the following procedure is carried out:

- An initial set $P_{1,l}$ of segments of potential bottlenecks is considered for each “local traffic jams” l including the segments with an on-ramp, with a reduction in the number of lanes, and/or with a lower capacity than the upstream segment (due to changes in $v_{f,i}, \rho_{c,i}$, and/or a_i).
- An estimation of the magnitude of the congestion ($MC_{i,l}$) for the potential bottleneck on segment i during traffic jam l is computed by using the following equation:

$$MC_{i,l} = \sum_{k=k_{i,l}}^{k_{e,l}} \Delta_i^{\text{NC}}(k)$$

with

$$\Delta_i^{\text{NC}}(k) = \begin{cases} \frac{\rho_i^{\text{NC}}(k) - \rho_{c,i}}{\rho_{c,i}}, & \text{if } \rho_i^{\text{NC}}(k) > \rho_{c,i} \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

where $k_{i,l}$ and $k_{e,l}$ are the time steps corresponding to the beginning and the end of traffic jam l .

- The potential bottleneck segments showing a much lower MC value than the maximum one ($MC_{i,l} < \theta \max_i(MC_{i,l})$ with θ a threshold parameter) are removed as potential bottlenecks, reducing the set of potential bottlenecks to $P_{2,l}$.
- In order to determine which bottlenecks are creating congestion by themselves and which ones are receiving congestion from upstream segments, the aggregated difference $D_{i,l}$, for traffic jam l , is computed as follows:

$$D_{i,l} = \sum_{k=k_{i,l}}^{k_{e,l}} \Phi_i^{\text{NC}}(k)$$

with

$$\Phi_i^{\text{NC}}(k) = \begin{cases} \frac{\rho_i^{\text{NC}}(k) - \rho_{i+1}^{\text{NC}}(k)}{\rho_{c,i}}, & \text{if } \rho_i^{\text{NC}}(k) > \rho_{c,i} \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

- The potential bottlenecks with small or negative differences ($D_{j,l} < \omega$, where ω is a parameter) are removed, reducing the set of potential bottlenecks to $P_{3,l}$.
- The sample Pearson correlation coefficient $r_{x,y} = r_{V_{c,i}^{\text{nom}}, \rho_i^{\text{nom}}}$ is computed for each correlation between a nominal VSL ($V_{c,i}^{\text{nom}}(\cdot)$) and the densities ($\rho_i^{\text{nom}}(\cdot)$) of a potential bottlenecks i included in $P_{3,l}$ for the nominal case (i.e. using $V_{c,i}^{\text{nom}}(\cdot)$). We have chosen the Pearson correlation coefficient since it is the most commonly used measure of bivariate association but other ones could be used such as the Spearman's rank correlation coefficient. The sample Pearson coefficient between $x_k = V_{c,i}^{\text{nom}}(k)$ and $y_k = \rho_i^{\text{nom}}(k)$ is defined as follows:

$$r_{x,y} = \frac{n \sum x_k y_k - \sum x_k \sum y_k}{\sqrt{n \sum x_k^2 - (\sum x_k)^2} \sqrt{n \sum y_k^2 - (\sum y_k)^2}} \quad (11)$$

with $\sum = \sum_{k_i}^{k_e}$ and $n = k_e - k_i$.

- Finally, each VSL is linked to the bottleneck in $P_{3,l}$ with the lowest value of the Pearson's correlation coefficient (closer to -1). The bottlenecks upstream of a VSL are not considered since a speed limit cannot substantially increase the flow of an upstream active bottleneck (except in the case of non-recurrent shock waves, which are not considered by SPERT). Note that the segments that do not have an active speed limit in the nominal VSL solution, will be also inactive using SPERT.

Step 5: Definition of the Density Thresholds

Comparing the optimal solutions for different demand profiles around the nominal one, it can be seen that, for a recurrent bottleneck, the activation of the VSL usually happens at similar values of the density of the bottleneck. However, the time at which this density is reached, is varying from one demand to another. Moreover, for the optimal solution the correlation between the activation/deactivation of the VSLs with respect to other variables (bottleneck flow, density of the segments with the VSL,) is also much lower than with respect to the bottleneck density.

Therefore, since SPERT is proposed in order to obtain an efficient VSL control strategy that approximates the behavior of the optimal controller without the need of any on-line optimization, the thresholds are defined in such a way that the speed limits are activated and deactivated at the same bottleneck density at which they would be activated or deactivated in the optimal solution for the nominal case. These threshold values allow to approach the optimal case as be seen in the simulated case studies.

More concretely, the thresholds for the dominant bottleneck densities that determine, for each traffic jam l , when a VSL on segment i has to be decreased to the speed value S_m are defined based on $\rho_B^{\text{nom}}(k_{S_m,i,l}^-)$, that is the density at the dominant bottleneck at time $k_{S_m,i,l}^-$ when the nominal VSL of segment i is decreased to S_m km/h for the first time (for the considered traffic jam):

$$\hat{\rho}_{i,S_m,l} = \rho_B^{\text{nom}}(k_{S_m,i,l}^-)$$

with

$$V_{c,i}^{\text{nom}}(k_{S_m,i,l}^-) = S_m \quad \text{and} \quad V_{c,i}^{\text{nom}}(k_{S_m,i,l}^- - 1) > S_m \quad \text{and} \quad k_{i,l} < k_{S_m,i,l}^- < k_{e,l} \quad (12)$$

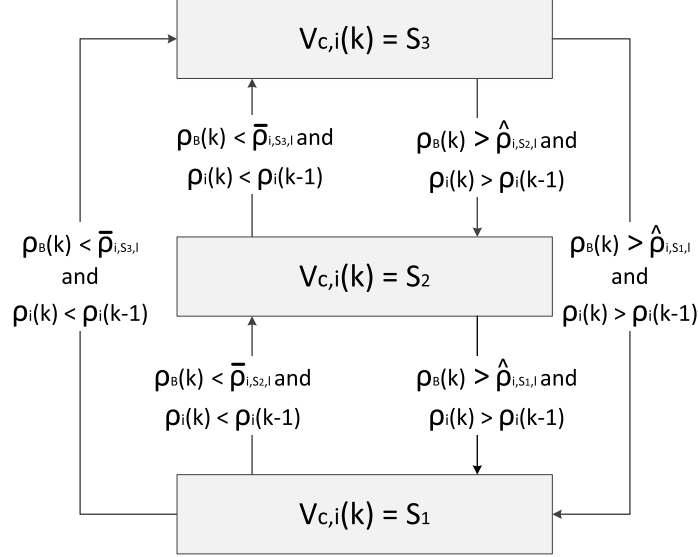


Figure 1: SPERT: Logic-based controller for VSL on segment i with 3 allowed values ($S_3 > S_2 > S_1$) during local traffic jam l .

Equivalently, the density thresholds that determine when a VSL on segment i has to be increased are defined based on $\rho_B^{\text{nom}}(k_{S_m,i,l}^+)$ (the density at the dominant bottleneck at time $k_{S_m,i,l}^+$ when nominal VSL of segment i is increased to S_m km/h for the first time for the considered traffic jam):

$$\bar{\rho}_{i,S_m,l} = \rho_B^{\text{nom}}(k_{S_m,i,l}^+)$$

with

$$V_{c,i}^{\text{nom}}(k_{S_m,i,l}^+) = S_m \text{ and } V_{c,i}^{\text{nom}}(k_{S_m,i,l}^+ - 1) < S_m \text{ and } k_{i,l} < k_{S_m,i,l}^+ < k_{e,l} \quad (13)$$

Therefore, using (12) and (13), the thresholds are defined in such a way that the speed limits are activated and deactivated at the same bottleneck density at which they would be activated or deactivated in the nominal case.

Step 6: Online Implementation

The online controllers are implemented using the logic in Fig. 1 and the density thresholds computed in the previous step. SPERT activates a variable speed limit on segment i when $\rho_{B,j}(k)$ (the density of the bottleneck dominating the corresponding speed limits $V_{c,i}(k)$) reaches the same value for which $V_{c,i}(k)$ was activated in the nominal case:

$$\begin{aligned} &\text{if } \rho_B(k) < \bar{\rho}_{i,S_m,l} \text{ and } V_{c,i}(k) < S_m \text{ and } k_{i,l} < k < k_{e,l} \\ &\text{then } V_{c,i}(k+1) = S_m \end{aligned} \quad (14)$$

Equivalently, SPERT deactivates the same VSL once $\rho_{B,j}(k)$ reaches the same value for which $V_{c,i}(k)$ was deactivated in the nominal case:

$$\begin{aligned} &\text{if } \rho_B(k) > \hat{\rho}_{i,S_m,l} \text{ and } V_{c,i}(k) > S_m \text{ and } k_{i,l} < k < k_{e,l} \\ &\text{then } V_{c,i}(k+1) = S_m \end{aligned} \quad (15)$$

Moreover, in order to avoid undesirable oscillations of the speed limit values and, thus, density and speed oscillations, an additional condition is included. This additional condition only allows to increase the VSL when the bottleneck density is decreasing and vice versa. When dealing with noisy measurements, these densities have to be an aggregation of data during the last minutes. When using ideal and soft demands, the following constraints can be used: $\rho_i(k) < \rho_i(k-1)$ for (14) and $\rho_i(k) > \rho_i(k-1)$ for (15).

Finally, if desired, strong VSL variations can be bounded (specially for lowering VSLs) in order to increase safety.

Table 1: Main VSL control algorithms designed to increase freeway efficiency.

Controller	Computational load		Solution provided	Optimality	Intuitiveness	Kinds of congestion	Robustness
	Off-line	On-line					
Optimal [5] (computed offline)	High	Very high	Continuous	Optimal solution (for the given demands)	Low	All	Lack of robustness
MPC [2, 3]	High	High	Continuous	Optimal solution (within a receding horizon)	Low	All	Lack of robustness against communication failures
Distributed MPC [12]	High	Medium	Continuous	Optimal solution (within a receding horizon)	Low	All	Lack of robustness (only at agent level)
Hybrid MPC [11]	High	Medium	Discrete	Optimal solution (within a receding horizon)	Low	All	Lack of robustness against communication failures
SPERT	High	Very low	Discrete	Suboptimal (based on the optimal solution)	Medium	Recurrent congestion	Robust (around typical demand)
SPECIALIST [16]	Low	Very low	Continuous	Not based on optimal solution	Medium	Shock waves	Robust (for shock waves)
Local MTFC [18]	Low	Very low	Continuous	Not based on optimal solution	Medium-low	Active bottlenecks	Robust (for jams caused by bottlenecks)
QL-based [21]	Very high	Low	Discrete	Based on reinforcement learning	Low	Recurrent bottlenecks	Robust (if properly trained)
Logic-based [6]	Medium	Very low	Discrete	Not based on optimal solution	Medium	Recurrent bottlenecks	Robust (for recurrent bottlenecks)

5 Advantages and Disadvantages of SPERT

Table 1 summarizes some characteristics of previously proposed VSL control algorithms for increasing freeway efficiency. The first row refers to the computational load required during each control step and the second one shows which algorithms provide a directly implementable discrete solution. The third and fourth rows analyze how optimal and intuitive (for the traffic community) the solution provided by each controller is. Finally, the last row comments the robustness of each controller against uncertainties and/or communication or measurement errors.

The main advantages, compared with previously proposed VSL control algorithms, of the proposed controller (SPERT) are:

- Controller implementation is quite easy because the online computation is almost instantaneous (only a few simple inequalities have to be evaluated for each VSL at each sample time) and only one variable has to be measured for each VSL (the density of the dominant bottleneck). More concretely, using an i5 CPU with 2.5 GHz, the computation time for a VSL signal during one controller time step is around 5 μ s. On the other hand, the computation time needed to find an optimal solution is highly variable and it depends on the size of the network, the optimization algorithm, the number and selection of the initial points, the horizons, etc. However, in all cases, the computation time of the optimal solution using a second-order traffic flow model is in the order of, at least, a few seconds. In addition, for large enough networks, it becomes quite difficult, if not impossible, to compute the optimal solution in real time because of the exponential increase of the computation time with the size of the network.
- The controller is implemented locally since it only needs to receive the value of the density of the corresponding bottleneck. Therefore, it is robust against communication and measurement failures in other segments different from the corresponding bottleneck. On the other hand, optimal techniques in general lack robustness against communication or measurement errors.
- The controller is based on the optimal solution outperforming other local controllers in situations where the global solution differs substantially from the local one as in [4]. For example, in some cases solving a bottleneck congestion could create a more intensive traffic jam in a dominant bottleneck downstream the solved one. In these and other cases the VSLs have to take into account the global performance of the network. The optimal solution computed offline allows to previously decide which VSLs will be activated for each bottleneck so the global network performance is taken into account.
- If a macroscopic model of the network is available or it can be identified automatically, the design process can be fully automated; so a control law for a large real network could be obtained without any human intervention. It has to be pointed out that for quite large networks, the off-line computation of the nominal VSL profiles may take such a long time that, in these cases, distributed algorithms [12] or other kinds of relaxation (like genetic algorithms [11]) have to be employed.
- The controller is more intuitive than MPC or even PI controllers for the traffic community, which is not always familiar with automatic control, since it only provides the thresholds at which the VSLs have to be changed.

The main disadvantage of the proposed controller is that SPERT is designed for solving recurrent congestion caused by bottlenecks and, thus, it will not be able to optimally solve non-recurrent congestion like:

- No-recurrent moving shock waves.
- Lane blocking accidents and disabled vehicles.
- Other lane blocking events (e.g., debris in the roadway, construction lane closures).
- Significant roadside distractions that alter driver behavior (e.g., roadside construction, electronic signs, a fire beside the freeway).
- Significant increases in traffic volume in comparison to “normal” traffic volumes.

However, a large percentage of the congestion created in the freeways around cities is due to recurrent bottlenecks (between 50 and 70% during peak periods), which create similar congestion profiles for different days [32].

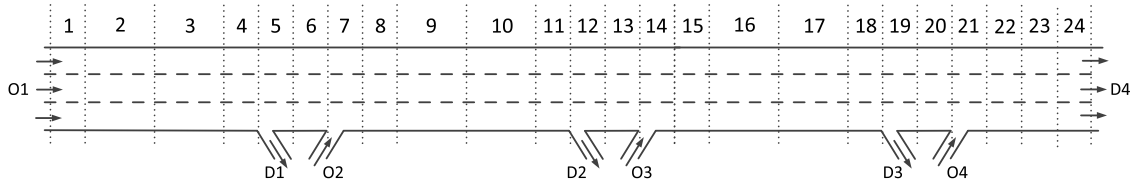


Figure 2: Simulated network.

Table 2: METANET parameters.

a	v_f	ρ_c	τ	μ_H
1.867	102 km/h	33.5 veh/(km lane)	18 s	20 km ² /h
μ_L	α	ρ_m	K	δ
80 km ² /h	0.1	180 veh/(km lane)	40	0.0122

For the previously mentioned non-recurrent kinds of congestion, other control algorithms should be used such as the ones of [16] and [18]. In future work, the higher-level controller (a set of triggering conditions defining when each controller has to be active or which typical demand has to be used) will be proposed and the behavior of new lower-level controllers will be studied in scenarios with other kinds of congestion (like shock waves or accidents).

6 Simulation Results

6.1 Simulated Network

A hypothetical 30 km long freeway stretch, shown in Fig. 2 is now used in order to simulate the proposed controller and to compare its performance with other previously proposed VSL control algorithms. The freeway has $N = 24$ segments with $\lambda_i = 3$ lanes and a length of $L_i = 2000$ m (for segments 2, 3, 9, 10, 16 and 17) or $L_i = 1000$ m (for the remaining segments). There are 22 VSLs (from segment 2 to segment 23), three on-ramps on segments 7, 14, and 21 (uncontrolled) and three off-ramps on segments 5, 12, and 19.

All the METANET parameters (which can be seen in Table 2) are considered to be the same for all the segments. The simulation time chosen is two and half hour corresponding to 75 controller sample steps ($T_c = 120$ s) and 900 simulation steps ($T = 10$ s). The set of allowed VSLs is $S = \{60, 80, 100\}$ km/h and no implementation constraints have been considered (i.e. the VSLs are allowed to change directly in space and time from 60 km/h to 100 km/h, and vice versa). The off-ramp split rates are considered constant and equal to 10%, 30% and 5% of the traffic flow for segments 5, 12, and 19, respectively ($\beta_5(k) = 0.2$, $\beta_{12}(k) = 0.3$, and $\beta_{21}(k) = 0.05 \forall k$). The on-ramps have a capacity of $C_{r,7} = C_{r,14} = C_{r,21} = 2000$ veh/h.

The considered typical demand for the mainline and the on-ramps can be seen in Fig. 3. These demands reproduce two flow increases during two consecutive peaks hours (for example, 8 AM and 9 AM). Other 26 scenarios have been considered in order to test to proposed controller under different traffic conditions.

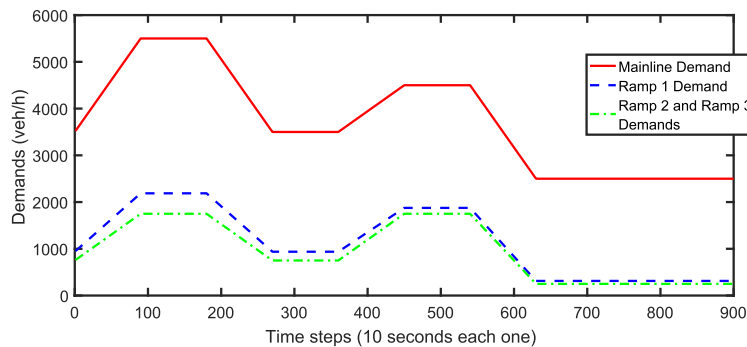


Figure 3: Typical demands.

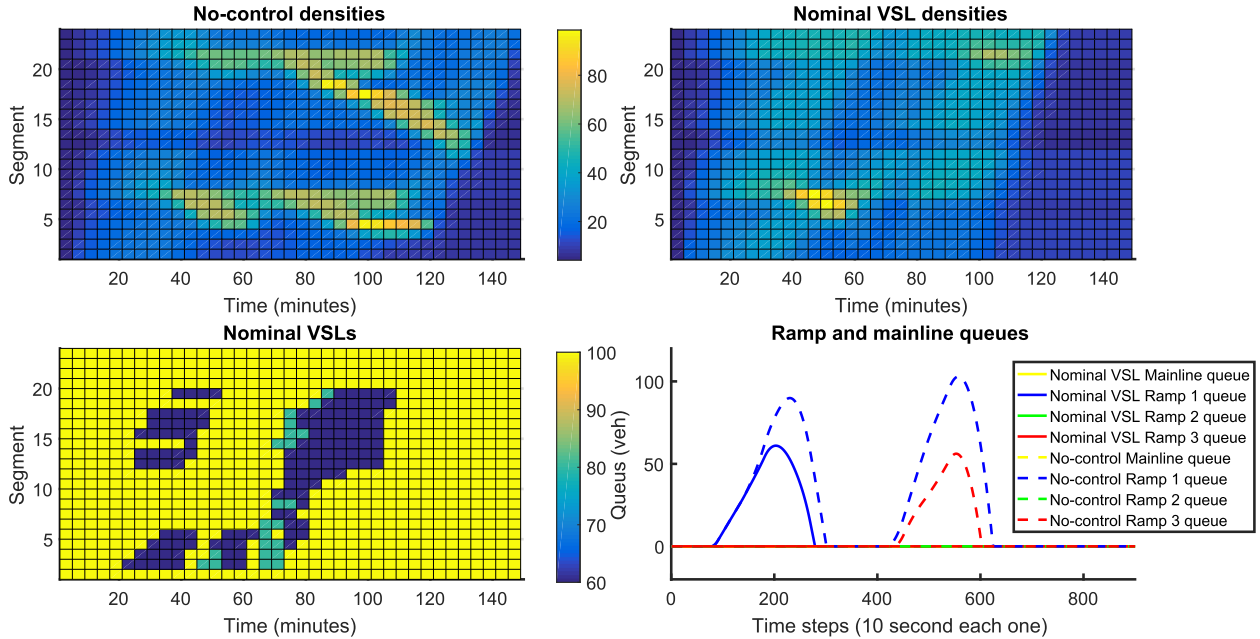


Figure 4: Densities, VSL and queues for no-control and nominal VSL in scenario 1.

These scenarios are obtained by considering all the possible combinations that can be obtained by increasing or decreasing (during the entire simulation) the demands of the mainline, the first ramp and/or the third ramp in a 10%. These scenarios can be seen on Table 7. The optimization problems (for determining the nominal VSL profiles and for the optimal controllers) have been solved as continuous ones using multi-start RPROP [27] and the continuous results have been discretized by rounding.

6.2 No-Control Case

The no-control case simulation (i.e. with no active VSL) entails a Total Time Spent (TTS) of 5608 veh h and congested density profiles as shown in Fig. 4. It can be seen that there are two main active bottlenecks, one on segment 7 and another one on segment 21, the last one being the dominant bottleneck.

6.3 Optimal Solution

For each scenario, the optimal solution has been computed using the real demands of the given scenario. The numerical results for the 27 considered scenarios are summarized in Table 7. In Scenario 1 (Typical Demand), the optimal controller reduces the TTS by 24.9% with respect to the no-control case, substantially decreasing the congestion and the on-ramp queues as can be seen in Fig. 4. The controller prioritizes the solving of the (dominant) bottleneck in segment 21 so the one in segment 7 is not completely solved (between minute 40 and 60).

6.4 Nominal VSL

For Scenario 1 (Typical Demand), the nominal VSL profiles and the profiles for the optimal solution are, by definition, the same and, obviously, they perform better than any other controller (like SPERT which reduces the TTS by 23.8%). However, it can be seen that the nominal VSL profiles (the optimal solution computed with the typical demands) perform suboptimally when the traffic conditions differ from the optimized ones. For example, for scenario 6, the TTS reduction is 12.0% versus 20.8% for the optimal case (computed using the real demands of the scenario).

6.5 Off-Line Computation of SPERT Parameters

The parameters of SPERT have been computed off-line based on the nominal VSL profiles obtained for the typical demands (Scenario 1) as explained in Section 4.

Table 3: Pearson correlation coefficients before minute 64 (traffic jam 1).

	VSL 2	VSL 3	VSL 4	VSL 5	VSL 6
Bottleneck 7	-0.737	-0.763	-0.791	-0.794	0
Bottleneck 21	-0.865	-0.898	-0.907	-0.856	0

Table 4: Pearson correlation coefficients after minute 64 (traffic jam 2).

	VSL 2	VSL 3	VSL 4	VSL 5	VSL 6
Bottleneck 7	-0.633	-0.632	-0.633	-0.568	-0.586
Bottleneck 21	-0.819	-0.819	-0.819	-0.848	-0.782

Firstly, the congestion shown is divided in time into two different traffic jams (one before minute 64 and the other after minute 64) since all speed limits are inactive at this time step ($V_{c,j}^{\text{nom}}(64 \cdot \frac{60}{10}) = 100 \text{ km/h}$ $\forall j$). Therefore, different thresholds $\bar{\rho}_{i,S_j,1}$ and $\bar{\rho}_{i,S_j,2}$ are used for the first part (64 minutes) and the second part of the simulation. The congestion is not divided in space because there is not a single VSL that is not activated during the entire simulation period (except for the final segments 20, 21, 22, and 23).

It has to be taken into account that during the first congestion (before minute 64), speed limits of segments 6, 7, 8, 9, 10, 11, and 18 are not active according to the nominal VSL profiles, and so they will remain inactive using SPERT. Equivalently, the VSLs of segments 20, 21, 22, and 23 will be set equal to 100 km/h (inactive) during the entire simulation since the nominal VSLs of those segments are always equal to their maximum value of 100 km/h.

Subsequently, the dominant bottlenecks have to be selected. Initially, the set of potential bottlenecks is $P_{1,1} = P_{1,2} = \{7, 14, 21\}$, including only the segments with on-ramps because, in this network, there are no reductions of the number of lanes or changes in the model parameters. The magnitude of the congestion ($MC_{j,l}$) is computed obtaining $MC_{7,1} = 0.428$, $MC_{14,1} = 0$, and $MC_{21,1} = 0.314$ for the first traffic jam and $MC_{7,2} = 0.502$, $MC_{14,2} = 0.206$, and $MC_{21,2} = 0.467$ for the second one. Therefore, segment 14 is removed as potential bottleneck, resulting in $P_{2,1} = \{7, 21\}$ for the first traffic jam. However, for the second traffic jam, segment 14 is not removed because $MC_{14,2}/MC_{21,2} > 0.1$ and $MC_{14,2}/MC_{7,2} > 0.1$. The next step involves the computation of $D_{j,l}$ for the remaining segments yielding $D_{7,1} = 0.508$ and $D_{21,1} = 0.341$ for the first traffic jam and $D_{7,2} = 0.664$, $D_{14,2} = 0.021$ and $D_{21,2} = 0.621$ for the second one. Therefore, $P_{3,1} = P_{3,2} = \{7, 21\}$ for both the first and the second traffic jam since segment 14 is also removed for the second traffic jam due to the low value of $D_{14,1}$ and $D_{14,2}$. It can be seen how $D_{j,l}$ clearly differences between bottlenecks causing congestion by themselves (segments 7 and 21) and bottlenecks receiving congestion from downstream segments (segment 14).

Finally, the segments with active speed limits in the Nominal VSL solution have to be correlated with segment 7 or 21. For the speed limits from segment 8 to segment 19, there is only potential bottleneck located downstream (bottleneck on segment 21) so this segment is selected as dominant bottleneck and used for SPERT. For the speed limits from segment 2 to segment 6 two possible downstream bottlenecks could be used: segment 7 and segment 21. This final allocation is done according to their Pearson coefficient. It has to be taken into account that a different correlation may appear for the first (before minute 64) and the second (after minute 64) traffic jam. Moreover, as indicated previously, for segment 5 to 11, there is no activation during the first traffic jam; so these correlations do not need to be computed. The Pearson correlation coefficients ($r_{V_{c,i}^{\text{nom}}, \rho_i^{\text{nom}}, l}$) obtained can be seen in Table 4. Since the values of the Pearson correlation coefficients corresponding to the bottleneck on the segment 21 are lower (closer to -1) than the ones corresponding to the bottleneck on segment 7, the bottleneck on segment 21 is chosen as dominant bottleneck for the entire set of VSLs.

The density thresholds are computed as explained in Section 4, Step 5. An example can be seen in Fig. 5, which shows the density of the dominant bottleneck (on segment 21) and the Nominal VSL of segment 16. It can be seen that the bottleneck density is 32.08 veh/(km lane) when the VSL is decreased from 100 km/h to 60 km/h in minute 28.17. Therefore, $\hat{\rho}_{16,60}$ for the first traffic jam (during the first hour) will be equal to 32.08 veh/(km lane). Subsequently, the VSL is increased again to 100 km/h in minute 48; so $\bar{\rho}_{16,100} = 32.38 \text{ veh/(km lane)}$. In this case, the nominal VSL of segment 16 never takes a value of 80 km/h for the first traffic jam so SPERT will change the speed limit of segment i between 100 km/h and 60 km/h.

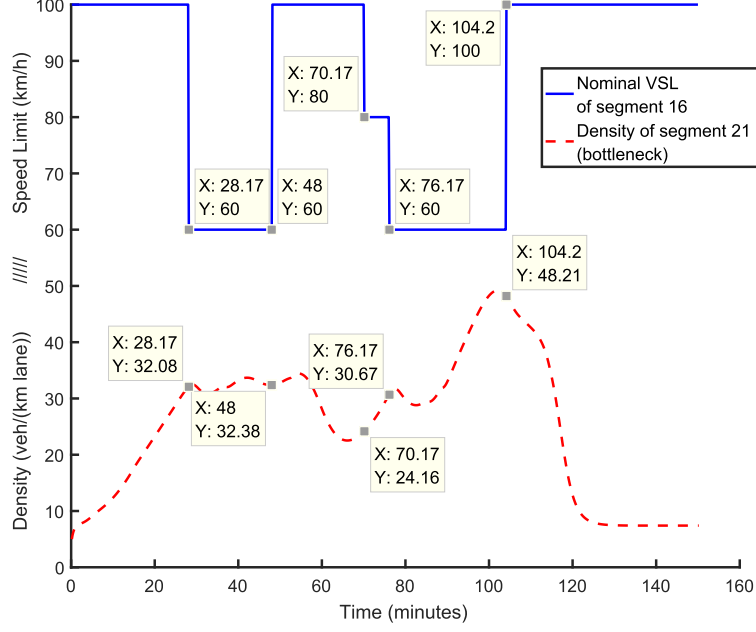


Figure 5: Dominant bottleneck density (segment 21) and VSL of segment 16.

Table 5: Density thresholds for the first traffic jam.

Segment	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
$\hat{\rho}_{i,80}$ ($\frac{\text{veh}}{\text{km lane}}$)	33.1	32.8	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$\hat{\rho}_{i,60}$ ($\frac{\text{veh}}{\text{km lane}}$)	23.2	25.5	30.0	32.8	∞	∞	∞	∞	∞	∞	25.5	27.7	32.8	27.7	32.1	32.0	∞	32.3
$\bar{\rho}_{i,80}$ ($\frac{\text{veh}}{\text{km lane}}$)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\bar{\rho}_{i,100}$ ($\frac{\text{veh}}{\text{km lane}}$)	32.8	32.8	33.4	33.4	0	0	0	0	0	0	33.0	33.4	33.1	33.4	32.4	33.0	0	33.9

For the second traffic jam, the VSL is first decreased to 80 km/h in minute 70.17 and then, in minute 76.17, it is decreased to 60 km/h. Therefore, $\hat{\rho}_{16,80} = 24.16$ veh/(km lane) and $\hat{\rho}_{i,60} = 30.67$ veh/(km lane) for the second traffic jam.

The full set of obtained density thresholds is shown in Tables 5 and 6. The density thresholds that are equal to ∞ veh/(km lane) or 0 veh/(km lane) indicate that these inequalities cannot be fulfilled for the considered VSL. For example, the VSLs from segment 6 to 11 are not activated during the first traffic jam but they are activated during the second, as can be seen in Fig. 4 and the corresponding values of the thresholds in Tables 5 and 6.

6.6 SPERT Performance

SPERT shows, for most situations, a behavior closer to the optimal solution than the Nominal VSL. In fact, for 25 of the other 26 scenarios considered, SPERT shows a higher TTS reduction than the Nominal VSL and the largest difference between the TTS obtained with SPERT and the minimum reachable TTS (optimal controller) is only 5.4%, in scenario 19.

It has to be taken into account that for some scenarios (11, 15, 25, and 27) the uncontrolled system only reaches congestion during a quite short period of time so the TTS cannot be reduced significantly (less than 1% reduction). In these cases, an incorrect use of the VSL could increase the TTS, which is, obviously, not desirable. For example, using the Nominal VSL the TTS is increased by 6.4%, 7.0%, 7.0%, 6.0%, and 8.1%, respectively. However, SPERT reacts to the decreased densities (compared with the typical case); so the

Table 6: Density thresholds for the second traffic jam.

Segment	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
$\hat{\rho}_{i,80}$ ($\frac{\text{veh}}{\text{km lane}}$)	23.1	23.1	23.1	∞	23.1	∞	23.0	∞	∞	∞	∞	∞	23.0	25.8	24.2	24.1	32.7	32.6
$\hat{\rho}_{i,60}$ ($\frac{\text{veh}}{\text{km lane}}$)	24.1	24.1	24.1	24.1	25.8	22.5	28.1	23.0	24.1	24.1	25.8	25.8	28.1	32.7	30.7	30.7	32.4	33.5
$\bar{\rho}_{i,80}$ ($\frac{\text{veh}}{\text{km lane}}$)	0	0	0	23.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\bar{\rho}_{i,100}$ ($\frac{\text{veh}}{\text{km lane}}$)	25.8	25.8	28.1	23.0	32.7	30.7	32.7	32.8	32.4	34.6	47.2	48.2	48.2	48.2	48.2	46.0	43.0	40.8

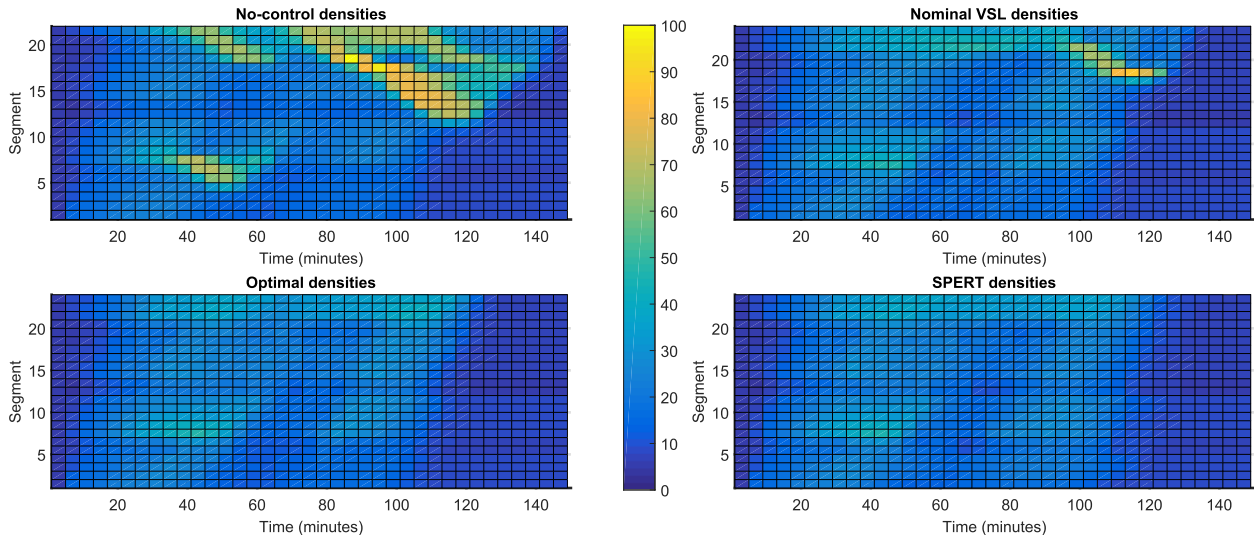


Figure 6: Density contour plots for Scenario 18.

TTS is almost not increased (0.5%, 1.0%, 1.0%, and 1.0 % increase, respectively).

In Fig. 6, the density contour plots for a representative case (Scenario 18) are shown. In this scenario, the demand of the ramp on segment 7 is reduced by 10% and the demand of the ramp on segment 21 is increased by 10%. It can be observed that the Nominal VSL are not able to completely solve congestion during the second traffic jam caused by the on-ramp located on segment 21 (because the demand on this on-ramp is higher than expected). On the other hand, the density profiles obtained using SPERT are very similar to the optimal ones, completely removing the traffic jam. Similar behaviors can be observed for the majority of the scenarios simulated.

6.7 Local MTFC Performance

A comparison with the controller proposed in [18] (Local Feedback-Based Mainstream Traffic Flow Control or Local MTFC) is also included. The parameters of the Local MTFC have been optimized, using a multi-start derivative-free optimization method, in order to maximize TTS reduction in Scenario 1. MTFC shows a good behavior for the majority of the simulated scenarios, but with a worse performance than SPERT. The obtained deviation from the optimal solution is of the same order of magnitude as the one observed in [19].

Since MTFC is designed to deal with both non-recurrent and recurrent congestion on bottlenecks and its off-line computation load is lower than with SPERT, the behavior of SPERT has to be significantly better than MTFC when dealing with recurrent congestion in order to justify its employment. However, as can be seen in Table 5, the obtained performance is considerably better using SPERT than using MTFC (10.3% vs 7.3%). Moreover, it can be seen in the results that, even with a significant deviation from the nominal solution, the behavior obtained by SPERT is still better than the one obtained by MTFC for most scenarios.

6.8 H. Logic-Based Controller

Finally, the controller proposed in [6] has been simulated and the results are also shown in Table 7. This controller also shows a suitable behavior for the simulated scenario. In fact, the obtained performance is close to, but slightly worse than the one obtained with SPERT (9.5% vs 10.3%). However, it has to be taken into account that the simulated logic-based controller uses 22 density measurements (the density of the segments with a VSL) while SPERT only uses one measurement (the density of the bottleneck).

7 Conclusions and Future Work

This paper has proposed a control algorithm (SPERT) for Variable Speed Limits (VSLs), based on the optimal solution in case of recurrent congestion, that can be applied in practice to large traffic networks.

Table 7: Numerical results.

Scenario	Demand Changes (%)			TTS Reduction (%)					
	Mainline	Ramp 1	Ramp 3	No Control	Nominal VSL	Optimal	SPERT	MTFC	Logic-based
1 (Typical Demand)	0%	0%	0%	0% (5608 veh h)	24,9%	24,9%	23,8%	15,0%	19,4%
2	+10%	0%	0%	0% (6841 veh h)	13,5%	17,6%	16,8%	11,0%	16,7%
3	-10%	0%	0%	0% (4384 veh h)	14,3%	19,3%	18,1%	16,3%	18,8%
4	0%	+10%	0%	0% (6219 veh h)	14,6%	21,2%	17,3%	14,0%	15,8%
5	0%	-10%	0%	0% (4969 veh h)	19,4%	22,2%	21,2%	17,8%	18,6%
6	0%	0%	+10%	0% (5679 veh h)	12,0%	20,8%	18,3%	12,7%	17,7%
7	0%	0%	-10%	0% (4555 veh h)	9,2%	9,4%	8,3%	0,2%	0,4%
8	+10%	+10%	0%	0% (6932 veh h)	-1,0%	2,6%	2,0%	2,1%	0,0%
9	+10%	-10%	0%	0% (6319 veh h)	14,9%	18,0%	16,9%	15,2%	16,9%
10	-10%	+10%	0%	0% (4627 veh h)	16,1%	20,3%	19,2%	16,5%	19,9%
11	-10%	-10%	0%	0% (3429 veh h)	-6,4%	0,8%	-0,5%	-2,1%	0,3%
12	+10%	0%	+10%	0% (6563 veh h)	8,7%	12,4%	9,6%	2,4%	9,2%
13	+10%	0%	-10%	0% (5601 veh h)	-4,8%	1,4%	-0,4%	-0,5%	1,0%
14	-10%	0%	+10%	0% (4408 veh h)	13,2%	17,7%	15,9%	12,4%	16,6%
15	-10%	0%	-10%	0% (3476 veh h)	-7,0%	0,2%	-1,0%	-3,5%	0,1%
16	0%	+10%	+10%	0% (6367 veh h)	14,6%	21,3%	15,9%	13,1%	16,2%
17	0%	+10%	-10%	0% (5225 veh h)	-0,8%	6,8%	3,1%	5,0%	0,3%
18	0%	-10%	+10%	0% (5232 veh h)	18,7%	22,1%	21,5%	17,9%	21,2%
19	0%	-10%	-10%	0% (3986 veh h)	0,7%	5,2%	3,1%	0,9%	1,1%
20	+10%	+10%	+10%	0% (7142 veh h)	1,2%	4,5%	3,9%	2,5%	1,4%
21	+10%	+10%	-10%	0% (6793 veh h)	-2,5%	1,2%	0,1%	0,8%	-0,7%
22	+10%	-10%	+10%	0% (6462 veh h)	15,8%	18,0%	17,3%	14,1%	17,1%
23	+10%	-10%	-10%	0% (5157 veh h)	-3,5%	0,6%	-0,5%	-2,9%	-0,2%
24	-10%	+10%	+10%	0% (4383 veh h)	9,2%	11,3%	10,7%	6,9%	11,2%
25	-10%	+10%	-10%	0% (3621 veh h)	-6,0%	0,3%	-1,0%	-3,9%	0,0%
26	-10%	-10%	+10%	0% (4313 veh h)	14,3%	19,4%	18,0%	16,0%	18,7%
27	-10%	-10%	-10%	0% (3346 veh h)	-8,1%	0,1%	-1,0%	-3,2%	-0,1%
Mean	-	-	-	-	7.2%	11.8%	10.3%	7.3%	9.5%
Standard deviation	-	-	-	-	9.58%	8.78%	8.59%	7.69%	8.74%

SPERT makes a trade-off between practical feasibility and optimality by combining advantages of optimal and easy-to-implement controllers.

The results show that SPERT approaches the performance of an optimal controller, substantially improving the performance of the off-line computed solution (Nominal VSL) and previously proposed VSL controllers like Local MTFC. More concretely, the mean TTS reduction for the 27 studied scenarios is 11.8%, 10.3%, 9.5%, 7.3% and 7.2% for the optimal solution, SPERT, a logic-based controller, Local MTFC, and Nominal VSL, respectively. On the other hand, the optimal controller has a high computational load while SPERT, the logic-based controller and MTFC are much faster and they can be implemented with a very low on-line computational load.

In future work, SPERT will be integrated in the framework of a two-level controller in order to properly regulate other kinds of congestion such as shock waves or unexpected capacity reductions.

Other possible future research topics are to consider multi-lanes macroscopic models [33], instead of the single-lane model currently used, in order to improve the performance of the nominal solution and to propose a variant of the QL-based VSL algorithm [21] that can be trained in a reasonable time for large networks.

Appendix

Table 8 is included in this appendix in order to explain, in a unified way, the variables used in this paper.

Table 8: Summary of variables.

$\rho_i(k)$	Traffic density
$v_i(k)$	Mean speed
$q_i(k)$	Traffic flow
$q_{r,i}(k)$	Traffic flow that enters the segment from an on-ramp
$\beta_i(k)$	Split ratio of an off-ramp
$V(k)$	Desired speed for the drivers
$V_{c,i}(k)$	Value of a VSL
$w_i(k)$	Queue length on a on-ramp
$D_i(k)$	Demand of an connected on-ramp
$J(k)$	Cost function
$V_{c,t}(k)$	Vector containing the VSL values
$V_{c,i}^{\text{nom}}(\cdot)$	Optimal VSL profiles computed using the typical demand
$\rho_i^{\text{NC}}(k)$	Densities of the no-control case
$\rho_i^{\text{nom}}(k)$	Density of the nominal case
$\rho_{B,j}(k)$	Density of the dominating bottleneck

References

- [1] E. van den Hoogen and S. Smulders, "Control by variable speed signs: Results of the Dutch experiment," in *Proc. 7th Int. Conf. Road Traffic Monitor. Control*, Apr. 1994, pp. 145-149.
- [2] A. Muralidharan and R. Horowitz, "Optimal control of freeways networks based on the link node cell transmission model," in *Proc. Amer. Control Conf. (ACC)*, Jun. 2012, pp. 5769-5774.
- [3] A. Hegyi, B. De Schutter, and J. Hellendoorn, "Model predictive control for optimal coordination of ramp metering and variable speed limits," *Transp. Res. C, Emerg. Technol.*, vol. 13, no. 3, pp. 185-209, Jun. 2005.
- [4] J. R. D. Frejo and E. F. Camacho, "Global versus local MPC algorithms in freeway traffic control with ramp metering and variable speed limits," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1556-1565, Dec. 2012.
- [5] R. C. Carlson, I. P. M. Papamichail, and A. Messmer, "Optimal motorway traffic flow control involving variable speed limits and ramp metering," *Transp. Sci.*, vol. 44, no. 22, pp. 238-253, 2010.
- [6] A. Darroudi, "Variable speed limit strategies to reduce the impacts of traffic flow breakdown at recurrent freeway bottlenecks," Florida Int. Univ., Miami, FL, USA, Tech. Rep., 2014.
- [7] M. Seraj, X. Wang, M. Hadiuzzaman, and T. Z. Qiu, "Optimal location identification of VSL signs for recurrent bottlenecks," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 82, no. 4, pp. 1084-1090, 2016.
- [8] Z. Li, P. Liu, W. Wang, and C. Xu, "Development of a control strategy of variable speed limits to reduce rear-end collision risks near freeway recurrent bottlenecks," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 866-877, Apr. 2014.
- [9] X.-Y. Lu and S. Shladover, "Review of variable speed limits and advisories: Theory, algorithms, and practice," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2423, pp. 15-23, 2014.
- [10] E. F. Camacho and C. Bordons, *Model Predictive Control*. Springer, 2010.
- [11] J. R. D. Frejo, A. Núñez, B. De Schutter, and E. F. Camacho, "Hybrid model predictive control for freeway traffic using discrete speed limit signals," *Transp. Res. C, Emerg. Technol.*, vol. 46, pp. 309-325, 2014.
- [12] J. R. D. Frejo and E. F. Camacho, "Feasible cooperation based model predictive control for freeway traffic systems," in *Proc. 50th IEEE Conf. Decision Control Eur. Control Conf.*, Dec. 2011, pp. 5965-5970.
- [13] H. Majid, M. Hajiahmadi, B. De Schutter, H. Abouaïssa, and D. Jolly, "Distributed model predictive control of freeway traffic networks: A serial partially cooperative approach," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Qingdao, China, Oct. 2014, pp. 1876-1881.
- [14] M. Papageorgiou, H. Hadj-Salem, and F. Middelham, "ALINEA local ramp metering: Summary of field results," *Transp. Res. Rec.*, vol. 1603, pp. 90-98, Jan. 1997.
- [15] J. R. D. Frejo, I. Papamichail, M. Papageorgiou, and E. F. Camacho, "Macroscopic modeling and control of reversible lanes on freeways," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 948-959, Apr. 2016.
- [16] A. Hegyi, S. P. Hoogendoorn, M. Schreuder, H. Stoelhorst, and F. Viti, "SPECIALIST: A dynamic speed limit control algorithm based on shock wave theory," in *Proc. 11th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2008, pp. 827-832.
- [17] A. Hegyi and S. P. Hoogendoorn, "Dynamic speed limit control to resolve shock waves on freeways-Field test results of the SPECIALIST algorithm," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2010, pp. 519-524.
- [18] R. C. Carlson, I. Papamichail, and M. Papageorgiou, "Local feedbackbased mainstream traffic flow control on motorways using variable speed limits," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1261-1276, Dec. 2011.

- [19] G. R. Iordanidou, C. Roncoli, and I. P. M. Papamichail, "Feedback-based mainstream traffic flow control for multiple bottlenecks on motorways," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 610-621, Feb. 2015.
- [20] E. R. Müller, R. C. Carlson, W. Kraus, and M. Papageorgiou, "Microsimulation analysis of practical aspects of traffic control with variable speed limits," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 512-523, Feb. 2015.
- [21] Z. Li, P. Liu, C. Xu, H. Duan, and W. Wang, "Reinforcement learning based variable speed limit control strategy to reduce traffic congestion at freeway recurrent bottlenecks," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 3204-3217, Nov. 2017.
- [22] M. Papageorgiou, I. Papamichail, A. Messmer, and W. Yibing, "Traffic simulation with METANET," in *Fundamentals of Traffic Simulation*. Springer, 2010.
- [23] C. F. Daganzo, "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transp. Res. Part B, Methodol.*, vol. 28, no. 4, pp. 269-287, Aug. 1994.
- [24] Y. Wang, M. Papageorgiou, G. Sarros, and W. J. Knibbe, "Real-time route guidance for large-scale express ring-roads," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Sep. 2006, pp. 224-229.
- [25] A. Messmer and M. Papageorgiou, "METANET: A macroscopic simulation program for motorway networks," *Traffic Eng. Control*, vol. 31, no. 8, pp. 466-470, 1990.
- [26] M. Papageorgiou, *Applications of Automatic Control Concepts to Traffic Flow Modeling and Control* (Lecture Notes in Control and Information Sciences). Springer, 1983.
- [27] A. Kotsialos, M. Papageorgiou, M. Mangeas, and H. Haj-Salem, "Coordinated and integrated control of motorway networks via non-linear optimal control," *Transp. Res. C, Emerg. Technol.*, vol. 10, no. 1, pp. 65-84, Feb. 2002.
- [28] C. Pasquale, D. Anghinolfi, S. Sacone, S. Siri, and M. Papageorgiou, "A comparative analysis of solution algorithms for nonlinear freeway traffic control problems," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 1773-1778.
- [29] J. R. D. Frejo and B. De Schutter, "A variable speed limit controller for recurrent congestion based on the optimal solution," in *Proc. Transp. Res. Board Annu. Meeting*, Jan. 2018.
- [30] *NIST/SEMATECH e-Handbook of Statistical Methods*. 2012 [Online]. Available: <http://www.itl.nist.gov/div898/handbook/>
- [31] G. A. Davis, "Estimating freeway demand patterns and impact of uncertainty on ramp controls," *J. Transp. Eng.*, vol. 119, no. 4, pp. 489-503, 1993.
- [32] M. E. Hallenbeck, J. M. Ishimaru, and J. Nee, "Measurement of recurring versus non-recurring congestion," Washington State Dept. Transp., Olympia, MA, USA, Tech. Rep., 2013.
- [33] I. Şahin and I. Altun, "Empirical study of behavioral theory of traffic flow: Analysis of recurrent bottleneck," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2088, pp. 109-116, 2008.