FREEWAY TRAFFIC CONTROL BASED ON NEURAL NETWORK ESTIMATION

Angela Di Febbraro [*] **Nicola Sacco** [*]

[*] *Department of Control and Computer Engineering - Polytechnic of Turin*
Corso Duca degli Abruzzi, 24
10129 – Torino, Italy

Abstract: Due to their characteristics, freeways appear to be ideal sites for testing new traffic regulation strategies, based on continuously improved information systems. In this framework, a particular consideration has recently been devoted to the application of neural networks (NNs) to freeway supervision and control. In this paper, the solution to a "classical" traffic control problem is tackled with. The traffic state variables given as inputs to such a problem can be computed via a macroscopic traffic model, thus requiring costly and complicated varying parameter identification, or via a significantly simpler NN filtering approach. While the second approach does not require to identify the parameters every time they change, it will be shown that the performances of such two computation procedures are comparable.

Keywords: Traffic systems, neural networks, adaptive filters.

1. INTRODUCTION

Maximising the efficiency of freeway networks would attract and satisfy its users, then contribute significantly to the traffic equalisation among the different available modes, and to a consequent better utilisation of the available infrastructures (see, for instance, Diakaki et al. (2003)). Moreover, owing to the restricted access via the input/output gates, freeways appear to be ideal sites for testing new traffic regulation strategies, based on continuously improved information systems (Papageorgiou and Kotsialos (2000)).

Traffic control problems usually require to determine both the present and the future traffic behaviour. In this framework, while the recent technology provides sensors which accurately measure the traffic variables, the future traffic behaviour can be computed or estimated by means of three main approaches:

(1) a suitable traffic model;
(2) an Extended Kalman Filter (EKF);
(3) a Neural Network Filter (NNF).

Among these approaches, NNF plays an interesting role, since it does not need to know the equations which describe the traffic dynamics. In fact, such equations contain many parameters, which have to be tuned for each considered freeway section. Then, while these tunings need the knowledge of a large amount of real traffic data, whenever they change, for instance due to the weather conditions, or accidents which can block some lanes, they have to be tuned again. In addition, such a tuning process results to be quite difficult due to the non-linear dynamics of the traffic behaviour (see Cremer and Papageorgiou (1981) for more details).

On the other hand, although the learning algorithms of NNFs also requires the knowledge of real traffic data, the NN-based prediction approach can react to the changes in the system behaviour better than the traffic models do, in the sense that NN filter can provide good traffic state predic-

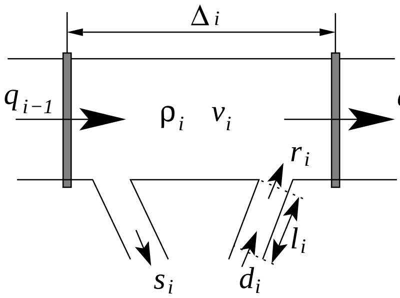

Fig. 1. The generic i-th freeway section.

tions also when the above mentioned parameters change, without any further learning procedure.

In this paper, to the end of pointing out the performances of a proposed NNF, the solution to a "classical" traffic control problem is tackled with. The traffic state variables given as inputs to such a problem will be computed via a macroscopic traffic model, thus requiring costly and difficult parameter identification, or via a significantly simpler NN filtering approach. It will be shown that the performances of such two computation procedures are comparable.

2. THE FREEWAY TRAFFIC MODEL

Freeway traffic systems can be suitably represented by means of a macroscopic non-linear second-order model based on the continuous variables traditionally describing fluid mechanics. To this end, the freeway is divided into N sections of length Δ^i, and all the aggregate variables in the generic section i, $i = 1, \ldots, N$, that is:

- the traffic volume q_k^i $[veh/h]$,
- the traffic density ρ_k^i $[veh/km]$,
- the mean vehicle speed v_k^i $[km/h]$,

are sampled in kT, $k \in \mathbb{N}$, being T the sampling period. Thus, the state equations describing the traffic density and the mean vehicle speed on each freeway section result to be:

$$\rho_{k+1}^i = \rho_k^i + \frac{\mathrm{T}}{\Delta^i}[q_k^{i-1} - q_k^i + r_k^i - s_k^i] \tag{1}$$

and

$$\begin{aligned} v_{k+1}^i &= v_k^i + \frac{\mathrm{T}}{\tau}\left[V(\rho_k^i) - v_k^i\right] + \\ &+ \frac{\mathrm{T}\zeta}{\Delta^i} v_k^i \left[v_k^{i-1} - v_k^i\right] + \\ &- \frac{\nu \mathrm{T}\left(\rho_k^{i+1} - \rho_k^i\right)}{\tau \Delta^i \left(\rho_k^i + \chi\right)} \end{aligned} \tag{2}$$

where the variables r_k^i and s_k^i give the vehicle volumes entering and leaving section i, $i = 2, \ldots, N-1$, at time $k \in \mathbb{N} \cup \{0\}$, if there are on-ramps or off-ramps. In addition, α, τ, ζ, ν, and χ are parameters depending on the freeway layout, and $V[\rho_k^i]$ is the fundamental traffic diagram

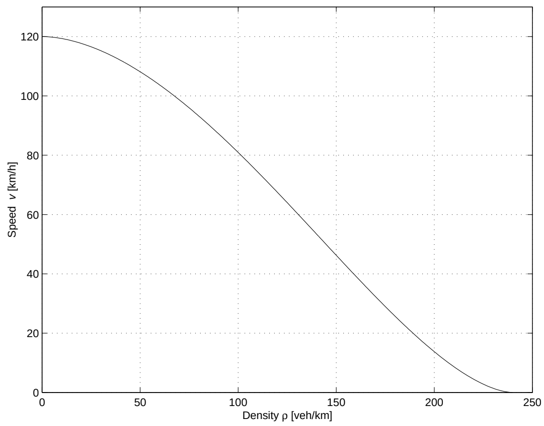

Fig. 2. Density-speed fundamental traffic diagram.

$$V(\rho_k^i) = V_f\left[1 - \left(\frac{\rho_k^i}{\rho_{max}}\right)^l\right]^m \tag{3}$$

which is depicted in Fig. 2. The parameters V_f, l, m, and ρ_{max} in (3) also depends on the freeway topology.

It is worth underlining that such parameters also depend on external (often uncontrollable) conditions such as weather conditions or accidents. As an example, when an event blocks or reduces, for instance, the number of the lanes of the freeway, then the maximum admissible traffic density ρ_{max} decreases.

Moreover, the traffic flow exiting the generic section i, $i = 1, 2, \ldots, N-1$, at each time $k \in \mathbb{N}$ results to be a dependent variable given by

$$q_k^i = \alpha \rho_k^i v_k^i + (1 - \alpha)\rho_k^{i+1} v_k^{i+1}, \tag{4}$$

whereas the off-ramps volumes s_k^i are related to the traffic volumes q_k^{i-1} through the relationship

$$s_k^i = \gamma_k q_k^{i-1}, \quad 0 < \gamma_k < 1. \tag{5}$$

By substituting relations (5) and (4) into (1), it turns out that

$$\begin{gathered} \rho_{k+1}^i = \rho_k^i + \frac{\mathrm{T}}{\Delta^i}[\alpha(1-\gamma_k)\rho_k^{i-1}v_k^{i-1} + \\ +(1 - 2\alpha + \alpha\gamma_k - \gamma_k)\rho_k^i v_k^i + \\ -(1-\alpha)\rho_k^{i+1}v_k^{i+1} + r_k^i], \end{gathered} \tag{6}$$

Finally, the equations of the dynamics of the first and the last sections are

$$\begin{matrix} \rho_{k+1}^i = \rho_k^i \\ v_{k+1}^i = V[\rho_k^i] \end{matrix} \quad i = \{1, N\}. \tag{7}$$

Thus, the model takes on the standard form

$$\begin{matrix} \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{y}_k = g(\mathbf{x}_k) \end{matrix} \tag{8}$$

where:

- $\mathbf{x}_k \triangleq \mathrm{col}\left[\rho_k^1, \ldots, \rho_k^N, v_k^1, \ldots, v_k^N\right]$ is the state vector;

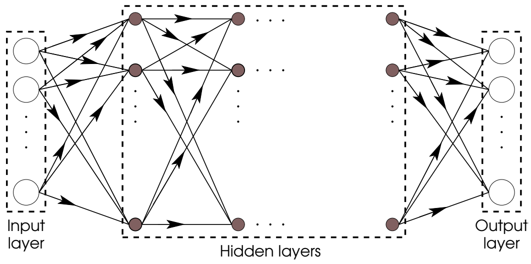

Fig. 3. A neural network.

- $\mathbf{u}_k \triangleq \mathrm{col}\left[r_k^1, \ldots, r_k^{N-1}, s_k^1, \ldots, s_k^{N-1}\right]$ is the input vector;
- $\mathbf{y}_k \triangleq \mathrm{col}\left[q_k^1, \ldots, q_k^{N-1}\right]$ is the output vector.

In this representation, it is clear that r_k^i, $i = 2, \ldots, N-1$, are the variables which allow to control and optimise the traffic behaviour, expressed by the state $\mathbf{x}_k$. Obviously, the more accurate is the prediction of the traffic variables, the more effective are the workable control and optimisation strategies.

3. NEURAL NET FILTERING APPROACH

3.1 Basics of Neural Networks

A neural network (NN) (see, for instance, Haykin (1999) for details) consists of a great number of local independent elaboration elements, i.e., neurons, which have several inputs and a single output. Each neuron has its own transfer function, which rules the local elaboration giving an output based on the present inputs and on a set of weights applied to each input. These weights are different for each neuron and their choice is a major problem in designing a NN. It is worth noting that each neuron works independently and, usually, asynchronously from each other, guaranteeing a great parallelism in the net.

Formally, each neuron i is characterised by a threshold θ^i, by an internal state S_k^i at each discrete time k, $k \in \mathbb{N}$, and by the weights $w_{i,j}$ of the connections with the other neurons of the net. Then, the dynamics of a NN is given by the *activation law*, which updates the state of each neuron, and by a *learning algorithm*, which updates the weights of the links.

In formulas, the next state is

$$S_{k+1}^i = F[P_k^i] = \frac{1}{1 + e^{-P_k^i}} \tag{9}$$

where $F[P_k^i]$ is the above mentioned *transfer function* and

$$P_k^i = \sum_{j=1}^{N} [w_{ij} \cdot S_k^j - \theta^i] \tag{10}$$

is the internal potential, for all $k \in \mathbb{N}$.

Several NN architectures are possible:

(1) *single layer architecture*: the NN consists in a single layer of neurons, which directly evaluate the outputs based on their inputs;
(2) *multiple layer architecture*: there is an input layer, which stores the inputs, and an output layer, which evaluates the output. In addition, there are several hidden layers between the input and output ones. Based on the kind of connections among the neurons, multiple layer NNs can be divided in:
    (a) *feedforward networks*, as the one depicted in Fig. 3, where the connections are only directed from the input layer towards the output layer through the hidden layers. The NNF described in this paper is based on a feedforward architecture;
    (b) *feedback networks*, where there are some connections from the output layer towards the input layer.

Finally, the learning algorithm updates the weights of the links on the basis of the error between the NN output and the measured output of the real system. In particular, in this work, the *Error Back Propagation* algorithm (see Rumelhart et al. (1986) for a more detailed description) is implemented. Such an algorithm updates the weights of the connections between the layers, starting from the output layer and back towards the input layer, through the hidden layers.

3.2 Neural network adaptive filter

In this section, some details about the NN predictor proposed in this paper will be given. To this aim, consider the NNF reported in Fig. 4. In such a figure it is possible to note that three NNs are implemented:

- the *state predictor* NN, which gives the prediction $\hat{\mathbf{x}}_{k+1|k}^{NN}$ of the state whereas it receives, as inputs, the vector

  $$\mathbf{U}_k = \mathrm{col}[\mathbf{u}_k, \mathbf{u}_{k-1}, \ldots, \mathbf{u}_{k-n_u}],$$

  which gathers the latest $n_u + 1$ input values, the output vector

  $$\mathbf{Y}_k = \mathrm{col}[\mathbf{y}_k, \mathbf{y}_{k-1}, \ldots, \mathbf{y}_{k-n_y}],$$

  which gathers the latest $n_y + 1$ output values, and, finally, the latest state value $\hat{\mathbf{x}}_{k|k}^{NN}$;
- the *output predictor* NN, which gives the prediction $\hat{\mathbf{y}}_{k+1|k}^{NN}$ of the output, whereas it receives, as inputs, the vector $\mathbf{U}_k$, the output vector $\mathbf{Y}_k$ and, finally, the state value $\hat{\mathbf{x}}_{k+1|k}^{NN}$ predicted by the state predictor NN;
- the *state filter* NN gives, at $k+1$, the updated prediction of the state $\hat{\mathbf{x}}_{k+1|k+1}^{NN}$ receiving as inputs the state $\hat{\mathbf{x}}_{k+1|k}^{NN}$ predicted by the state

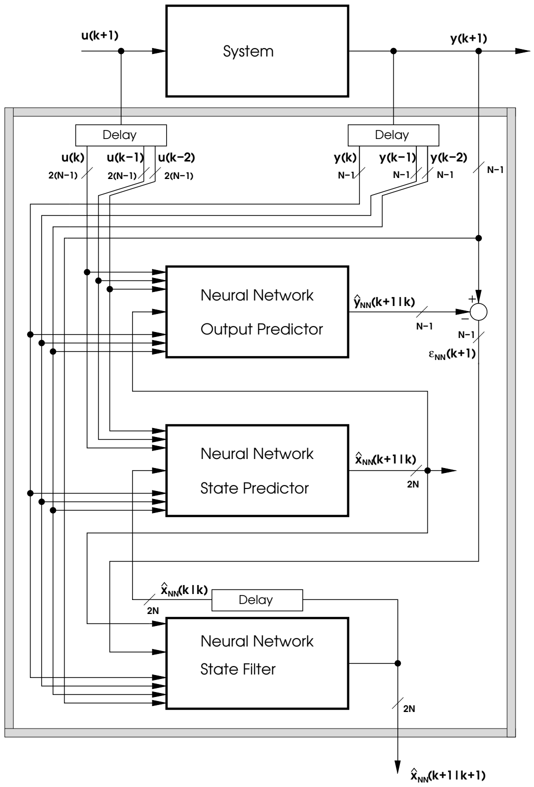

Fig. 4. Neural network filter.

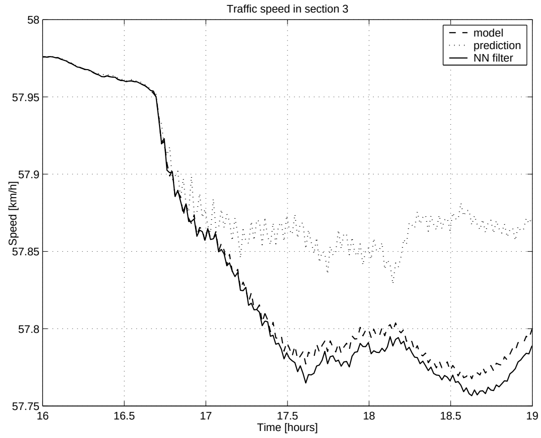

Fig. 5. Predicted and filtered traffic speed.

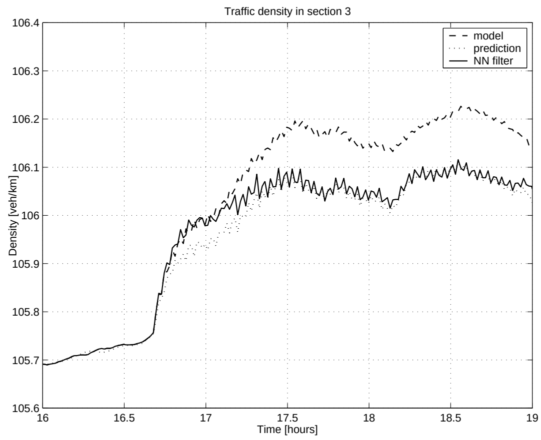

Fig. 6. Predicted and filtered traffic density.

predictor, the vector $\mathbf{Y}_k$, the actual output y_{k+1}, and the innovation term

$$\epsilon_{k+1} = \mathbf{y}_{k+1} - \hat{\mathbf{y}}^{NN}_{k+1|k}. \tag{11}$$

When the measurements of the output variables are not available for some reason, the innovation term ϵ_k in (11) can not be computed. As a consequence, the NN state filter at the bottom of the Fig. 4 can not work and then, the resulting filter, which simply consists of the state and output predictor, works as an "open loop" predictor without information about the estimation error. It is reasonable to expect that the performances of such a predictor will be worse than the performances of the complete filter.

To the end of understanding the differences between the two implementations, consider the graphs in Fig. 5 and Fig. 6, where the traffic speed and the traffic provided by the model are compared with those given by the NNF with and without the measures of the output variables. As expected, the performances of the predictor are worse than the performances of the complete filter.

On the other hand, if the prediction horizon is quite short, it is reasonable to expect that the performances of the two implementations are comparable, as pointed out in Fig. 5 and Fig. 6.

4. THE CONTROL PROBLEM

In this section, the problem of clearing possible freeway congestions is stated as a Finite-Horizon (FH) problem.

With the aim of simplifying the notation, let us introduce the set $\mathcal{S}$ of the N freeway sections, the set $\mathcal{I}$ of the freeway sections which have on-ramps and off-ramps and the set of the time instants $\mathcal{K} = \{0, 1, \ldots, K-1\}$.

Then, consider the problem of minimising the total time spent by vehicles in the freeway system. In this framework, the control variables result to be the on-ramp vehicle volumes, whereas the control strategy consists of limiting the number of vehicles entering the freeway, with the aim of reducing the congestion.

In doing so, a suitable cost function is given by the sum of the total travel time and the waiting time on the on-ramps, and takes on the form

$$J = \sum_{k=1}^{K} T \underline{\rho}_k \underline{\Delta} + \sum_{k=1}^{K} T \underline{l}_k^T \underline{e} \tag{12}$$

being $\underline{\Delta} \triangleq \mathrm{col}[\Delta^i,\, i = 1, \ldots, N]$, and $\underline{e} \triangleq \mathrm{col}[e_i = 1,\, \forall\, i \in \mathcal{I}]$.

Moreover, $\underline{l}_k$ is the vector which gathers the queues on the on-ramps, which are given by the equation

$$l_{k+1}^i = l_k^i + T(d_k^i - r_k^i), \quad \forall\, k \in \mathcal{K},\, i \in \mathcal{I} \quad (13)$$

where d_k^i is the demand for access to the ramp in section i at time kT. Analogously, the equation

$$h_{k+1}^i = h_k^i + T(q_k^{i-1} - q_k^i), \quad \forall\, k \in \mathcal{K} \quad (14)$$

where h_k is the queue length, whereas q_k^{i-1} and q_k^i are the traffic flows entering and exiting section i, respectively, gives the queue length in each section i, $i = 2, \ldots, N$.

The problem (12) is subject to constraints (2), (6), (13), and

$$\begin{array}{ll} 0 \le \rho_k^i \le \rho_k^{\max}, & \forall i \in \mathcal{S}, \forall k \in \mathcal{K} \\ 0 \le v_k^i \le v_k^{\max}, & \forall i \in \mathcal{S}, \forall k \in \mathcal{K} \\ 0 \le l_k^i \le l_k^{\max}, & \forall i \in \mathcal{I}, \forall k \in \mathcal{K} \\ r_k^{i,\min} \le r_k^i \le r_k^{i,\max}, & \forall i \in \mathcal{I}, \forall k \in \mathcal{K} \end{array} \quad (15)$$

where

$$\begin{aligned} r_k^{i,\min} &= \max\left\{ r^{i,\min}, d_k^i - \frac{1}{T}(l^{i,\max} - l_k^i) \right\} \\ r_k^{i,\max} &= \min\left\{ r^{i,\max}, d_k^i + \frac{1}{T} l_k^i \right\} \end{aligned} \quad (16)$$

being $r^{i,\min}$ and $r^{i,\max}$ fixed parameters dependent on the road characteristics.

As regards the solution to problem (12), a classical steepest descent method is considered. In doing so, the constraints (15) of the problem have been removed, then considering the approximated cost function

$$\begin{aligned} \tilde{J} = J &+ \sum_{i \in \mathcal{S}} \sum_{k=1}^{K} \Big\{ K_\rho \{ [\max(0, -\rho_k^i)]^2 + \\ &+ [\max(0, \rho_k^i - \rho_k^{\max})]^2 \} + \\ &+ K_v [\max(0, v_k^i - v_k^{\max})]^2 \Big\} + \\ &+ \sum_{i \in \mathcal{I}} \sum_{k=1}^{K} \Big\{ K_l \{ [\max(0, -l_k^i)]^2 + \\ &+ [\max(0, l_k^i - l_k^{\max})]^2 \} + \\ &+ K_r \{ [\max(0, r_k^{i,\min} - r_k^i)]^2 + \\ &+ [\max(0, r_k^i - r_k^{i,\max})]^2 \} \Big\} \end{aligned} \quad (17)$$

where the new terms represent a Lagrangian relaxation of the constraints (15), and K_ρ, K_v, K_l, and K_r are large constant positive scalars. It is worth noting that such penalty terms are differentiable with respect to their arguments, allowing the use of the gradient method to find the optimal solution.

Finally, it is worth remarking that the computation of the cost function (17) requires the knowledge of the future values of the traffic variables, which can be obtained by means of a traffic model, as in Di Febbraro et al. (2001), or by means of the proposed NNF.

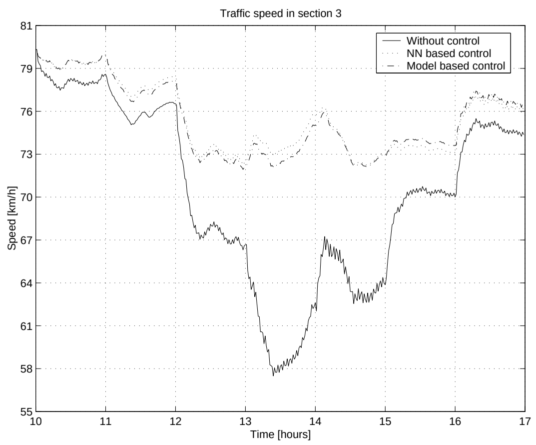

Fig. 7. Traffic speed in section 3.

Both these computation methods have advantages and disadvantages. While from a performance point of view both the procedures give satisfactory results, as shown in the next section, the NN approach results to be worse from a computational point of view. In addition, while the use of the second order model described in Sec. 2 requires the tuning of several time-varying parameters (see Cremer and Papageorgiou (1981) for more details), which needs a large amount of real traffic data, the use of the NN adaptive predictors requires a learning procedure of the NNF, which also requires a large amount of data.

5. SIMULATION RESULTS

In this section, a comparison between the performances of the problem (17) solved by computing the traffic dynamics by means of the model, with respect to the traffic dynamics predicted by a NN approach, is reported.

The test site here considered consists in 5 sections of a three-lane freeway near to a congestion. In such a situation an increase in the traffic density and, as a consequence, a decrease of the traffic speed, as reported in Fig. 7, will cause a queue if in at least a section the traffic volume exceeds the freeway capacity.

In such situations, the classic control strategy implemented in this paper simply consists in limiting the number of vehicles entering the freeway through the on-ramps, as said above. Note that, although such a regulation strategy results in the rise of vehicle queues on the on-ramps, if the control variables are suitably chosen, the total amount of time spent by vehicles on the freeway can be reduced.

In the graphs of Fig. 5, Fig. 8, Fig. 9, and Fig. 10, the traffic speed and the vehicle queues relevant to a congested situation optimised by means of problem (17) are reported. In such

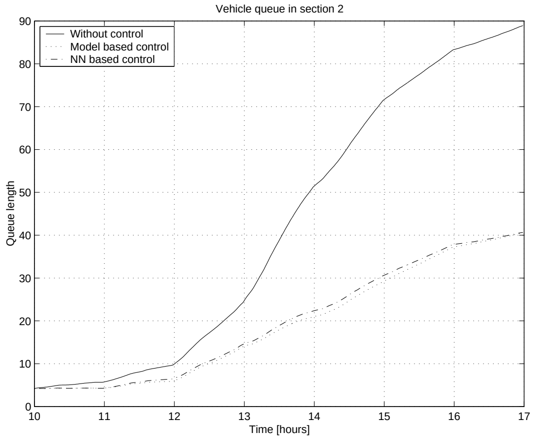

Fig. 8. Queue in section 2.

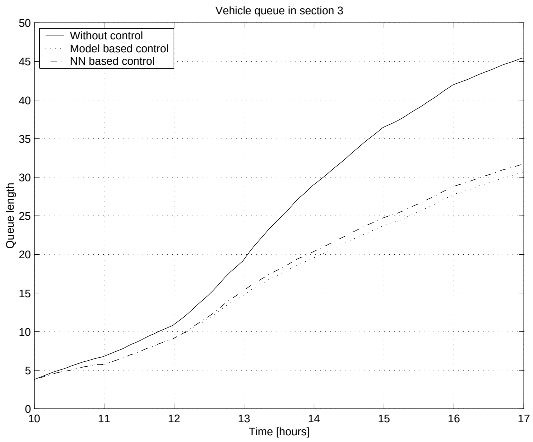

Fig. 9. Queue in section 3.

figures, it is possible to compare the performances of the control problem when the traffic dynamics is determined by the second order model, to its performances when using the NN predictor. In particular, it is easy to note that the two approaches almost give the same results. In effect, due to the particular structure of the problem, if the time horizon on which the problem is solved is short enough, the performances of the predictor are similar to those of the complete filter, as said in the above Sec. 3.2.

At the end, it is worth noting that the NN learning algorithm is based on traffic data measured in a situation without congestions, and made up via an off-line procedure. That results in the main advantages of the NN approach, i.e., in the fact that the performances of the predictions do not deteriorate significantly when the state conditions change.

6. CONCLUSIONS

The main contribution of this work stands in the comparison of the performances of a selected control problem, in the two frameworks in which the necessary input traffic data are computed by

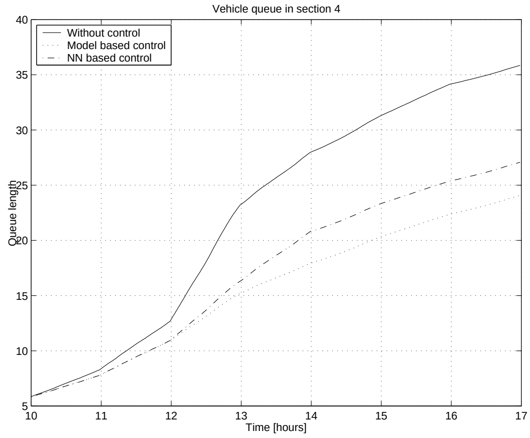

Fig. 10. Queue in section 4.

a classical second order macroscopic traffic model, or by a NN adaptive predictor.

A major advantage of the proposed predictor relies in the fact that there is no need to know the model of the traffic dynamics, which is characterised by several varying parameters that requires a quite difficult tuning. On the other hand, from a computational point of view, the proposed approach results heavier than the approach based on the analytical traffic model.

Such a comparison has proved through the experimental results reported in Sec. 5 that the NN approach is still effective in those situations when traffic control is necessary, i.e., in case of traffic congestions, without any new learning procedure.

REFERENCES

M. Cremer and M. Papageorgiou. Parameter identification for a traffic flow model. *Automatica*, 17:837–843, 1981.

A. Di Febbraro, T. Parisini, S. Sacone, and R. Zoppoli. Neural approximations for the feedback optimal control of freeway systems. *IEEE Trans. on Vehicular Technology*, 50:302–313, 2001.

M. Diakaki, C. Dinopoulou, V. Kotsialos, M. Papageorgiou, and A. Y. Wang. Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12):2043–2067, December 2003.

S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, Upper Saddle River, NJ (USA), 1999.

M. Papageorgiou and A. Kotsialos. Freeway ramp metering: An overview. *Proc. IEEE Conf. on Intelligent Transportation Systems, Dearborn (MI), USA*, pages 228–239, October 2000.

D.E. Rumelhart, G.E. Hinton, and R.J. Williams. *Learning Internal Representation by Error Propagation*, volume 1, Foundation of *Parallel Distributed Processing: Explorations in Microstructure of Cognition*, pages 318–362. MIT Press, Cambridge, MA (USA), 1986.