【土木計画学研究・講演集 No.19(2) 1996年11月】

Estimation of Traffic Flow Variables Using Neural-Kalman Filtering Techniques

Nasser POURMOALLEM*, Takashi NAKATSUJI**

1. INTRODUCTION

Macroscopic traffic flow models, which are based on a hydrodynamic theory of traffic flow phenomena, contain several traffic variables and can therefore describe extensive traffic states in great detail. By combining such a macroscopic model with a Kalman filtering technique[1], several methods have been proposed for estimating traffic states on freeways and urban streets, as shown by Gazis *et al.*[2],and so on.

Incorporating the Payne-type macroscopic model[3] with the Kalman filter, Cremer[4] presented a feed-back method for estimating traffic states on a freeway. In the method, traffic density and space mean speed that were estimated by the macroscopic model were adjusted so that the flow rate and time mean speed at the observation points would coincide with actually observed ones. This method is very effective even for congested traffic states because it adopts density and space mean speed as the state variables. And it is very promising for future route guidance systems because it can estimate the traffic states on real time.

This paper aims to investigate the ability of the neural-Kalman filtering method to estimate traffic states on a freeway. Intending to extend the Cremer method, we examined how accurately a neural network model could describe the state and observation equations and realize a model parameter that was dependent on traffic states.

Cremer assumed that flow rate can be expressed as the weighted average of products of density and space mean speed on the adjacent segments. Similarly, time mean speed is the weighted average of the space mean speeds on those segments. That is, the observation equations were linearly described. However, actual traffic flows are not so simple. In particular, when traffic is in a congested state, the relationships would be very complicated. So, it is necessary to develop a non-linear relationship in the observation equations. In addition, in the Cremer method, only the traffic states in the nearest upstream and downstream segments are employed in the relationships. This means that the estimation precision depends on how long the segments are. To reflect the traffic states in the segments which are located farther, we need to modify the model, in which the traffic states in any number of the segments are incorporated into the observation equations, if necessary. Moreover, the traffic states at a given point are strongly influenced by the states in the upstream links when the traffic is in a free flow state, whereas they are affected by the downstream states when the traffic is heavily congested. Any constant parameter would not reflect such phenomena precisely. As mentioned above, we introduced a parameter that was negatively related to traffic density. In this way, some model parameters should be dependent on traffic states. However, such parameters would make the estimation procedure very complicated because the differential operations required in the Kalman filter would make it almost impossible.

Neural network models[5] have some promising abilities: They can accurately describe non-linear phenomena; they can organize their structures flexibly according to the observed data; also, they can deal with any kind of numbers, not only quantitative numbers, but also qualitative numbers, and even fuzzy numbers. When they are applied to a dynamic estimation problem, they can easily establish a steady non-linear relationship between the input and output signals. They require no preliminary knowledge of both the state and the observation equations.

To cope with the above problems, we tried some new approaches to the Cremer method. That is, we redefined the method based on a neural network model; first, we described the observation equations using a neural network model in order to establish a steady non-linear relationship between the state and the observation variables. In addition, we expressed the state equations, too, using another neural network model. The introduction of the neural network models made it possible to employ any parameters that were dependent on traffic states because the differential matrices in the Kalman filter were easily derived. As a result, this new method improved the estimation precision. We referred this new approach to the neural-Kalman filtering method.

2. THEORETICAL BACKGROUNDS

(1) Macroscopic Traffic Flow Model[4]

We divide a road on a freeway into several segments. The Payne-type model describes the traffic flow dynamic is employed in the Original Cremer model (OC model), the dynamic equations are defined as follows:

$$c_i(k+1) = c_i(k) + \frac{\Delta t}{\Delta L_i}\left[q_{i-1} - q_i + r_i - s_i\right]_{(k)} \quad (1)$$

$$v_i(k+1) = v_i(k) + \frac{\Delta t}{\tau}\left[V(c_i) - v_i\right]_{(k)} + \frac{\Delta t}{\Delta L_i}\left[v_i\left(v_{i-1} - v_i\right)\right]_{(k)} + \frac{\nu}{\tau}\frac{\Delta t}{\Delta L_i}\left[\frac{c_i - c_{i+1}}{c_i + \kappa}\right]_{(k)} \quad (2)$$

$$q_i(k) = \left[\alpha c_i v_i + (1-\alpha)c_{i+1}v_{i+1}\right]_{(k)} \quad (3)$$

$$w_i(k) = \left[\alpha v_i + (1-\alpha)v_{i+1}\right]_{(k)} \quad (4)$$

where $c_i(k)$ is the density of segment i at time k, $v_i(k)$ is the space mean speed, $q_i(k)$ is the flow rate, and $w_i(k)$ is the time

Keywords: Macroscopic traffic flow model, Cremer model, Kalman filter, Neural network model

* Member of JSCE, Doctoral Candidate,
** Member of JSCE, Dr. of Eng., Associate Professor,
Dept. of Civil Eng., Faculty of Eng., Hokkaido University,
Kita-13, Nishi-8, Kita-Ku, Sapporo, 060, JAPAN,
Tel.: (+81)-11-706-6215, E-mail: nasser@traf3260.civil.hokudai.ac.jp
Fax.: (+81)-11-726-2296, E-mail: naka@civil.hokudai.ac.jp

mean speed. $r_i(k)$ and $s_i(k)$ are possible entrance and exit ramp flow rates. ΔL_i is the segment length and Δt is the time interval of simulation. τ, ν, κ, and α are the model parameters. $V(c_i(k))$ in Eq.(2) is the steady-state speed, which is defined by a density-speed characteristics (k-v) curve[6]:

$$V\left(c_i(k)\right) = v_f\left[1 - \left(\frac{c_i(k)}{c_{max}}\right)^{(l-1)}\right]^{\left(\frac{1}{m-1}\right)} \tag{5}$$

where v_f is the free speed, c_{max} is the jam density, m and l are the sensitivity factors.

(2) Kalman Filter Model [1]

Choosing $c_i(k)$ and $v_i(k)$ as the state variable vector x_k, and as the observation variable vector y_k, we defined the following Kalman filter (KF)[1]:

$$x_{k+1} = f\left(x_k\right) + \xi_k \tag{6}$$

$$y_k = g\left(x_k\right) + \zeta_k \tag{7}$$

We linearize these equations as following:

$$\Delta x_{k+1} = \Phi_k \Delta x_k + \xi_k \tag{8}$$

$$\Delta y_k = \Psi_k \Delta x_k + \zeta_k \tag{9}$$

where Δ is the difference of vectors, ξ_k and ζ_k are the noise vectors. $\Phi_k = \partial f/\partial x$ is the dynamic matrix and $\Psi_k = \partial g/\partial x$ is the observation matrix. Calculating Φ_k and Ψ_k step by step, we can correct the state variables every time we obtain the newly observed data y_k:

$$\hat{x}_k = \tilde{x}_k + K_k\left(y_k - \tilde{y}_k\right) \tag{10}$$

where $\tilde{x}_k = f(\hat{x}_{k-1})$ and $\tilde{y}_k = g(\tilde{x}_k)$. The vector $\tilde{x}_k$ and $\tilde{y}_k$ are referred to as the one-step predictor of x_k and y_k, and $\hat{x}_k$ as the filtered estimate of x_k. K_k is Kalman gain matrix.

(3) Variant-Weighting Factor Model[7]

The OC model assumes a constant weighting factor in Eqs.(3) and (4). However, when traffic is in a free-flow state, flow rate and time mean speed at a given point are mainly dominated by the traffic states in the upstream, whereas when traffic is in a heavy state, they are largely influenced by the states in the downstream because some growing congestion generated in a downstream segment propagates upwards. A constant weighting factor cannot describe such phenomena. We introduced a weighting factor that was dependent on density:

$$\alpha\left(c_i(k)\right) = e^{-\beta c_i(k)} \tag{11}$$

where β is a curvature in the range of 0.0 to 1.0. We called this model a variant-weighting factor model (VWF model). Since this function decreases monotonously with density, it can represent the above-mentioned traffic flow phenomena very well. It should be noted here that the introduction of such a factor would make the structure of both the state and observation equations complicated. Consequently, it would become burdensome to differentiate the equations and derive the matrices Φ_k and Ψ_k. This is why we introduced a neural network model in this paper.

(4) Multiple Section Method[7]

We extended the OC model so that we were able to treat a road section where there are any number of observation points. The original model is applicable to a single road section where traffic data are observed only at both/either entrance and/or exit of the section. When the model is applied to a long road section, in which several observation points are located inside, we have to divide it into several subsections at every observation point. Since this subdivision not only interrupts the propagation of traffic flow over the whole section, but also increases the frequency of extrapolations in Eqs. (3) and (4), the estimation precision would be inevitably decreased. We treated such a road section as a single section.

We redefined the dynamic equations so as to correspond to the observation condition. For example, for the flow rate $q_i(k)$ in Eq.(1), we used the actually observed ones. This inevitably required the redefinition of the matrices of Φ_k and Ψ_k. We called this generalized model the multiple section model (MS model).

3. NEURAL-KALMAN FILTER

(1) Neural Network Model[5]

We used a multilayer neural network model as shown in **Fig.1**, which consists of three layers;

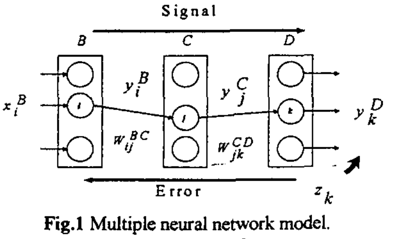


Fig.1 Multiple neural network model.

an input layer *(B)*, a hidden layer *(C)*,, and an output layer *(D)*. x_i^B represents an input signal and y_i^B an output signal. The output signal y_k^D is calculated as follows:

$$y_k^D = h\left(\sum_j W_{jk}^{CD} h\left(\sum_i W_{ij}^{BC} h\left(x_i^B\right)\right)\right) \tag{12}$$

where $h(x)$ is the sigmoid function. For adjusting synaptic weights, we need target signals that are given by theoretical equations or observed data. We repeated the back-propagation operations[5] until the following average squared sum of the between output signal y_k^D and target single z_k became sufficiently small:

$$J = \frac{1}{N_D}\sum_k^{N_D}\left(y_k^D - z_k\right)^2 \tag{13}$$

where N_D is the number of neurons in the output layer.

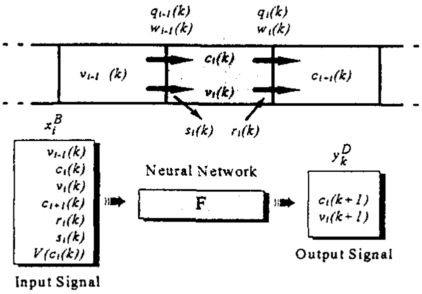
It should be noted here that it is very easy to produce the derivative of an output single y_k^D to an input single x_i^B because Eq. (12) is definitely defined by analytical functions. It follows:

$$\frac{\partial y_k^D}{\partial x_i^B} = y_k^D\left(1 - y_k^D\right)\sum_i W_{jk}^{CD} y_j^C\left(1 - y_j^C\right)W_{ij}^{BC} \tag{14}$$

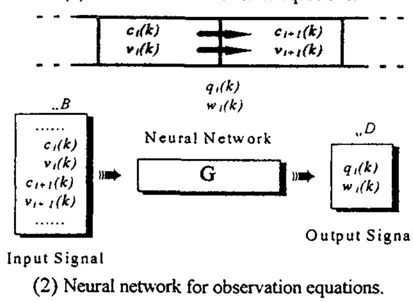
As will be stated later, this derivative function constitutes the components of matrices Φ_k and Ψ_k.

First, we developed a neural network for the state equation of Eqs.(1) and (2), as shown in **Fig.2(1)**. We emulated the basic structure of the equation. Traffic variables on the right sides were used as input signals, while variables on the left sides were used as output signals. Although the average speed $V(c_i(k))$ in Eq.(5) was dependent on $c_i(k)$, we treated it as an input signal because it contains some independently-determined parameters. We produced the target signal using Eqs. (1) and (2). It should be noted here that the neural models here were used only for estimating the matrix Φ_k because Eqs.(1) and (2) could accurately estimate the traffic states. Second, we developed another neural network for the observation equations, as shown in **Fig.2(2)**. Although we emulated the basic structure of Eqs.(3) and (4), the state variables not only in the nearest segments but also in the more distant segments in both the upstream and the downstream segments were employed as input signals. Naturally, the target signal came from the actually observed data. The neural models here were used not only for estimating the observation variables but also for defining the matrix Ψ_k.

We produced both the neural network for each segment. Preparing a lot of target signals for extensive traffic conditions in advance, we adjusted the synaptic weights so that the difference between the output and the target signals were minimized. The completion of the training of synaptic weights makes it possible produce the components of matrices Φ_k and Ψ_k.



(1) Neural network for state equations.



(2) Neural network for observation equations.

Fig.2 Neural network modeling of state and observation equations.

(2) Neural-Kalman Filter

In conventional Kalman filters, both the state and the observation equations have to be analytical functions. Here, we proposed an alternative filter, in which the equations were described by neural network models. This made it possible to construct the observation equations as precisely as the observed data were. The introduction of a model parameter, such as the weighting factor in Eq. (11), that was dependent on traffic states required no difficulties in deriving the matrices Φ_k and Ψ_k because the synaptic weights were already adjusted so as to reflect the effects and Eq. (14) easily produced them. Although it is beyond the scope of this paper, since neural network models can deal with even qualitative numbers as the input signals, it would be possible to represent any local characteristics of each segment into both the state and the observation equations. We referred this new approach to the neural-Kalman filtering method.

Fig.3 is the block diagram to estimate the traffic states using the neural-Kalman filter. The painted boxes depict what are different from the conventional Kalman filter. However, the fundamental algorithm is identical to it. First, based on the estimates $\hat{x}_{k-1}$ at the previous time k-1, we predicted the state variable $\tilde{x}_k$ at time k using Eqs.(1) and (2). In this process, the flow rate and the time mean speed at the points where traffic data were not observed were also estimated by the neural network g but not Eqs. (3) and (4). Prior to obtaining the actual observed data y_k, we estimated $\tilde{y}_k$ using the neural model g. At the same time, using the neural derivatives of Eq.(14), we calculated the matrices Φ_k and Ψ_k, which determined the Kalman gain K_k. Then we corrected the predicted estimates $\tilde{x}_k$ and obtained the new ones $\hat{x}_k$ according to Eq.(10).

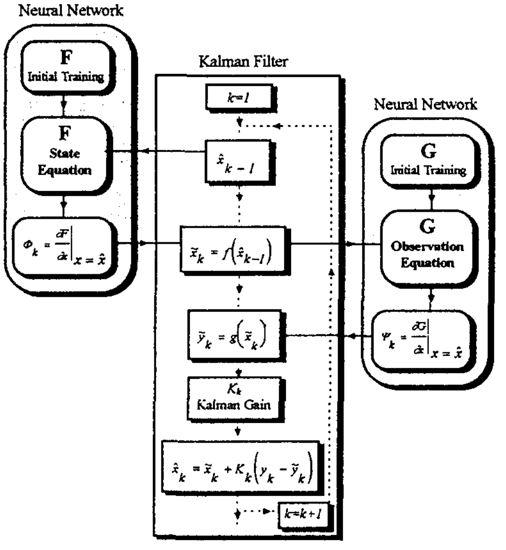


Fig.3 Block diagram of neural-Kalman filtering method.

4. NUMERICAL EXPERIMENT

(1) Traffic Data

The observed data used here came from a road section, which was 5130 meters long with two on-ramp and three off-ramp, on the Yokohane Line of the Metropolitan Expressway in Tokyo. We used the traffic data from Oct. 28 to Nov. 1 in 1993. We defined three subsection, which were divided 3 or 4 segments and a checking point (CP). We assumed that traffic data were only at four observation points (OP), as shown in **Fig.4**.

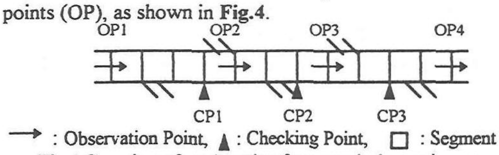


Fig.4 Overview of road section for numerical experiments.

(2) Initial Training

(a) Observation Equations

We trained the Neural network model for the flow rate and time mean speed equations. We supposed two types of neural structures, as shown in **Fig.5**. **Fig.6** depicts the average RMS errors of output signals for 60 checking patterns at four observation points for each type of the NN model. We can see that the NN model of type 2 gives smaller RMS errors for all the observation points. This means that by incorporating the traffic states of the two adjacent segments in both upstream and downstream into model, we can estimate the observation variables more precisely.

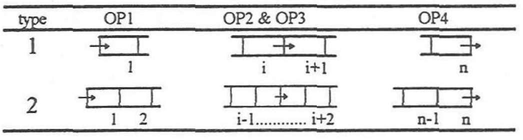


Fig.5 Types of neural network models of observation equations.

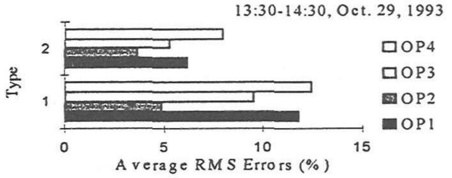


Fig.6 RMS errors of neural observation systems for checking data.

(b) State Equations

We trained the NN model for the state equations of Eqs. (1) and (2) to produce the derivative of the matrix Φ_k. According to whether segments have an on-ramp or an off-ramp, we classified the segments into three types, as shown **Fig.7**. That is, the number of neurons in an input layer is 7 for segments that have no on- and off-ramps, and 8 for segments that have either on- or off-ramp. In this analysis, we always allocated five neurons to the intermediate layer. **Fig.8** depicts the average RMS errors of output signals for 120 sets of checking data at all the segments. We can see that the errors are small enough except for a few segments where the errors exceed 10%. It should be noted that the estimates are not corrected yet by the actually observed data.

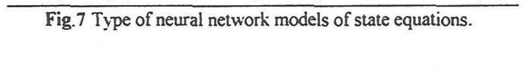


Fig.7 Type of neural network models of state equations.

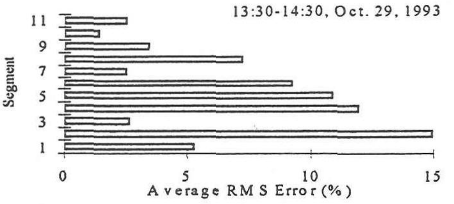


Fig.8 RMS errors of neural state equations for checking data.

(3) Estimation of Traffic States

Fig.9 shows the comparison of the average RMS errors of flow rate at three checking points for the OC and the NKF models. The NKF model produced much better estimates for all the data sets than the OC model. And the RMS errors are sufficiently small. Moreover, the deviation of RMS errors of the NKF model was smaller than that of the OC model.

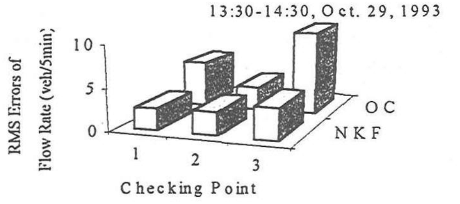


Fig.9 Comparison of average RMS errors of flow rate evaluated by the OC and the NKF models.

5. CONCLUSIONS

The major findings are summarized as follows:

(1) Integrating a NN model into a KF, we proposed a procedure to estimate the traffic states on a freeway road.

(2) The NN models for describing state equations and observation equations made it possible to easily produce the derivative matrices that were needed in the KF.

(3) The neural observation model was somewhat better in estimating flow rate and time mean speed than the analytical equations used in the OC model.

(4)The NKF model produced much better estimates for all the data sets than the OC model.

REFERENCES

1) C. Leondes: *Advances In Control Systems Theory and Applications*, Vol. 3, pp.219-292, Academic Press, New Youk, 1966.

2) D. C. Gazis, M. W. Szeto: *Design of Density-Measuring System for Roadways*, Highway Research Board Record, No. 388, pp.44-52, 1972.

3) H. J. Payne: *Model of Freeway Traffic and Control*, Simulation Councils Proceeding Series, Vol. I, No. I, Mathematical Model of Public System, pp. 51-61, 1971.

4) M. Cremer: *Der Verkehrsfluβ auf Schnellstraβen*, Springer-Verlag, New York, 1979.

5) J. Dayhoff: *Neural Network Architecture, An Introduction*, Van Nostrand Reinhold, New Yourk, 1990.

6) A. D. May: *Traffic Flow Fundamentals*, Prentice Hall, New Jersey, pp.283-320, 1990.

7) N. Pourmoallem, T, Nakatsuji: *A Multiple Section Method for Estimating Real-Time Traffic States on Freeway*, Infrastructure Planning Review, 18(2), pp.377-380, 1995.