IV-238

A NEURAL-KALMAN FILTERING METHOD FOR ESTIMATING TRAFFIC STATES ON FREEWAYS

N. POURMOALLEM, Student Member, HOKKAIDO University
T. NAKATSUJI, Regular Member, HOKKAIDO University

1. INTRODUCTION

A multilayer neural network model was integrated into a Kalman filtering method for estimating traffic states on a freeway. The Cremer model, which is a macroscopic traffic flow model combined with a Kalman filter, is revised: The observation equations that relate the state variables, to the observation variables, were described using a neural network model. The flow rate and time mean speed were estimated more precisely than those produced by an analytical model. Moreover, the state equations that define traffic flow dynamics were also described by another multilayer neural model. By using the neural networks, the derivatives of both state and observation equations that play an important role in correcting the estimates of state variables were easily obtained. This made it possible to implicitly realize a model parameter that was dependent on traffic states. By integrating those multilayer neural network models into a Kalman filtering technique, a procedure for estimating traffic states was proposed. This neural-kalman method was applied to a road section on the Metropolitan Expressway in Tokyo and it was examined how precisely the method could work as compared with the original Cremer model.

2. THEORETICAL BACKGROUNDS

(1) Macroscopic Traffic Flow Model[1)]

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 (OC) model, the dynamic equations are defined as follows:

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

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

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

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

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 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[3)]:

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

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

(2) Kalman Filter Model [2)]

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)[2)]:

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

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

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) Multilayer Neural Network Model[4)]

We used a neural network (NN) model, which consist of three layers; an input layer *(B)*, a hidden layer *(C)* , an output layer *(D)*. x_i^B represents an input signal and y_k^D an output signal. W_{ij}^{BC} and W_{jk}^{CD} are called synaptic weights. We repeated the back-propagation operations[4)] until average squared sum of the between y_k^D and target single z_k became sufficiently small. 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. It follows:

$$\frac{\partial y_k^D}{\partial x_i^B} = y_k^D\left(1 - y_k^D\right)\Sigma_j W_{jk}^{CD} \cdot y_j^C \cdot \left(1 - y_j^C\right) W_{ij}^{BC} \tag{11}$$

This derivative constitutes components of matrices Φ_k and Ψ_k. For describing the state equations of Eqs. (1) and (2) using a NN model, the traffic variables, such as $v_{i-1}(k-1)$, $c_i(k-1)$, $v_i(k-1)$, $c_{i+1}(k-1)$, $r_i(k)$, $s_i(k)$, and $V(c_i(k))$, on the right sides of the equations, were used as input signals, while the variables, such as $c_i(k)$, and $v_i(k)$, on the left sides as output signal. For the observation equations, the traffic variables, such as $c_i(k)$, $v_i(k)$, $c_{i+1}(k)$, and $v_{i+1}(k)$, on the right sides of the equations, were used as input signals, while the variables, such as $q_i(k)$, and $w_i(k)$, on the left sides as output signal.

3. NEURAL-KALMAN FILTER (NKF)

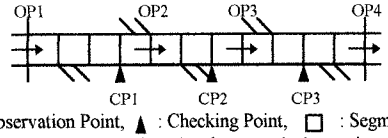
In conventional KF, both state and the observation equations have to be analytical functions. We proposed an alternative algorithm, in which the equations were described by the NN models. First, based on the estimates $\hat{x}(k-1)$ at the previous time k-1, we predict the state variable $\tilde{x}(k)$ at time k using Eqs. (1) and (2). In this process, to estimate the flow rate and time mean speed at the points where traffic data were not observed, we used the NN model of observation variables but

not Eqs. (3) and (4). Prior to obtaining the actual observed data *y(k)*, we estimate $\tilde{y}(k)$ using the NN of observation variables. At the same time, using the neural derivative of Eq. (11), we calculate the matrices Φ_k and Ψ_k which are required for determining the Kalman gain K_k. Then we correct the predicted $\tilde{x}(k)$ and obtain the new estimates $\hat{x}(k)$. Before we go to the next time step, we reidentify the NN to reflect the effects of the latest observed data into both systems. That is, adding the estimates $\tilde{x}(k)$ and the observed data *y(k)* to the existing training patterns, we adjust the synaptic weights again.

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. 1.



→ : Observation Point, ▲ : Checking Point, □ : Segment

Fig. 1 Overview of road section for numerical experiments.

(2) Initial Training

(a) Observation Equations

We trained the NN model for the observation equations. We supposed two types of neural structures, as shown in Fig. 2. Fig. 3 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.

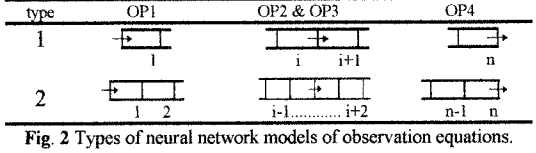


Fig. 2 Types of neural network models of observation equations.

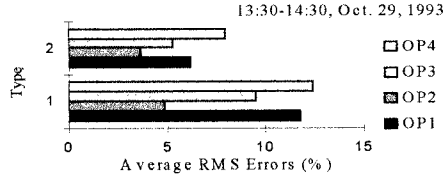


Fig. 3 RMS errors of neural observation systems for checking data.

(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. 4. 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. 5 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.

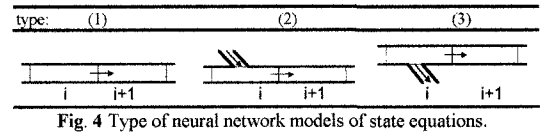


Fig. 4 Type of neural network models of state equations.

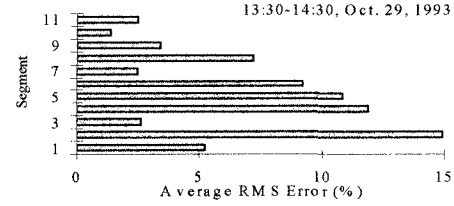


Fig. 5 RMS errors of neural state equations for checking data.

(3) Estimation of Traffic States

Fig. 6 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.

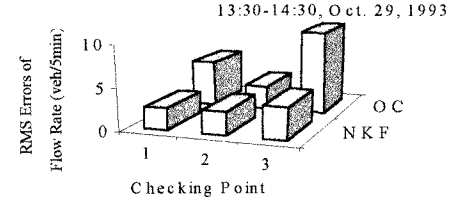


Fig. 6 Comparison of average RMS errors of flow rate and time mean speed evaluated by the OC and the NKF models.

5. CONCLUSIONS

The major finding 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) M. Cremer: Der Verkehrsfluβ auf Schnellstraβen, Springer-Verlag Berlin Heidelberg New York, 1979.
2) C.L.eondes: Advances in Control Systems, Volume 3, pp.219-292, 1966.
3) A D. May: Traffic Flow Fundamentals, Prentice Hall, New Jersey, pp.283-320, 1990.
4) J. Dayhoff: Neural Network Architecture, Van Nostrand Reinhold, 1990.