J. Infrastructure Plan. and Man., JSCE, No. 569/IV-36, 105-114, 1997 July

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

Nasser POURMOALLEM[1], Takashi NAKATSUJI[2] and Akira KAWAMURA[3]

[1]Member of JSCE, Doctoral Candidate of Eng., Dept. of Civil Eng., Faculty of Eng., Hokkaido University
(Kita-13, Hishi-8, Kita-ku, Sapporo, 060, JAPAN)
[2]Member of JSCE, Dr. of Eng., Associate Professor, Dept. of Civil Eng., Faculty of Eng., Hokkaido University
[3]Member of JSCE, Dr. of Eng., Associate Professor, Dept. of Civil Eng., Hakodate National College of Technology
(14-1 Tokura-cho, Hakodate 040, JAPAN)

A neural network model was integrated into the Cremer model for estimating traffic states on a freeway. In the Cremer model, which is a macroscopic traffic flow model combined with the Kalman filter, the observation equations that relate the state variables to the observation variables were described using a neural model. The state equations that define traffic flow dynamics were also defined by another neural model. By using the neural models, non-linear characteristics of traffic flows were represented. And the derivative matrices in the filter were easily obtained. Moreover, a parameter that was dependent on traffic states were implicitly realized. The method was applied to a road section in the Metropolitan Expressway.

Key Words: *macroscopic traffic flow model, cremer model, Kalman filter, neural network model*

1. INTRODUCTION

Traffic flow simulation is extensively used by planners and engineers to evaluate the effects of alternatives for existing and future traffic operations. The analysis involves the use of traffic flow models. Of those models, the macroscopic flow model, which was first proposed by Payne[1] and later modified by Cremer[2], is effective in describing traffic flow phenomena more precisely because the model is based on not only the conservation law of traffic flow but also the equation of motion. Even now the model has been modified and extended by many researchers, such as Ross et at.[3], Papageorgiou[4], Michalopoulos et at.[5] and Sanwal et at.[6]

Cremer[2] proposed a feed-back method using a Kalman filtering technique for estimating traffic states on a freeway: Traffic density and space mean speed that were estimated by the Payne model were adjusted so that the flow rate and time mean speed at observation points would coincide with actually observed ones. The method has a potential for future traffic control systems because it can estimate traffic states in real time without any driver's behavior model. To further improve the estimation precision, Cremer *et at.*[7] treated the model parameters as the state variables and then estimated them together with the other state variables.

We[8),9)] also improved the Cremer model. Considering the fact that the traffic states at a point are strongly affected by the traffic 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, we introduced a parameter that was dependent on traffic states. We called the model the Variant Weighting Factor (VWF) model. Moreover, to apply the Cremer model to a road section where traffic detectors were installed at any point, we extended the VWF model so that it could include any number of observation points in a road section. The model called the Multiple Section (MS) model made it possible to treat the road section continuously without subdividing it into several subsections.

However, the improvements on the Cremer model produced other difficulties in the estimation procedure: The introduction of a model parameter that was responsive to traffic states made the model equations very complicated and the differential operations required in the Kalman filter inevitably became very burdensome. The observation points that are located at any point in a road section also brought the same difficulty. In addition, there still remains another shortcoming in the original Cremer (OC) model: It employs only the traffic states in the nearest upstream and downstream segments in both the state

and observation equations. This means that the estimation precision depends on how long the segments are. To reflect the traffic states in the segments which are located in a certain length, we need to reorganize the equations by incorporating the traffic states in any number of the segments.

Neural network models have some promising abilities: They can accurately describe non-linear phenomena; they can organize their structures flexibly according to the observed data. 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 model. That is, we redefined the model 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 easy to derive the differential matrices in the Kalman filter. This eventually opened a way to employ any model parameters that were dependent on traffic states. We referred this approach to the neural-Kalman filtering method. Recently, Vytholkas[10] proposed a method that combined the Kalman filter with a neural network model for forecasting traffic states in urban networks. But he used the Kalman filter to identify the synaptic weights of the neural network, not to adjust the traffic states. In other words, he replaced the back-propagation method with a filtering technique. That is, the method is one of the Kalman-neural network models[11].

This paper aims to investigate the ability of the neural-Kalman filtering method to estimate traffic states on a freeway. 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. First, we briefly present the fundamental theoretical backgrounds of the Cremer model, including the improvements we made in the previous papers[8),9)]. Then, we explain how we can deal with the above problems using a neural network model. Next, we investigated how effective the neural-Kalman filtering method was. We applied the method to a road section on the Metropolitan Expressway in Tokyo and compared the results with those estimated by the Cremer model. Finally, we will summarize our research thus far and give our outlook on future work.

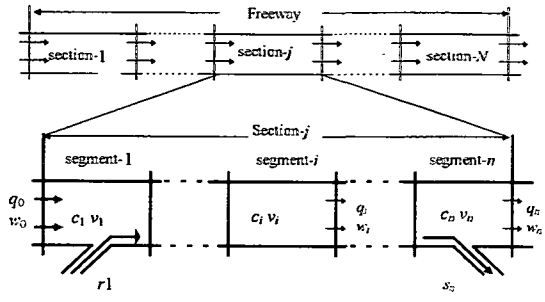


Fig. 1 Traffic flow variables on a freeway section.

2. THEORETICAL BACKGROUNDS

(1) Macroscopic Traffic Flow Model

Since we already defined the macroscopic model in previous papers[8),9)], we present only the notation of traffic variables and the fundamental equations used in this analysis:

a)Traffic variables (Fig.1)

$c_i(k)$: density of segment i at time k
$v_i(k)$: space mean speed of segment i at time k
$q_i(k)$: flow rate at boundary point of adjacent segments i and $i+1$
$w_i(k)$: flow rate at boundary point
$r_i(k)$: flow rate at entrance ramp i at time k
$s_i(k)$: flow rate at exit ramp i at time k

b) State equations

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

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

where

$$V(c_i(k)) = v_f\left[1 - \left(\frac{c_i(k)}{c_{max}}\right)^{(l-1)}\right]^{\left(\frac{1}{1-m}\right)} \tag{3}$$

ΔL_i : segment length
Δt : simulation time interval
τ : time constant
ν : sensitivity factor
κ : density constant.
v_f : free speed
c_{max} : the jam density
m and l : sensitivity factors

c) Observation equations

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

$$w_i(k) = \left[\alpha v_i + (1-\alpha) v_{i+1}\right]_{(k)} \quad (5)$$

where α is the weighting factor ranging 0.0 to 1.0. A detailed discussion of the Kalman filter can be found by investigating the references[1),2)].

(2) Kalman Filter[12),13)]

Choosing density and space mean speed as the state variable vector x_k and flow rate and time mean speed as the observation variable vector y_k, we can rewrite Eqs.(1) and (2) as a state equation and Eqs.(4) and (5) as an observation equation:

$$x_{k+1} = f(x_k) + \mathbf{B}u_k + \xi_k \quad (6)$$

$$y_k = g(x_k) + \zeta_k \quad (7)$$

where

$$x_k = \left[c_1, v_1, \ldots\ldots c_i, v_i \ldots\ldots c_n, v_n\right]^T_{(k)} \quad (8)$$

$$y_k = \left[q_0, w_0, \ldots\ldots, q_p, w_p\right]^T_{(k)} \quad (9)$$

where

u_k : inflow volume from inflow links
$\mathbf{B}$: the coefficient matrix
ξ_k and ζ_k : noise vectors
n : number of segments
p : number of observation points

Since Eqs. (6) and (7) are non-linear, we linearize them around the nominal solution x_k^* :

$$x_{k+1} = \Phi_k x_k + b_k + \mathbf{B}u_k + \xi_k \quad (10)$$

$$y_k = \Psi_k x_k + d_k + \zeta_k \quad (11)$$

where

$$b_k = f(x_k^*) - \Phi_k x_k^*$$
$$d_k = g(x_k^*) - \Psi_k x_k^* \quad (12)$$

$$\Phi_k = \frac{\partial f}{\partial x} \qquad \Psi_k = \frac{\partial g}{\partial x} \quad (13)$$

Calculating Φ_k and Ψ_k step by step, we can correct the state variables as follows:

$$\hat{x}_k = \tilde{x}_k + K_k\left[y_k - \tilde{y}_k\right] \quad (14)$$

where

$$\tilde{x}_k = f(\hat{x}_{k-1}) + \mathbf{B}u_k \quad (15)$$

$$\tilde{y}_k = g(\hat{x}_{k-1}) \quad (16)$$

$$K_k = M_k \Psi_k' (\Psi_k M_k \Psi_k' + \Gamma)^{-1}$$
$$P_k = M_k - K_k \Psi_k M_k \quad (17)$$
$$M_{k+1} = \Phi_k P_k \Phi_k' + \Omega$$

where the vectors $\tilde{x}_k$ and $\tilde{y}_k$ are referred to as the one-step predictor of x_k and $\hat{x}_k$ as the filtered estimate of x_k. K_k is the Kalman gain matrix at time k. And Γ and Ω are covariance matrix of ξ_k and ζ_k.

(3) Variant-Weighting Factor Model[8),9)]

The original Cremer (OC) model assumes a constant weighting factor α in Eqs.(4) and (5). 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:

$$\alpha\left(c_i(k)\right) = e^{-\beta c_i(k)} \quad (18)$$

where β is a curvature in the range of 0.0 to 1.0. We called this model a Variant-Weighting Factor (VWF) model. Since this function decreases monotonically 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 in Eqs.(12) and (13). This is the reason why we introduced a neural network model in this paper.

(4) Multiple Section Model[8),9)]

As shown in Eq. (2), to estimate the space mean speed $v_1(k)$ of the first segment, the space mean speed $v_0(k)$ preceding the entrance is needed. Similarly, to estimate the space mean speed $v_n(k)$ of the last segment, the density $c_{n+1}(k)$ beyond the exit is also needed. In the OC model, those variables are estimated in an extrapolating manner. Since the OC model is applicable to a single road section where traffic data are observed only at both/either entrance and/or exit of the section, we have to divide it into several subsections at every observation point when we apply the model to a long road section, in which several observation points are located inside. This subdivision

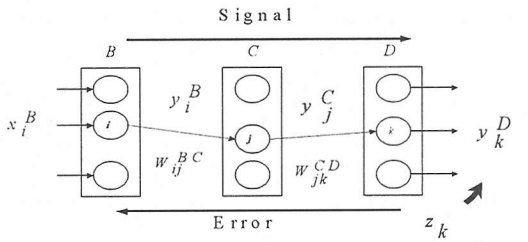


Fig.2 Multiple neural network model.

not only interrupts the propagation of traffic flow over the whole section, but also increases the frequency of extrapolations. Consequently, the estimation precision would be inevitably decreased.

Treating 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 (MS) model.

3. NEURAL-KALMAN FILTER

(1) Neural Network Model[14),15)]

We used a multilayer neural network model as shown in **Fig.2**, which consists of three layers; an input layer, a hidden layer, and an output layer. x_i^B represents an input signal and y_i^B an output signal. The output signal y_k^D is calculated as follows:

$$y_k^D = h\left(\sum_j W_{jk}^{CD} h\left(\sum_i W_{ij}^{BC} h\left(x_i^B\right)\right)\right) \qquad (19)$$

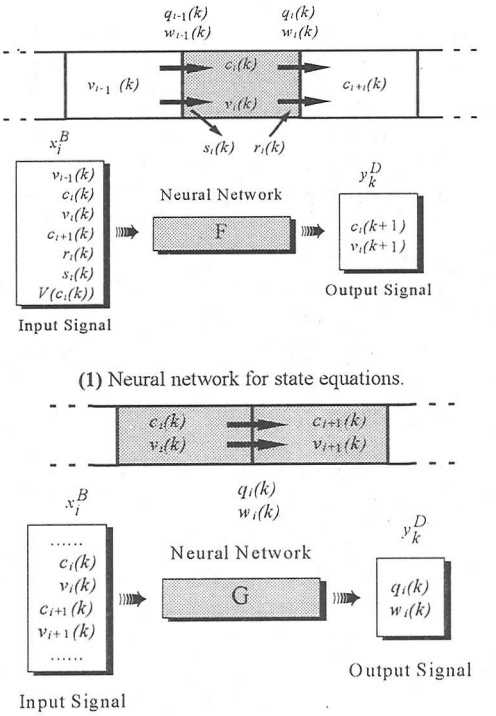
where $h(x)$ is the sigmoid function. For adjusting synaptic weights, we need target signals that are given by theoretical equations or observed data. We repeated the back-propagation operations[14)] until the following average squared sum of the between output signal y_k^D and target signal z_k became sufficiently small:

$$J = \frac{1}{N_D}\sum_k^{N_D}\left(y_k^D - z_k\right)^2 \qquad (20)$$

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 signal y_k^D to an input signal x_i^B because Eq. (19) is definitely defined by analytical functions. It follows:

$$\frac{\partial y_k^D}{\partial x_i^B} = y_k^D\left(1-y_k^D\right)\sum_j W_{jk}^{CD} y_j^C\left(1-y_j^C\right)W_{ij}^{BC} \qquad (21)$$



(1) Neural network for state equations.

(2) Neural network for observation equations.

Fig.3 Neural network modeling of state and observation equations.

As will be stated later, this derivative function constitutes the components of matrices Φ_k and Ψ_k in Eqs.(12) and (13).

First, we developed a neural network model for the state equation of Eqs.(1) and (2), as shown in **Fig.3(1)**. 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.(3) 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 in Eq.(12) 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.3(2)**. Although we emulated the basic structure of Eqs.(4) and (5), 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 in Eq.(13).

We produced both the neural networks 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.

(2) Neural-Kalman Filter

In conventional Kalman filters, both the state and the observation equations have to be analytical functions. However, traffic phenomena are often too complicated to describe analytically. The relationship between the state and observation variables is such a case: Although Cremer defined the relationship using Eqs. (4) and (5), traffic flow volume and average speed actually observed at a point are affected not only by the traffic states in the neighborhood, but also by various road and traffic conditions, such as curvature, grade, percentage of heavy vehicles and allowable speed. Analytical equations are almost impossible to reflect the effects of those conditions. 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. This eventually realized a model parameter, such as the weighting factor in Eq. (18), that was dependent on traffic states, because the synaptic weights were already adjusted so as to reflect the effects of the variant parameter. We referred this new approach to the neural-Kalman filtering method.

Fig.4 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. (4) and (5). 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.(21), 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.(14). Since the neural network model adopted here is a sort of nonlinear regression equation, it is important to choose reasonable input signals that well represent the

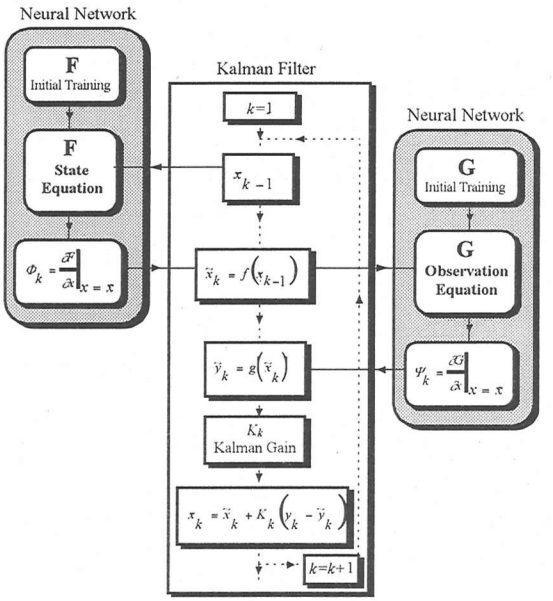


Fig.4 Block diagram of neural-Kalman filtering method.

output signals for establishing effective neural state and observation equations.

4. OBSERVATION DATA

The observed data used here came from a road section on the Yokohane Line of the Metropolitan Expressway. It is 5130 meters long. We used the traffic data from Oct. 28 to Nov. 1 in 1993. Traffic detectors were installed on both the nearside and the passing lanes every several hundred meters and traffic data of flow, occupancy, and average speed were compiled every 1 minute. Since the macroscopic model here does not represent the behaviors of individual vehicles on each lane, we aggregated the traffic data on both lanes into the ones on an equivalent single lane. This produced a traffic volume of greater than 20 or 30 vehicles per minute, which was more than enough to estimate traffic states every minute. As shown in **Fig.5**, we divided the road section into 11 segments whose lengths Δl_i ranged from 400 to 600 meters.

Then we assumed that traffic data were observed only at four points of OP1 (entrance), OP2, OP3, and OP4 (exit), although the data were actually observed at all the boundary points. For the original Cremer model, which is applicable only to a road section, we defined three subsections, SS1, SS2, SS3 in **Fig.5,** which were divided at every observation point. Each subsection contained 3 or 4 segments. **Table 1** shows the configuration of each subsection. On the other

Table 1 Configuration of subsections.

Subsection	length (km)	segments	on-ramp	off-ramp
1	1.92	4	1	1
2	1.23	3	-	1
3	1.98	4	1	1

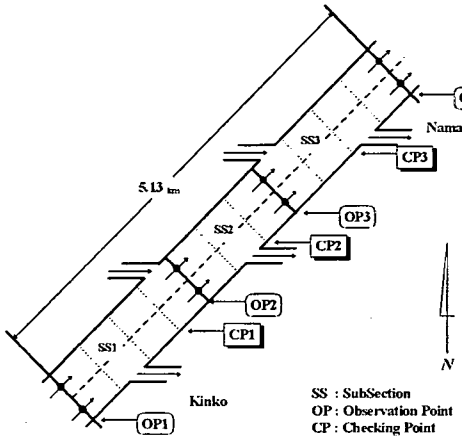


Fig.5 Overview of road section for numerical experiments.

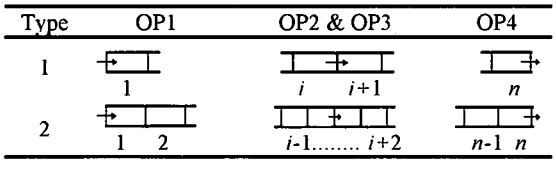


Fig.6 Types of neural network models of observation equations.

hand, for the neural-Kalman method, we used the multiple section (MS) model while treating the whole road section as a single section. That is, there were 11 segments with four observation points. To examine the effectiveness of each model, we set three checking points of CP1, CP2, and CP3.

Prior to the numerical analyses, the simulation time interval Δt must be carefully examined. It should be noted here that the Payne-type macroscopic model is designed to describe the dynamic behavior every few seconds to a maximum 20 or 30 seconds by defining the equation motion of Eq.(2). The model can accurately represent the transition phenomena such as the propagation of a congestion wave. Cremer[2] discussed the relationship between the time interval Δt and the segment length Δl. Although they primarily depend on what the model is used for, Cremer indicated that the ratio of Δl to Δt should be in a certain range. He recommended 10 seconds of Δt and 500m of Δl as the standard for freeways. According to his recommendation, we divided the road section into segments of 400 to 600 meters and simulated the traffic flows every 10 seconds. Since the traffic data are measured every one minute, we assumed that the traffic flows are stationary during the time.

5. NUMERICAL EXPERIMENTS

(1) Model Parameters

The macroscopic model includes several model parameters, such as is v_f, c_{max}, m, l, α, τ, υ, and κ. We estimated them based on a traffic simulation model as Cremer[7] did. Picking out a time period of two hours each day, we estimated all the model parameters at each subsection for each time period. That is, we optimized the parameters so as to minimize the following objective function J_j at each checking point:

$$J_j = \frac{1}{h}\sum_{k}^{k}\left[\left(\left(\hat{q}_j - q_j\right)/\sigma_{q_j}\right)_k^2 + \left(\left(\hat{w}_j - w_j\right)/\sigma_{w_j}\right)_k^2\right] \quad (22)$$

where $q_j(k)$ and $\hat{q}_j(k)$ are observed and estimated flow rates at checking point j at time k. $w_j(k)$ and $\hat{w}_j(k)$ are the time mean speed. σ_{qj} and σ_{wj} are their standard deviation. h is the total number of time steps.

(2) Initial Training

a) Observation Equations

First, we trained the neural network (NN) model for the observation equations. That is, we built a relationship between the state and observation variables at each observation point in **Fig.5**. Considering the fact that the observation variables at a given point were related to the state variables in both the upstream and downstream segments, we assumed two types of neural structures, as shown in **Fig.6**. That is, we assumed the effects of traffic states not only in the nearest segment but also in the farther segments. For example, the type 2 model at observation point 2 inputs the density and the space mean speed of two upstream and two downstream segments and then outputs the flow rate and the time mean speed.

Table 2 shows the number of neurons in each layer for each type of model. The number of neurons in the intermediate layer was determined experimentally. It should be noted that what is important here is not the number itself, but the balance between the total number of synaptic weights and the number of training data sets. In other words, by preparing a sufficient, but not too great a number of training data sets in comparison with the total number of synaptic weights, we can normally realize a stable relationship between the input and output signals. As stated below, we prepared several times the number of training data sets than the number of synaptic weights.

Table 2 Number of neurons for observation equations.

Type	number of neurons	OP1	OP2	OP3	OP4
1	input layer	2	4	4	2
	intermediate layer	2	3	3	2
	output layer	2	2	2	2
2	input layer	4	8	8	4
	intermediate layer	3	5	5	3
	output layer	2	2	2	2
number of training data		240 (13:30-14:30,Oct. 28,30, Nov. 1)			
number of checking data		60 (13:30-14:30, Oct. 29)			

Picking out a time period of one hour a day from Oct. 28 to Nov. 1, we produced 240 sets of training data and 60 sets of checking data because the observed data were compiled every one minute. Although it may seem to be a bit curious to select the traffic data prior to the training as the checking data, we employed the data on Oct. 29 as the checking data because the time period contained extensive traffic data.

The training procedure was simple. First, we assumed the initial synaptic weights. Then we gave the input signals into the input layer and calculated the output signals, which corresponded to the observation variables. Then, using the back-propagation method, we adjusted the synaptic weights so as to minimize the difference between the actually observed and estimated variables. In the actual computation, the input and output signals were normalized by appropriate numbers. We iterated this procedure until the Root Mean Squared (RMS) error became sufficiently small for all of the training patterns.

The capability of a NN model can be evaluated by the estimation precision for checking data that are not used for the training. After the completion of the training, we gave the input signals of the checking data and calculated the output signals using Eq.(19). Then we compared them with those that were actually observed. **Fig.7** depicts the average RMS errors of output signals for 60 checking patterns at four observation points for each type of NN model in **Fig.6**. 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 the upstream and the downstream into the model, we could estimate the observation variables more precisely. As shown in **Fig.7**, the errors of the type 2 model were nearly 5%. Judging from our experiences, RMS errors less than 5% for checking patterns could produce satisfactory results in the later filtering process.

Also, we trained other types of the NN model, which input the traffic states in more segments in both upstream and downstream. However, the RMS errors were not less than those for the type 2 model. This is

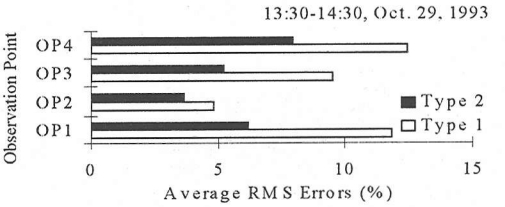


Fig.7 Average RMS errors of neural observation systems for checking data (13:30-14:30, Oct. 29,1993).

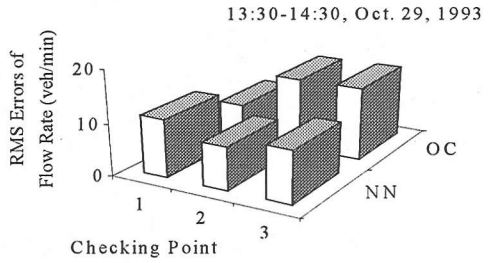
probably because the number of training patterns were insufficient compared with that of the input signals.

To evaluate the estimation precision further, we investigated how precise the NN model could estimate flow rate and time mean speed at checking points whose data were not used for the training and compared them with those by the original Cremer (OC) model. **Fig.8** shows the average RMS errors of **(1)** flow rate and **(2)** time mean speed at three checking points in **Fig.5**. We can see that the errors of the NN model were somewhat smaller than those by the OC model except for the flow rate at CP1 and the time mean speed at CP2. In total, the NN model gave better results for 4 of 6 data sets. However, it is a bit too early to judge the effectiveness of the neural network model at this stage because the estimates are not corrected yet by the Kalman filter.

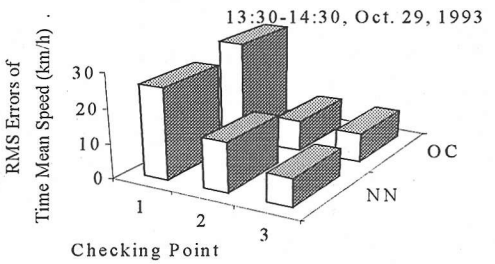
b) State Equations

Next, we trained the neural network (NN) model for the state equations of Eqs.(1) and (2) to produce the dynamic derivative of the matrix Φ_k. That is, we built a relationship between the state variables at time k and those at time $k+1$ for each segment. As mentioned before, the neural model was designed to emulate the state equations themselves: We input $v_{i-1}(k)$, $c_i(k)$, $v_i(k)$, $c_{i+1}(k)$, $q_{i-1}(k)$, $q_i(k)$, $r_i(k)$, $s_i(k)$, and $V(c_i(k))$ and output $c_i(k+1)$ and $v_i(k+1)$. According to whether segments have an on-ramp or an off-ramp, we classified the segments into three types, as shown in **Fig.9**. 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 an on- or off-ramp. We allocated five neurons to the intermediate layer. As well as in the neural networks for the observation equations, the number of neurons in the intermediate layer did not affect the estimation precision because we provided a sufficient number of training patterns.

Picking out a time period of two hours a day, 13:00 to 15:00, from Oct. 28 to Nov. 1, we produced 480 sets of training data and 120 sets of checking data: For each time period, we simulated the traffic flow using Eqs.(1) and (2) by inputting the actually observed flow rates. Although we obtained 3600 sets of data in



(1) Flow rate.



(2) Time mean speed.

Fig.8 Average RMS errors of neural observation systems at checking points, compared with those of original Cremer model(13:30-14:30, Oct. 29,1993).

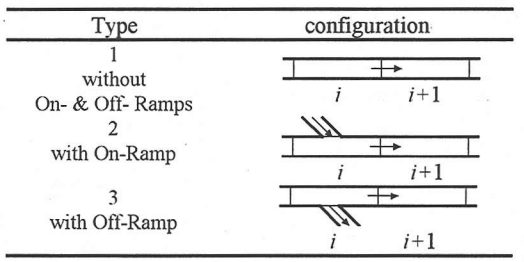


Fig.9 Type of neural network models of state equations.

total because we simulated every 10 seconds for 10 hours, we thinned out the results every 1 minute. Consequently we got 600 sets of data, 480 as the training data and 120 as the checking data. For the same reason as in the previous section, we used the data on Oct. 29 as the checking data. This training procedure is the same as in the observation equation.

After the training was completed, we examined how precisely the NN models could estimate the state variables. Giving the input signals of the checking data, we calculated density and space mean speed at each segment using Eq.(19) and compared them with those that were estimated by Eqs.(1) and (2). **Fig.10** depicts the average RMS errors of output signals for120 sets of checking data at all the segments. We can see that the errors are small enough for some segments because the errors do not exceed 5%. But the errors may be somewhat large in the other segments. It

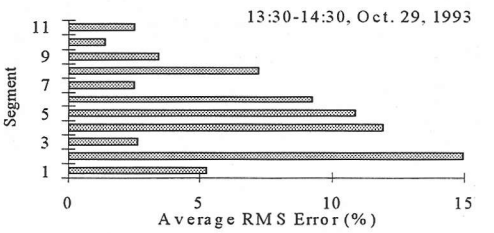


Fig.10 Average RMS errors of neural state equations for checking data (13:30-14:30, Oct. 29, 1993).

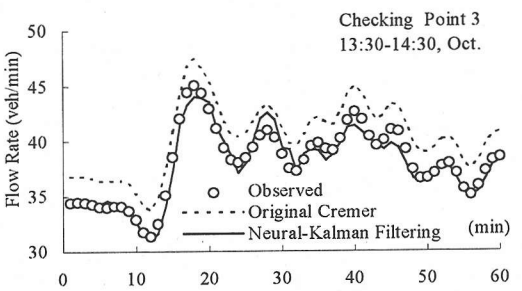
should be noted that the estimates are not corrected yet by the actually observed data. Moreover, as will be shown later, since we used the neural state equations only for producing the matrix Φ_k but not for estimating the state variables, the errors did not strongly affect the estimation precision.

(3) Estimates of Traffic States

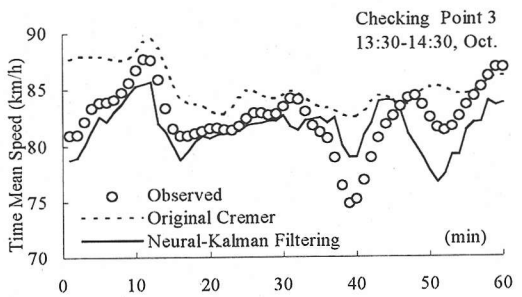
To examine how effective the neural-Kalman filtering (NKF) model was for estimating traffic states, we applied it to a road section shown in **Fig.5** and compared the flow rate and the time mean speed at the checking points and compared them with those estimated by the original Cremer (OC) model. As the checking data, we took the time period from 13:30 to 14:30 on Oct. 29, which included traffic data in both light and heavy traffic states.

Prior to the estimation by the NKF model, we have to clarify how the observation errors induced into the traffic detector data affect the estimation precision. Using the detector data observed on some urban roads, Nakatsuji[16] investigated how large the noises in traffic volume were and how the Kalman gain decreased as the noises increased. The error of traffic volume on the arterial roads was small enough (less than 5% in most cases) and the Kalman gain was little affected. Unfortunately, since we do not have any true observed data at this road section, we could not directly evaluate the effect of the noises. However, considering the fact that the noises were less on arterial roads than on streets in commercial districts, we assumed that the traffic data observed on the Metropolitan Expressway are not more contaminated by noises than the data on the arterial roads. That is, we assumed the error to be 5% for both traffic volume and time mean speed. In addition, what is important here is to recognize that the decrease of the Kalman gain does not mean the direct deterioration of the estimation precision. As shown in Eq.(14), it means that the observed data contribute less in the correction process. Anyway, we have to carefully examine what we estimated because the effect of noises have not been accurately accessed yet.

According to the procedure in **Fig.4**, we first

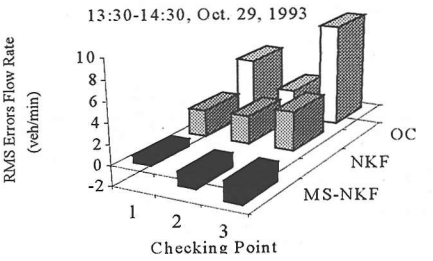


(1) Flow rate.

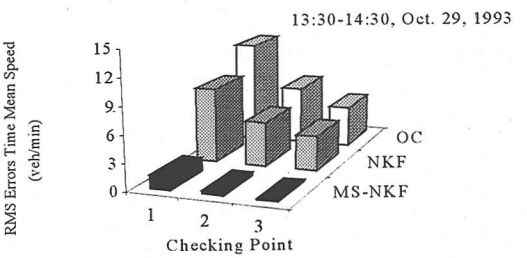


(2) Time Mean Speed

Fig.11 Estimation of flow rate and time mean speed estimated by neural-Kalman filtering model, compared with those by original Cremer model and observed ones. (CP3, 13:30-14:30, Oct. 29,1993)



(1) Flow rate.



(2) Time Mean Speed

Fig.12 Comparison of average RMS errors of flow rate and time mean speed evaluated by original Cremer (OC) model, neural-Kalman filtering (NKF) model and the difference of RMS errors between the NKF and MS models. (13:30-14:30, Oct. 29,1993)

estimated the density and space mean speed of all the segments and then corrected them using the neural-Kalman filter. Since the real state variables were unknown, we evaluated the method by comparing the estimated flow rate and the time mean speed at three checking points with those of actually observed ones. **Fig.11(1)** shows the estimates of the flow rate at checking point 3. The x-axis represents time and the y-axis represents the flow rate. The estimates were compiled every 1 minute to compare with the observed ones. We can see that the OC model overestimated the flow rate during the time period. On the other hand, the NKF model got very much closer to the observed ones than the OC model. Similarly, **Fig.11(2)** shows the estimates of time mean speed. The OC model was not successful in describing the variation at the early stage and detecting the speed drop at around 40 minutes. On the other hand, the NKF model agrees well with the observed data, but the model is not successful in describing the variation at the end of the time period. We further investigated the effectiveness of the NKF model. **Fig.12** shows the comparison of the average RMS errors of **(1)** flow rate and **(2)** time mean speed at three checking points at the same time period for the OC and NKF models. First, we can see that both the OC and the NKF models greatly improved the estimates by the filtering operations, compared with the ones in **Fig.8**, which were not corrected yet. This means that the filtering operations were very effective in estimating the traffic states on the road section. Also, the NKF model produced much better estimates for all the data sets than the OC model. And the RMS errors are sufficiently small. As mentioned above, we used the multiple section (MS) model in the NFK model. Therefore, the improvements in **Fig.12** may be brought by the MS model rather than the NKF model. For reference, we showed the difference of RMS errors between the MS and NKF models, MS-NKF which are depicted by the black bars in **Fig.12**. In the MS model both the state and observation equations were analytically described as well as in the OC model. We can see that the NKF model gives better results for 1 of 3 data sets of flow rate and for 3 of 3 sets of time mean speed. Although the errors for the flow rate were not as good as expected, the values are small enough. The results for time mean speed are satisfactory.

In this way, the neural-Kalman filtering model was effective in estimating the traffic states on the freeway road section used in this analysis. However, we still have a lot of problems to be solved. The model is a bit time-consuming because the neural operations of Eq.(19) are carried out every time traffic variables are estimated. Application to a large road network that contains many observation points is another problem.

5. CONCLUSIONS

To estimate traffic states on a freeway more precisely, we integrated a multilayer neural network model into a Kalman filtering method. Intending to extend the Cremer model, we investigated how a multilayer neural network model could describe both state and observation equations. We applied the new method to a road section on the Metropolitan Expressway in Tokyo and compared the results with those produced by the original Cremer model. The major findings are summarized as follows:

(i) By integrating the neural network models into a Kalman filtering technique, we proposed a new procedure to estimate the traffic states on a freeway road section.

(ii) These neural models for both state and observation equations made it possible to easily produce the dynamic derivative matrices that were needed for the Kalman filter.

(iii) The neural observation model, which inputs density and space mean speed of two adjacent segments, was somewhat better in estimating flow rate and time mean speed than the analytical equations used in the original Cremer model.

(iv) The method produced much better estimates than the original Cremer model. Moreover, the neural-Kalman filtering method estimated them somewhat better than the multiple section model which we proposed earlier.

Traffic detector data actually observed at a point are influenced not only by the traffic states in the neighborhood, but also by various road and traffic conditions around the point. Employment of such conditions as the input signal of the neural network equations would serve not only to improve the estimation precision but also to promptly detect the occurrence of incidents. In addition, since neural network models have a promising ability of accurately learning what they have experienced, we could realize an adaptive traffic control system, which is not based on any driver behavior models, by thoroughly training the relationship between any countermeasures for traffic congestion and the resulting traffic states.

ACKNOWLEDGMENT: This research was supported by a grant from the Ministry of Education, Science, Sports and Culture, Government of Japan, Grant-in-aid for scientific research (06650579).

REFERENCES

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

2) Cremer, M. : Der Verkehrsfluβ auf Schnellstraβen, Springer Verlag, New York, 1979.

3) Ross, P. : Traffic Dynamics, Trsnspn. Res, B. Vol. 22, N0. 6, pp.421-435, 1988

4) Papageorgiou, M. : Macroscopic Modeling of Traffic Flow on the Boulevard Peripherique in Paris, Trsnspn. Res, B. Vol. 23, No. 1, pp.29-47, 1989

5) Michalopoulos, P. G., Ping, Y. and Lyrintzis, A. : Continuum Modeling of traffic dynamics for congested freeways, Trsnspn. Res, B. Vol. 27, No. 4, pp.315-332, 1993.

6) Sanwal, K. K., Petty, K., Warand, J. and Fawaz, Y. : An Extended Macroscopic Model for Traffic flow, Trsnspn. Res, B. Vol. 30, No. 1, pp.1-9, 1996.

7) Cremer, M. and Papageorgiou, M. : Parameter Identification for A Traffic Flow Model, Automatica, Vol. 17, No. 6, pp. 837-843, 1981.

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

9) Pourmoallem, N. and Nakatsuji, T. : Multiple Section Model for Estimating Traffic States on a Freeway, Proc. 15th Traffic Engineering Conference, pp.13-16, 1995.

10) Vythoulkas, P. C. : Alternative Approaches to Short Term Traffic Forecasting for Use in Driver Information Systems, Proc. of the 12th Int. Trans. Traffic Theory, pp.485-506, 1993.

11) Murase, H., Koyama, S. and Ishida, R. : Kalman-Neural Computing by Personal Computers, Morikita Shupan, Tokyo, 1994.

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

13) Sorenson, H. W. : Kalman Filtering: Theory and Application, IEEE Press, New York, 1985.

14) Dayhoff, J. : Neural Network Architecture, An Introduction, Van Nostrand Reinhold, New York, 1990.

15) Wasserman, P. D. : Neural Computing Theory and Practice, Van Nostrand Reinhold, New York, 1989.

16) Nakatsuji, T. and Kaku, T. : Improvement of Prediction Scheme of Traffic Flow on Urban Streets Based on Piecewise Stationary, Infrastructure Planning Review, No.4, pp.101-108, 1986.

(Received May 21,1996)

高速道路の交通状態を推定するためのニューラル・カルマンフィルター法の開発

ナセル　プールモアレム・中辻　隆・川村　彰

高速道路の交通状態を推定するためのCremerモデルをニューラルネットワークモデルを用いて再定義した．Cremerモデルは，マクロ交通流モデルとカルマンフィルタを組み合わせて，観測データによるフィードバックを行いながら交通状態をリアルタイムに推定する手法であるが，状態方程式と観測方程式ともに多層階層型のニューラルネットワークモデルによって表現した．このモデル化によって，観測方程式における非線形な交通流特性を表現することができるようになった．また，カルマンフィルタにおいて必要とされる導関数行列の導出が容易になり，交通状態に依存したモデルパラメータの定義が可能となった．新手法を首都高速道路の道路区間に適用し元のCremerモデルとの比較を行い有用性の検証を行った．