Journal of Southeast University (English Edition) Vol. 26, No. 3, pp. 466 – 470 Sept. 2010 ISSN 1003—7985

Proactive traffic responsive control based on state-space neural network and extended Kalman filter

Guo Xiucheng Li Yan Yang Jie

(School of Transportation, Southeast University, Nanjing 210096, China)

Abstract: The state-space neural network and extended Kalman filter model is used to directly predict the optimal timing plan that corresponds to futuristic traffic conditions in real time with the purposes of avoiding the lagging of the signal timing plans to traffic conditions. Utilizing the traffic conditions in current and former intervals, the network topology of the state-space neural network (SSNN), which is derived from the geometry of urban arterial routes, is used to predict the optimal timing plan corresponding to the traffic conditions in the next time interval. In order to improve the effectiveness of the SSNN, the extended Kalman filter (EKF) is proposed to train the SSNN instead of conventional approaches. Raw traffic data of the Guangzhou Road, Nanjing and the optimal signal timing plan generated by a multi-objective optimization genetic algorithm are applied to test the performance of the proposed model. The results indicate that compared with the SSNN and the BP neural network, the proposed model can closely match the optimal timing plans in futuristic states with higher efficiency.

Key words: state-space neural network; extended Kalman filter; traffic responsive control; timing plan; traffic state prediction

Control methodologies of traffic signals have gradually improved, along with the advancements in technology. As the traffic demand at most intersections has homologous changes over time of day, traffic engineers develop multiple signal timing plans to accommodate these changes. In the commonly used time-of-day (TOD) control, signal timing plans are predetermined based on numerous archived traffic data. This type of control is suggested to be used in places where traffic patterns are predictable, but not short-term fluctuations in traffic arrivals or long-term changes in traffic patterns[1]. On the other hand, adaptive control systems adjust signal timings in response to real-time traffic conditions, which help to relieve the traffic congestion. The Sydney coordinated adaptive traffic system (SCATS)[2] and the split cycle offset optimization technique (SCOOT)[3] are the most widely deployed systems. Adaptive systems, however, require major investments in terms of infrastructure and communication hardware.

Alternatively, existing controllers can be operated with the traffic responsive plan selection (TRPS) control mode which can provide an operation level close to that of adaptive control systems. Like other adaptive control modes, the traffic responsive control mode has the ability to switch timing plans in response to traffic patterns. However, the main drawback of the TRPS control is due to the fact that the timing plan given by the TRPS method lags behind the change in the real-time traffic pattern.

Various computational intelligent algorithms are developed to solve adaptive and responsive control problems. These methods include the fuzzy sets[4], the genetic algorithm[5], the reinforcement learning[6], and the artificial neural networks (ANN)[7]. Most of these algorithms are based on the distributed approach, where an agent is assigned to update the traffic signals of a single intersection based on the traffic flow in all the directions of that intersection. Some models only implemented and tested the controller on a simplified traffic network with one intersection[8], and the effectiveness of the proposed controller for controlling a large-scale traffic network cannot be established. Meanwhile, as ANN is a gradient descent search method, its error function falls into local extreme points easily. When it is used in larger space searches, multi-peak functions or non-differentiable functions, the ANN cannot effectively find the global minimum point.

A proactive model that utilizes a nonlinear state-space neural network (SSNN) and extended Kalman filter (EKF) is presented for the traffic responsive plan selection control mode. Instead of the common “black-box” approach, the SSNN topology is loosely based on the geometry of the arterial route of interest[9]. Accordingly, the SSNN shows that the internal states are closely related to actual traffic conditions on the sections along the route, and can directly predict the optimal timing plan that corresponds to the futuristic traffic conditions in real time. The Kalman filter technique is applied in the model to calibrate neural networks, which can greatly reduce the training time of the SSNN.

1 State-Space Formulation of Traffic Responsive Control Problem

The real-time traffic signal plan involves both spatial and temporal relationships between some detector observable quantities such as counts and occupancies. The proposed model is based on a discrete state-space model, in which the timing plan of a system in the time interval t is uniquely predicted and defined by its predecessor state in the previous time period $t-1$ and the inputs to the system at time interval t. In this paper, the system inputs refer to the counts and the occupancies obtained from the detectors.

As indicated in Fig. 1, the traffic state of the road network $\boldsymbol{s}_i(t)$ on time interval t is uniquely defined by its previous state and the inputs $\boldsymbol{u}_i(t)$ during the current time peri-

Received 2009-10-18.
Biography: Guo Xiucheng (1964—), male, doctor, professor, seuguo@163.com.
Foundation items: The National Natural Science Foundation of China (No. 50422283), the Soft Science Research Project of Ministry of Housing and Urban-Rural Development of China (No. 2008-K5-14).
Citation: Guo Xiucheng, Li Yan, Yang Jie. Proactive traffic responsive control based on state-space neural network and extended Kalman filter[J]. Journal of Southeast University (English Edition), 2010, 26(3): 466 – 470.

od. The inputs are the traffic states acquired via system detectors. In our application, only counts and occupancies from the system detectors are considered.

The generic function $\boldsymbol{y}(t+1) = g[\boldsymbol{w}, \boldsymbol{s}(t)]$ estimates the traffic signal plan in the future time interval accompanying the traffic state at time interval $t-1$ and takes a vector of all the system detector states at time interval t, $\boldsymbol{u}(t)$, as inputs. It also incorporates a parameter vector $\boldsymbol{w}$, which is the traffic signal plan at time interval t to adjust the model during calibration.

Interpretation of the state variable $\boldsymbol{s}_i(t)$ of detector i depends on the chosen function $f(\cdot)$ [10]. Here, $\boldsymbol{s}_i(t)$ is interpreted as a representative for the input traffic conditions on detector i at time interval t. Different classes of state-space models can be derived from the generic formulae:

1) Autoregressive (integrated) moving average (AR(I)MA) models: $\boldsymbol{s}^{\mathrm{T}}(t) = \boldsymbol{u}^{\mathrm{T}}(t)$, $g(\cdot)$ is a linear function and let $f(\cdot)$ be the identity function (i. e., f: $\boldsymbol{x} \rightarrow \boldsymbol{x}$). The state of a link in this case is simply a vector which is comprised of time mean speed and flow of the link.

2) Linear state-space models: $f(\cdot)$ and $g(\cdot)$ are linear functions.

3) Nonlinear state-space models: If $f(\cdot)$ is allowed to be nonlinear, $g(\cdot)$ may be either linear or nonlinear. Some examples of the nonlinear state-space models can be classified into a specific class of the recurrent neural network (RNN), which are further investigated in the next section.

Based on the highly nonlinear characteristics of the TRPS traffic signal control problem, it can be argued that the third-class nonlinear state-space models offer the best choice for the traffic signal plan selection.

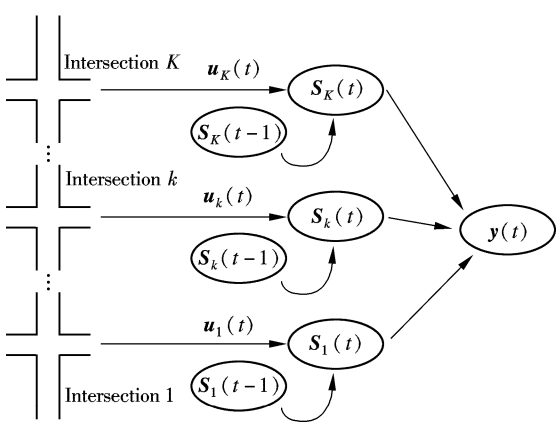

Fig. 1 State-space dynamics of a road network with K adjacent intersections

2 State-Space Neural Network and Extended Kalman Filter

2.1 State-space neural network

The SSNN is derived from the RNN proposed by Elman, who provided insight into how the RNN manages to represent spatiotemporal patterns in an efficient distributed manner through its weights[11]. The basic idea of the SSNN is to add a context layer as a short memory that stores previous internal states to learn complex spatiotemporal patterns. The concept of taking into account previous states resembles a Markov chain. Each time a step pattern is presented, the unit computes its activation similar to a feed-forward network. However, its net input contains a term that reflects the internal states of the network before the pattern can be seen. When subsequent patterns are presented, the states of the hidden units and the output units will be the updated functions of everything in the network.

Looking at the mathematical description of the SSNN, the hidden layer vector $\boldsymbol{s}(t)$ is calculated from the input vector $\boldsymbol{x}(t)$. A weighted sum of input and bias (here, fixed at 1) is calculated, and the results are transformed by the transfer function:

$$\begin{bmatrix} \boldsymbol{s}_1(t) \\ \boldsymbol{s}_2(t) \\ \vdots \\ \boldsymbol{s}_m(t) \end{bmatrix} = \begin{bmatrix} h\left[\sum_{i=1}^{n} w_{i,1}^{il} \boldsymbol{x}_i(t) + \sum_{e=1}^{m} w_{e,1}^{ll} \boldsymbol{s}_1(t-1) + v_1^{il} b_1 \right] \\ h\left[\sum_{i=1}^{n} w_{i,2}^{il} \boldsymbol{x}_i(t) + \sum_{e=1}^{m} w_{e,2}^{ll} \boldsymbol{s}_2(t-1) + v_2^{il} b_2 \right] \\ \vdots \\ h\left[\sum_{i=1}^{n} w_{i,m}^{il} \boldsymbol{x}_i(t) + \sum_{e=1}^{m} w_{e,m}^{ll} \boldsymbol{s}_m(t-1) + v_m^{il} b_m \right] \end{bmatrix} \tag{1}$$

where $\boldsymbol{s}_m$ is the value of the m-th hidden neuron; $w_{i,m}^{il}$ is the weight connecting with the i-th input neuron and the m-th hidden neuron; $w_{e,m}^{ll}$ is the weight connecting with the e-th hidden neuron and the m-th context neuron; v_m^{il} is the weight of the bias associated with the m-th hidden neuron; b_m is the bias with fixed value 1 for the m-th hidden neuron; and $h(\cdot)$ is the transfer function.

The well-known nonlinear sigmoid transfer function is used to take the values from the summation results and turn them into values between 0 and 1.

$$h(z) = \frac{1}{z + e^{-z}} \tag{2}$$

Similarly, the output layer vector $\boldsymbol{y}(t)$ is calculated as

$$\begin{bmatrix} \boldsymbol{y}_1(t) \\ \boldsymbol{y}_2(t) \\ \vdots \\ \boldsymbol{y}_l(t) \end{bmatrix} = \begin{bmatrix} h\left[\sum_{i=1}^{m} w_{i,1}^{l0} \boldsymbol{s}_i(t) + v_1^{l0} b_1 \right] \\ h\left[\sum_{i=1}^{m} w_{i,2}^{l0} \boldsymbol{s}_i(t) + v_2^{l0} b_2 \right] \\ \vdots \\ h\left[\sum_{i=1}^{m} w_{i,l}^{l0} \boldsymbol{s}_i(t) + v_l^{l0} b_l \right] \end{bmatrix} \tag{3}$$

where $w_{i,l}^{l0}$ is the weight connecting with the i-th hidden neuron and the l-th output neuron; v_l^{l0} is the weight of bias associated with the l-th output neuron; and b_l is the bias with a fixed value of 1 for the l-th hidden neuron.

2.2 Training SSNN with EKF

The behavior of the neural network can be formulated by a nonlinear discrete time system:

$$\boldsymbol{y}_k = g(\boldsymbol{\theta}_k) + \boldsymbol{v}_k \tag{4}$$

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \boldsymbol{\omega}_k \tag{5}$$

where $\boldsymbol{\theta}_k$ is the weight parameters of the neural network specified as a stationary process; $\boldsymbol{\omega}_k$ is the process noise; $\boldsymbol{y}_k$

is the observation vector; $\boldsymbol{v}_k$ is the observation noise; and $g(\cdot)$ is the nonlinear function of the state.

With the Taylor series, Eq. (4) can be expanded around the state estimate $\hat{\boldsymbol{\theta}}$ as

$$\boldsymbol{y}_k = g(\hat{\boldsymbol{\theta}}) + \frac{\partial g(\hat{\boldsymbol{\theta}})}{\partial \hat{\boldsymbol{\theta}}}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) + o(\boldsymbol{\theta}) \tag{6}$$

Ignoring the higher-order terms, the EKF solution to the training problem is given by

$$\left.\begin{aligned} \hat{\boldsymbol{\theta}}_k &= \hat{\boldsymbol{\theta}}_{k-1} + \boldsymbol{K}_k[\boldsymbol{y}_k - g(\hat{\boldsymbol{\theta}}_{k-1})] \\ \boldsymbol{K}_k &= \boldsymbol{P}_k\boldsymbol{H}_k(\boldsymbol{R}_k + \boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{P}_k\boldsymbol{H}_k)^{-1} \\ \boldsymbol{P}_{k+1} &= \boldsymbol{P}_k - \boldsymbol{K}_k\boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{P}_k \end{aligned}\right\} \tag{7}$$

At each time step, an output vector $\hat{\boldsymbol{y}}_k$ is yielded through the SSNN model with the input vector $\boldsymbol{x}_k$. The error vector $\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k$ is used to calculate the derivative matrix $\boldsymbol{H}_k$. The Kalman gain matrix $\boldsymbol{K}_k$ is computed as a function of the derivative matrix $\boldsymbol{H}_k$, the approximate error covariance matrix $\boldsymbol{P}_k$, and the measurement covariance noise matrix $\boldsymbol{R}_k$. The weight parameters are updated with the Kalman gain matrix, the error vector, and the current values of the weight. Finally, the Kalman gain matrix, the derivative matrix, and the current approximate error covariance matrix are used to update the approximate error covariance matrix. The topology of the SSNN and the process of training the SSNN with the EKF are shown in Fig. 2.

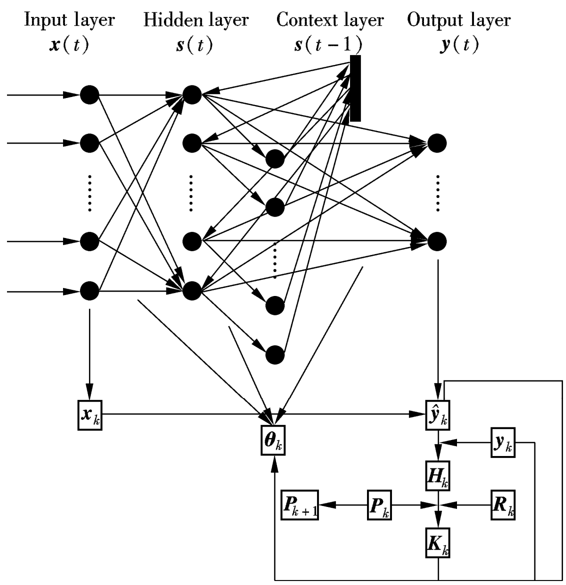

Fig. 2 Topology of SSNN and flow diagram for EFK training SSNN

3 Evaluation Setup

3.1 Data description

The traffic data from Nanjing is used to evaluate this method. The test area contains four adjacent intersections along Guangzhou Road. The traffic data used to train the neural network is collected at the site for 7 d using 17 system detectors. The seventh day's data (work day) is selected to test the model. Fig. 3 shows the intersection layout and the specified structure of the SSNN model.

The input data $\boldsymbol{x}(t)$ consists of traffic counts and occu-

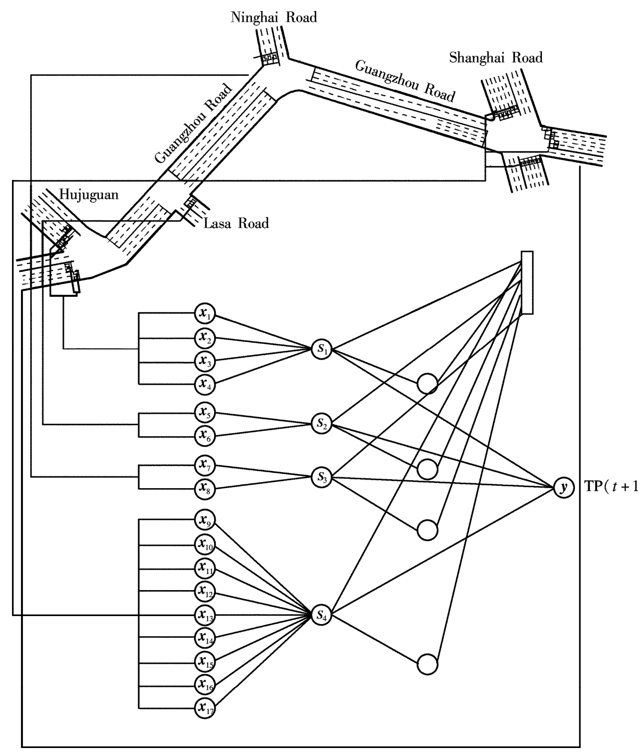

Fig. 3 Layout of intersections and structure of SSNN model

pancies collected by system detectors in terms of an individual stream at each intersection every 15 min. The neurons represent the corresponding traffic conditions at the intersections. For instance, vector $\boldsymbol{x}_1$ reflects all the northbound right turn vehicle's counts and the corresponding occupancies obtained via the system detectors in the first intersection. Four neurons $\boldsymbol{s}_i(t)$ ($i = 1$, 2, 3 and 4) in the hidden layer represent the corresponding traffic states of these intersections at time interval t. The output data $\boldsymbol{y}(t)$ reflects the traffic signal timing plan for the next time instance. To ensure fast and stable learning, the input and output data are linearly scaled to the interval (0.1, 0.9).

3.2 Design of timing plans

A nondominated sorting genetic algorithm (NSGA-Ⅱ)[12] is used to determine a maximum of eight timing plans that can result in minimal delay and queue length of the selected intersections. All the possible signal timing plans are originally designed by Synchro 7 software. The NSGA-Ⅱ algorithm has the ability to select the optimal timing plan among the possible timing plan sets based on the associated traffic states. The selected timing plans are optimized with the associated traffic states so that overall delays and queue lengths are minimized[13].

The NSGA-Ⅱ results in a selection of four timing plans to handle all the traffic states. The optimal timing plan associated with the traffic state at time interval $t+1$ is selected as the output for time interval t during the training phase. The rationale here allows the SSNN-EKF model to predict the optimal timing plan associated with the future traffic state and implement it in a proactive fashion, rather than implementing the optimal timing plan associated with the current state in the future with a lagging or reactive control.

3.3 Model solution

The functions in the neural network toolbox and the Kalman filter toolbox from Matlab are used to develop a program in order to solve the model presented above. In order to obtain the solution of the SSNN-EKF model, the selected model needs to initiate two classes of parameters: neural network weight parameters and KF noise. No prior knowledge has been found to initiate these two parameters, and random parameters are used to examine the initiation. Using real traffic data presented in section 3.1 to inspect the effects on the outputs of the proposed model, we find that, like other heuristic algorithms, the differences between predictions and observations of the SSNN-EKF model are often significant at the beginning. This is because the allocations of weight parameters are inappropriate. However, the error is reduced sharply as more observations become available to the SSNN-EKF model until the deviation is reduced into a reasonable region. After this initial "warm-up" period, the SSNN-EKF model tends to continue making correct predictions.

4 Results and Data Analysis

In order to compare the reliability and accuracy of the SSNN and EKF model, the performance of the SSNN (trained by the Levenberg-Marquardt method) and another nonlinear algorithm, the BP neural network, is investigated. Fig. 4 shows a comparison between the optimal timing plan for the future state at $t+1$ vs. the optimal plan predicted by the BP neural network, the SSNN-LM and the SSNN-EKF model to be implemented at time $t+1$ in a typical day.

Tab. 1 shows the extracted results of the proposed model with those of the other two models. All the results of the three selected models are used to do the correlation analysis with the optimal timing plan obtained from the NSGA-Ⅱ. The statistical results of the differences between the test models and the NSGA-Ⅱ as well as the CPU time used to run the program are also shown in Tab. 1. As shown in Tab. 1, we can see that there is no great difference in these parameters, such as the Pearson correlation, the sum of squares and the cross-products and covariances of the two models. The established results between the SSNN-LM and the SSNN-EKF model are similar just because the basic component uses an SSNN to learn the nonlinear mapping from the input to the output. All the statistical parameters of the BP neural network fall into an unsatisfied region; however, the BP neural network does work well when the traffic state is constant. But when there is a sudden change in the traffic state, it becomes unstable. Focusing on computational time, the BP neural network has excellent performance, while the SSNN uses a relatively long time for its training process. Training with the extended Kalman filter, the computational time of the SSNN is greatly reduced. The test results show that the SSNN-EKF model performs better than the other two models.

Tab. 1 Comparison of selected models tested with real traffic data

Parameter	BPNN	SSNN	SSNN-EKF
Pearson correlation	0.402	0.835	0.897
Sum of squares and cross-products	1.317	2.361	2.848
Covariance	0.013	0.023	0.024
Error timing plan/Total timing plan	38/96	22/96	15/96
CPU time/s	40.8	1 983.3	58.2

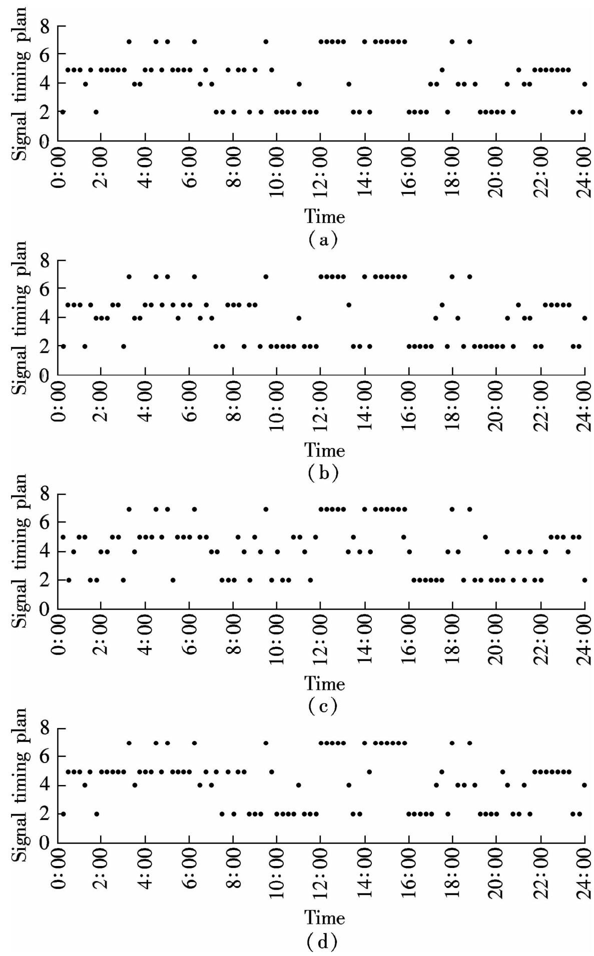

Fig. 4 Comparison of test results with real traffic data. (a) NSGA-Ⅱ; (b) SSNN; (c) BPNN; (d) SSNN-EKF

5 Conclusion

In this paper, an SSNN-EKF model for proactive TRPS traffic signal control is developed. An arterial road network in Guangzhou Road, Nanjing is used to test the performance of the proposed model. The performance of the proposed model is compared with the other two existing models. With optimal timing plans produced by a multi-objective genetic algorithm (NSGA-Ⅱ), the SSNN-EKF model has the best performance of these three models.

The study results are very promising for the traffic control on coordinated actuated traffic signal control. The proposed model can be applied in the master controller for a road network which requires adaptive traffic control without a huge cost. Current coordinated actuated traffic signals control systems can apply the proposed model via selected ports. Further research should be considered to the model preference under congested conditions. Moreover, more algorithms on categorizing timing plans obtained from the model into clusters should be studied.

References

[1] Gordon L, Tighe W. *Traffic control systems handbook*[M]. Washington: FHWA, 1996: 56-69.

[2] Robertson D I, Hunt P B. Optimizing networks of traffic signals in real time—the SCOOT method[J]. *IEEE Transac-*

tions on Vehicular Technology, 1991, **40**(1): 11 − 15.

[3] Lowrie R, The Sydney coordinated adaptive traffic system—principles, methodology, algorithms[C]//*Proc of IEE International Conference on Road Traffic Signaling*. London, 1982: 67 − 70.

[4] Chiu S, Chand S. Self-organizing traffic control via fuzzy logic[C]//*Proc of the IEEE Conference on Decision and Control*. San Antonio, TX, USA, 1993: 1987 − 1902.

[5] Lee J, Abdulhaia B, Shalabya A, et al. Real-time optimization for adaptive traffic signal control using genetic algorithms[J]. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, 2005, **9**(3): 111 − 122.

[6] Mikami S, Kakazu Y. Genetic reinforcement learning for cooperative traffic signal control[C]//*Proc of IEEE Conference on Evolutionary Computation*. Orlando, FL, USA, 1994: 223 − 228.

[7] Ella B. Reinforcement learning in neurofuzzy traffic signal control[J]. *European Journal of Operational Research*, 2001, **131**(2): 232 − 241.

[8] Wei W, Zhang Y. FL-FN based traffic signal control[C]//*Proc of IEEE International Conference on Plasma Science*. Honolulu, HI, USA, 2002: 296 − 300.

[9] Liu H, van Zuylen H J, van Lint H, et al. Predicting urban arterial travel time with state-space neural networks and Kalman filters[J]. *Transportation Research Record*, 2007 (1968): 99 − 108.

[10] van Lint J W C, Hoogendoorn S P, van Zuylen H J. Freeway travel time prediction with state-space neural networks: modeling state-space dynamics with recurrent neural networks [J]. *Transportation Research Record*, 2002(1811): 30 − 39.

[11] Elman J L. Finding structure in time[J]. *Cognitive Science*, 1990(14): 179 − 211.

[12] Deb K, Pratap A, Agarwal S, et al. A fast elitist multi-objective genetic algorithm: NSGA-Ⅱ[J]. *IEEE Transactions on Evolutionary Computation*, 2002, **6**(2): 182 − 197.

[13] Shen F, Yang X G. Research on multi-object optimization algorithm of urban road traffic lights [J]. *Journal of Tongji University: Natural Science*, 2009, **37**(7): 898 − 902. (in Chinese)

基于状态空间神经网络和扩展卡尔曼滤波的主动交通感应控制

过秀成　李　岩　杨　洁

(东南大学交通学院,南京 210096)

摘要:为改善方案选择式交通感应控制输出的交通信号配时方案滞后于实时交通状态的缺点,提出用状态空间神经网络和扩展卡尔曼滤波模型预测未来交通状态的优化配时方案.采用能反映道路网络几何特征的状态空间神经网络拓扑结构,结合当前时段和前一时段的路段交通状态,预测下一时段交通状况并选择与其相匹配的信号配时方案;应用扩展卡尔曼滤波训练状态空间神经网络,提高其训练效率及精度.选用南京市广州路的实测交通数据和由多目标遗传算法得出的最优信号控制方案验证模型的有效性.研究结果表明,与 BP 神经网络和状态空间神经网络相比,所提出的模型能够根据道路状况选择合适的交通控制方案.

关键词:状态空间神经网络;扩展卡尔曼滤波;交通感应控制;配时方案;交通状态预测

中图分类号:U491