Journal

IEEE Transactions on Neural Networks and Learning Systems, 34(10)

ISSN

2162-237X

Authors

Hong, Wanshi
Tao, Gang
Wang, Hong
et al.

Publication Date

2023-10-01

DOI

10.1109/tnnls.2022.3146811

Traffic Signal Control with Adaptive Online-Learning Scheme Using Multiple-Model Neural Networks

Wanshi Hong, Gang Tao, *Fellow, IEEE*, Hong Wang, *Fellow, IET*, Chieh Wang, *Member, IEEE*

***Abstract*—This paper proposes a new traffic signal control algorithm to deal with unknown-traffic-system uncertainties and reduce delays in vehicle travel time. Unknown-traffic-system dynamics are approximated using a recurrent neural network (NN). To accurately identify the traffic system model, an online-learning scheme is developed to switch among a set of candidate NNs (i.e., multiple-model NNs) based on their estimation errors. Then, a bank of optimal signal-timing controllers is designed based on the online identification of the traffic system. Simulation studies have been carried out for the obtained control strategies using multiple-model NNs, and desired results have been obtained. Moreover, compared with the widely used actuated traffic signal control schemes, it is shown that the proposed method can reduce vehicle travel delays and improve traffic system robustness.**

***Index Terms*—Traffic signal control, online learning, multiple-model neural networks**

I. Introduction

TRAFFIC signal control has been an important research topic for decades because efficient control can be extremely beneficial to network traffic systems, bring a safer and smoother traffic flow, and improve economic competitiveness [1]. Increasing automobile traffic demands, especially in urban areas, is creating an urgent need for better traffic signal control strategies.

Traffic signal control uses the concept of *phases*, in which directions of movement are grouped [2]. A traffic *cycle* consists of a set of predefined sequences of traffic phases, which combines green signals allocated to a set of lanes simultaneously for nonconflicting movements at an intersection, with yellow and red signals to provide a safe transition between different movements.

This manuscript has been authored in part by UT-Battelle LLC under contract DE-AC05-00OR22725 with the US Department of Energy (DOE).

W. Hong is in the Sustainable Energy & Environmental Systems Department at Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA (email: wanshihong@lbl.gov).

G. Tao is with the Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22904, USA (email: gt9s@virginia.edu).

H. Wang and C. Wang are with the Buildings and Transportation Science Division, Oak Ridge National Laboratory Oak Ridge, TN 37831, USA (email: wangh6@ornl.gov; cwang@ornl.gov).

Traditionally, there are two types of traffic signal control methods, namely *pretimed* and *actuated*. Pretimed traffic signal control presets the green signal time using the Webster formula based on the historic traffic data collected over different time periods [3]. Actuated traffic signal control, on the other hand, requires detection of traffic movements and sets the green signal based on current traffic conditions rather than historic ones [4]. Self-organizing traffic light control [5], for example, is a fully actuated traffic signal control method with additional demand-responsive rules. With increased traffic demand, both methods have shown some drawbacks. For example, pretimed control does not take current traffic conditions into consideration, making it unresponsive to traffic changes and less robust. Actuated control, on the other hand, does not consider historic traffic information and is much better suited for an isolated intersection control. Moreover, neither method considers the effects of conditions at neighboring intersections, which can be a crucial factor in the presence of traffic congestion.

Many research approaches to improving traffic signal control effectiveness have been considered. For instance, the GreenWave [6] method is a classic control method used in the transportation field. It optimizes the offset to reduce the vehicle stops along a certain direction. However, this method can only optimize unidirectional traffic. To optimize two opposite directions, a control method called Maxband [7, 8, 9] has been developed to optimize the number of vehicle stops by finding a maximum bandwidth based on the signal planning of intersections along an arterial. These methods, although may be suited for coordination of signalized intersections along a corridor, do not take effects of all approaches of neighboring intersections into consideration and thus are not suited for complicated, grid-like urban traffic networks. Max-pressure control [10] is another optimal traffic signal control approach that aims at reducing the risk of traffic flow oversaturation by minimizing the traffic "pressure," which is computed at an intersection between neighboring intersections. However, this method requires a high sampling rate of the "pressure" signal. There are also many artificial intelligence approaches to addressing the traffic signal control problem, such as swarm intelligence [11, 12], fuzzy logic algorithm [13, 14, 15], and reinforcement learning approaches [16, 17, 18]. For example, in [19, 20] the authors use several adaptive dynamic programming approaches to develop an optimal traffic signal control scheme and identify the unknown traffic dynamics. In [21, 22, 23], some data-driven machine-learning approaches are proposed for smart traffic signal control and optimal traffic management.

Indeed, using intelligent techniques for traffic signal control research has become a trend in recent years. Machine-learning-based methods and reinforcement learning, in particular, have become the most popular approaches. However, most learning-based methods require a large number of attempts to "learn" the traffic flow pattern and relevant dynamics. For dynamic programming approaches, historic traffic dynamic analysis is needed at every step of the optimization, which increases the computational complexity and makes them difficult for real-time implementations. For time-series-based dynamic problems like traffic signal control, such complexity would lead to a heavy computation burden at each time step and require extensive data collection. To deal with such drawbacks, this work proposes an online-learning method that calculates the desired optimal control signal within one pass through an intersection. This is realized by taking the traffic signal system as a kind of unknown and uncertain nonlinear system, where modeling and control using the neural network-based on-line learning and control for various uncertain nonlinear systems has been studied in recent years with some remarkable results being documented in [24, 25, 26, 27, 28, 29, 30]. Moreover, the multiple-model approach is an effective approach for adapting the unknown and varying system dynamics by switching to more appropriate controllers followed by tuning or adaptation [31, 32], which is a suitable traffic system control design for quickly changing traffic patterns. Nonetheless, to our best knowledge, there has been no research that explores the use of multiple-model or NN-based online learning approaches for traffic signal control. As a result, to guarantee the responsiveness and reliability of the online-learning-based optimal traffic signal control, a multiple neural network (NN) structure is used here to model the traffic system with an appropriate selection algorithm applied. Fig. 1 illustrates the online-learning-based optimal traffic signal control method developed in this paper.

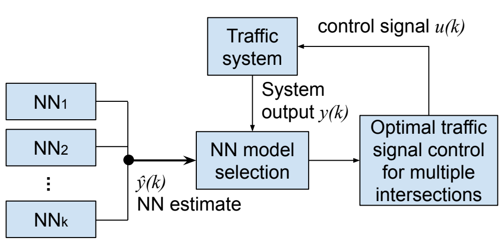


Fig. 1: Online-learning-based optimal traffic signal control.

In this paper, a traffic signal control scheme is developed based on an NN approximation of the original traffic system model that represents the dynamics between the traffic delay and the signal timing (i.e., duration of the green light) at intersections. Instead of testing different NN structures with offline training, we propose the use of an online multiple-model NN scheme in which the original traffic system model is estimated by a set of NNs with different structures, whose weight parameters are updated along with system operation using online-collected signal-timing and traffic-delay data. The best NN can then be determined by a switching mechanism that finds the most accurate approximation of the original traffic system at each time step. In this context, we first extend our previous work [33, 34, 35] on traffic signal control using a linear-model-based online optimal control scheme with an online NN-based control scheme to accurately capture the unknown nonlinear traffic dynamics. We then develop a multiple-model NN-based adaptive control scheme to deal with the case in which both the NN parameters and structures are uncertain in order to cover a wider range of traffic system operations.

The basic idea of a multiple-model NN-based adaptive control scheme is similar to that developed in the multiple-model adaptive control literature, such as the work by Narendra et al. [32]. The specific design of our adaptive control scheme is applied to the traffic signal control system. As shown in the literature, the technical foundations of multiple-model adaptive control are matured, including control switching, stability, and robustness, and are applicable to our adaptive control scheme for the traffic control system.

The focus of our work is on traffic signal control using a single-model or multiple-model NN-based adaptive control design to deal with uncertainties of the traffic control system. The *main contributions* of this paper include:

- development of new multiple-model NN modeling with an adaptive scheme to update the multiple NNs and a switching identification scheme to find the most accurate NN approximation of the unknown traffic system dynamics,
- design of an optimal control scheme for traffic control based on multiple-model NN system identification, and
- evaluation and verification of the effectiveness and advantage of the proposed signal timing control scheme for minimizing traffic delays in comparison to conventional traffic signal control schemes.

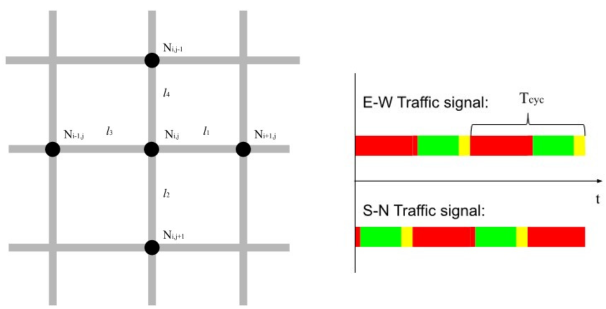
The remainder of this paper is organized as follows: Section II introduces the system model and states the control objective; Section III presents the nominal NN-based system identification design and parameterization, where the accurate approximation of the unknown traffic network system is assumed; Section IV derives the multiple-model NN-based design; Section V develops the adaptive-control-based traffic signal control algorithm; Section VI presents the simulation study to verify the control design; and Section VII draws conclusions based on the simulation results and discusses potential future work.

II. System Model and Control Objective

In this section, the traffic network model is introduced. Then, the control objective is given in terms of total traffic delay minimization when vehicles pass through the network-wide intersections. Once the traffic delay is minimized, it is expected that energy (fuel) consumption will also be minimized for the affected traffic flows.

A. *Traffic Network Model*

To model a traffic network with m intersections, we make the following assumptions that can represent the majority of situations: 1) we assume that each intersection connects two roads, one in the N–S direction and one in the E–W direction; 2) we assume that the standard traffic signal control at each intersection shares the same fixed signal length T_{cyc} with two phases: the E–W approaches share one phase, and the N–S approaches share the other phase. These assumptions are made to simplify the formulation of the control algorithm. Fig. 2 shows an example of the traffic network system, where Fig. 2(a) is the traffic network example; $N_{i,j}$ is the number of the intersection; l_1, l_2, l_3, l_4 are the distances from $N_{i,j}$ to its neighboring intersections; and Fig. 2(b) is the traffic signal timing sequence in terms of green, yellow, and red traffic lights and their timing durations.



(a) Traffic network example (b) Traffic signal sequence example

Fig. 2: Traffic network system.

The traffic network system described above with m intersections can be modeled with a discrete-time nonlinear system as follows:

$$\Delta z_o(t+1) = \psi(\Delta z_o(t)) + h(\Delta z_o(t))\Delta v(t), \qquad (1)$$

where $z_o(t) = [z_{oNS1}(t), z_{oEW1}(t), z_{oNS2}(t), z_{oEW2}(t)\ldots; z_{oNSm}(t), z_{oEWm}(t)]^T \in R^{2m}$ is the system state vector, which is the average traffic delay in the E–W direction and N–S direction at each intersection in the network; and $v(t) = [v_1(t), v_2(t), \ldots, v_m(t)]^T \in R^m$ is the control input, where the components are the green signal percentage for the E–W direction at each intersection. Given the cycle length as T_{cyc}, the green signal percentage is defined as the green signal time duration divided by T_{cyc}. Therefore, the green signal percentage is a variable in $(0, 1)$. In line with the minimum and maximum green signal time duration in practice, v_{min} and v_{max} are defined as the minimum and maximum green signal percentages, respectively. As a result, we will always have $v_i \in [v_{min}, v_{max}]$, $(i = 1, \ldots, m)$, indicating that the green signal percentage should not be less than v_{min} or exceed v_{max} in each direction at every intersection, where $v_{min}, v_{max} \in (0, 1)$ and $v_{min} < v_{max}$ are the prespecified parameters. It can be seen that since the cycle length and the yellow light duration are fixed as well, the green signal percentage can uniquely control the traffic flow through the intersection. For example, when the green signal percentage in the E–W direction is higher, then the traffic flow in the E–W direction will be faster, indicating that the traffic delay along the E–W route will be less. In equation (1), $\Delta z_o(t) = z_o(t) - z_o(t-1)$, $\Delta v(t) = v(t) - v(t-1)$ is the increment of z and v at time step t. $\psi \in R^{2m}$, $h \in R^{2m\times m}$ are the unknown nonlinear functions with respect to Δz_o, $t+1$, and t represents $kT + T$ and kT with $T = T_{cyc}$ being the sampling interval, which is chosen to be the same as the signal cycle length.

For this problem, the states are the average traffic delay at each intersection, and the control signal v will not change within one signal cycle. To simplify the notation, we denote $\Delta z_o = x_o$ and $\Delta v = u$ and rewrite the original nonlinear model as

$$x_o(t+1) = \psi(x_o(t)) + h(x_o(t))u(t). \qquad (2)$$

Because the direct mathematical relationship between traffic delay and green signal period is unknown, ψ and h are unknown nonlinear functions and are very likely to be time-varying. Thus, we need to design a parameterizable model that can be used to accurately approximate and estimate the unknown nonlinear relationship ψ and h online using the measured input (green signal percentage) and the average traffic delay at intersections.

B. Control Objective

The control objective is to design an optimal control input $u(t) = u(kT_{cyc})$ within the range $[u_{min}, u_{max}]$ for the system (2) to achieve minimal travel delay at the $t+1$ time step. u_{min}, u_{max} are the maximum decrement and increment amounts for $u(t)$. This control objective can be summarized by solving the following optimal control problem with cost function and constraint as

$$\min_{u_j(t)\in[u_{min},u_{max}]} J_o(t),\ j = 1, 2, \ldots, m \qquad (3)$$

$$J_o(t) = \mathbb{P}^T x_o(t+1),\ t = 1, 2, \ldots, \qquad (4)$$

where $\mathbb{P} \in R^{2m}$ with every element greater or equal to zero, which indicates the weighted summation of traffic delay decrements x_o (one possible choice of $\mathbb{P}$ is $\mathbb{P} = \mathbb{1} = [1, 1, \ldots, 1]^T \in R^{2m}$), and u_{min}, u_{max} are the lower and upper bounds for the control signal $u(t)$, which indicates the minimum and maximum change in green signal period between signal cycles. These bounds are used to improve system robustness. This cost function will change at each time step $t = kT_{cyc}$, meaning that the optimal controller $u(t)$ is calculated based on $J_o(t)$ at each time step t. The cost function represents the weighted summation of the average traffic delay of all the intersections in the traffic network at the next time step. At each time step t, we can minimize $x_o(t+1)$, indicating a minimum traffic delay operation. Moreover, because the plant to be controlled is a traffic system, the use of historical information in the cost function is not necessary because the traffic system contains high randomness. In this work, we design a one-step optimization scheme.

III. Nominal NN-Based Design and Parameterization

In this section, the nominal NN-based design is presented. First, the nominal NN approximation is given, and then the nominal NN-based control law is derived.

A. Nominal NN Approximation

Given the traffic system contains high randomness because the number of vehicles on the road during any time interval is random, it is difficult to capture all system dynamics in (2) with a linear approximation. To develop an effective traffic signal control scheme, better identification of the traffic network system is necessary. NN approximation has commonly been used for nonlinear complex function approximations and many areas of machine-learning studies. More importantly, use of machine-learning methods to address time series problems has recently become a key focus area [36, 26, 37], which inspired this research to use NN structures in the approximation of the traffic system dynamic model. Using recurrent NNs, a nominal approximation model of (2) can be constructed as

$$x_o(t) = x(t) + \delta(t), \tag{5}$$
$$x(t+1) = A^* x(t) + W_1^* S_1(x(t)) + W_g^* \Phi_g(x(t), u(t)), \tag{6}$$

where $x(t) \in R^{2m}$ is the NN approximation of system state $x_o(t)$, $\delta(t)$ is the approximation error, $u(t)$ is the NN control input, and $A^* \in R^{2m \times 2m}$ is the system matrix. W_1^*, S_1, W_g^* and Φ_g are the parameterized NN components

$$W_g^* = [W_2^*, W_3^*, \dots, W_{m+1}^*] \tag{7}$$
$$\Phi_g(t) = [S_2(x(t))u_1(t), \dots, S_{m+1}(x(t))u_m(t)]^T, \tag{8}$$

where $W_p^* \in R^{n \times l_p}$, $p = 1, 2, \dots, m+1$, l_p is the number of layers for the pth NN, and $S_p(x) \in R$ is the activation function that shows how the pth NN operates on the state x. The NN structure of the pth NN is given as

$$S_p(x(t)) = [S_1^p, S_2^p, \dots, S_{l_p}^p] \in R^{l_p} \tag{9}$$
$$S_j^p(x) = \prod_{i \in I_p} (s^p(x_i))^{d_i^p(j)}, \ j = 1, \dots, l_p \tag{10}$$
$$s^p(x_i) = \frac{k_1}{1 + e^{-x_i} + k_2}, \tag{11}$$

where with $k_1 > 0, k_2 > 0$ being the design parameters, $d_i^p(j)$ are nonnegative integers, and I_p is the subset of set $\{1, 2, \dots, 2m\}$. It has been shown in [38, 39] that ideal matrices $W_1^*, \dots, W_{m+1}^*$ with appropriate $S_1(x), \dots, S_{m+1}(x)$ exist such that (6) can approximate (1) to any degree of accuracy on any compact set.

Parameterization of $x_o(t)$. With (5) – (8), we can further parameterize the original system output $x_o(t)$ as follows:

$$x_o(t) = \theta^* \Phi(t-1) + \delta(t), \tag{12}$$

where

$$\theta^* = [A^*, W_1^*, W_2^*, \dots, W_{m+1}^*] \tag{13}$$
$$\Phi(t) = [x(t)^T, S_1(x(t))^T, (S_2(x(t))u_1(t))^T, \dots, (S_{m+1}(x(t))u_m(t))^T]^T, \tag{14}$$

are the system parameter vector and the regressor vector. Since the ideal matrices $W_1^*, \dots, W_{m+1}^*$ with appropriate $S_1(x), \dots, S_{m+1}(x)$ exist such that (6) can approximate (1) to any degree of accuracy on any compact set. As a result, we can make the assumption that the approximation error $\delta(t)$ is an exponential decay term, whose effect can be ignored. We assume that $\delta(t)$ satisfies

$$|\delta(t)| \le \delta_1(t)||\Phi(t)||_2 + \delta_2(t), \tag{15}$$

where $\delta_1(t) \in L^\infty$ and $\delta_2(t) \in L^\infty$.

B. Nominal Control Design

To achieve the control objectives stated in the previous section, we need to solve the optimization problem defined in (3) – (4). With the assumption in the previous section that (6) is a good approximation of (1), the optimization problem (3) – (4) can be reformulated with the NN approximation model as

$$\min_{u_j^*(t) \in [u_{min}, u_{max}]} J(t), \ j = 1, 2, \dots, m \tag{16}$$
$$J(t) = \mathbb{P}^T x(t+1), \ t = 1, 2, \dots, \tag{17}$$

where x is the NN approximation of the original traffic delay state x_o. When the accurate approximation of the original traffic network system is known, we can design the nominal optimal controller by solving the optimization problem and applying (6) – (17) to obtain

$$J(t) = \mathbb{P}^T x(t+1) = \mathbb{P}^T A^* x(t) + \mathbb{P}^T W_1^* S_1(x(t)) + \mathbb{P}^T W_g^* \Phi_g(x(t), u^*(t)). \tag{18}$$

Because at each time step t the traffic delay term $x(t)$ is measurable, the above equation is linear with respect to the nominal optimal controller $u^*(t)$. This optimization problem therefore becomes a linear programming problem, where $u^*(t)$ is selected to minimize $J(t)$.

Optimal Control Design. To solve for the optimal control signal $u^*(t)$, by taking the derivative of the cost function with respect to $u^*(t)$, we have

$$\begin{aligned} \frac{\partial J(t)}{\partial u^*(t)} &= \frac{\partial \mathbb{P}^T W_g^* \Phi_g(x(t), u^*(t))}{\partial u^*(t)} \\ &= [\frac{\partial \mathbb{P}^T (W_2^* S_2(x(t)) u_1^*(t))}{\partial u_1^*(t)}, \dots, \\ &\quad \frac{\partial \mathbb{P}^T (W_{m+1}^* S_{m+1}(x(t)) u_m^*(t))}{\partial u_m^*(t)}] \\ &= d^*(t), \end{aligned} \tag{19}$$

with $x(t)$ available at time step t and $S_{m+1}(x(t))$ as a constant at each time step t. We then have

$$d^*(t) = [d_1^*(t), d_2^*(t) \dots, d_m^*(t)]^T \in R^m \tag{20}$$
$$\begin{aligned} d_j^*(t) &= \frac{\partial \mathbb{P}^T (W_{j+1}^* S_{j+1}(x(t)) u_j^*(t))}{\partial u_j^*(t)} \\ &= \mathbb{P} W_{j+1}^* S_{j+1}(x(t)), \ j = 1, 2, \dots, m. \end{aligned} \tag{21}$$

With $x(t)$ known at time step t, $d_j^*(t)$ is a constant at every time step, indicating that the control design for $u^*(t)$ is a linear programming problem. The control input for minimal cost function J lies on the boundary of $u^*(t)$ based on the sign of $d_j^*(t)$ at any time step t. As a result, we arrive at the following control design:

$$u_j^*(t) = \begin{cases} u_{max}, & d_j^*(t) < 0 \\ 0, & d_j^*(t) = 0 \\ u_{min}, & d_j^*(t) > 0, \end{cases} \tag{22}$$

where for $d_j^*(t) = 0, j = 1, \dots, m$, the solution for $u_j^*(t)$ is trivial. Thus, we manually set $u_j^*(t) = 0$ for this situation.

System Robustness. For the linear programming problem, the solution for the control signal $u^*(t)$ will always occur on one of the boundaries. Because the traffic system operates around $z(t)$ and $v(t)$ at time step t, to guarantee the system's robustness, the maximum increment for the control signal $u^*(t)$ should be relatively small (i.e., $|u^*(t)| < \bar{u}$, $u_{max} = \bar{u}$, $u_{min} = -\bar{u}$, and $\bar{u} < 0.1$ is the maximum amount of increment for $u^*(t)$). Moreover, the boundary condition for $v_j(t) \in [v_{min}, v_{max}]$ should be satisfied at the same time, with $v_{min} = 0.2,\ v_{max} = 0.8$ as the minimum and maximum green cycle percentage selected to guarantee the shortest green signal period on each direction. Therefore, the green signal percentage $v_j(t)$, $j = 1, \dots, m$ can be determined based on the following conditions:

$$v_j(t) = \begin{cases} v_j(t-1) + u_j^*(t) & v_j(t-1) + u_j^*(t) \in [v_{min}, v_{max}] \\ v_{min}, & v_j(t-1) + u_j^*(t) < v_{min} \\ v_{max}, & v_j(t-1) + u_j^*(t) > v_{max} \end{cases} \tag{23}$$

Note that this nominal NN approximation of the original traffic system may vary as the traffic operation condition changes (e.g., traffic volume variation, vehicle speed variation). It is therefore difficult to know the nominal NN model at all times, so in this paper we use a multiple-model NN estimation scheme to estimate the original traffic network system.

IV. Multiple-Model NNs and Parameterization

Although ideal matrices $W_1^*, \dots, W_{m+1}^*$ with appropriate $S_1(x), \dots, S_{m+1}(x)$ exist such that (6) can accurately approximate (1). In reality, the exact values of $W_1^*, \dots, W_{m+1}^*$ are unavailable, so we need to design an appropriate parameter update law to estimate $W_1^*, \dots, W_{m+1}^*$. Moreover, unlike offline NN approximations in which we can test and compare the approximation accuracy with different NN models, it is difficult to find the ideal approximation model with only one choice of NN structure in online approximations. Thus, in this section, we propose use of multiple-model-based NNs to approximate the traffic network system in (1).

A. Multiple-Model-Based NN Approximation

To accurately approximate (1) using NNs, we propose use of N multiple NN structures. Using recurrent NNs, the approximation models can be constructed as

$$\begin{aligned} x_{(i)}(t+1) = & A_{(i)}^* x_{(i)}(t) + W_{(i)1}^* S_{(i)1}(x_{(i)}(t)) \\ & + W_{(i)g}^* \Phi_{(i)g}(x_{(i)}(t), u_{(i)}(t)), \end{aligned} \tag{24}$$

where $x_{(i)}(t) \in R^{2m}$, $u_{(i)}(t) \in R^m$ are the system state and control input, respectively; $A_{(i)}^* \in R^{n \times n}$ are the system matrix; $i = 1, 2, \dots, N$, represents the ith NN in the multiple-model NN family; and $W_{(i)1}^*$, $S_{(i)1}$, $W_{(i)g}^*$ and $\Phi_{(i)g}$ are the parameterized NN components

$$W_{(i)g}^* = [W_{(i)2}^*, W_{(i)3}^*, \dots, W_{(i)m+1}^*] \tag{25}$$

$$\begin{aligned} \Phi_{(i)g}(t) = & [S_{(i)2}(x_{(i)}(t)) u_{(i)1}(t), \dots, \\ & S_{(i)m+1}(x_{(i)}(t)) u_{(i)m}(t)]^T, \end{aligned} \tag{26}$$

where $W_{(i)p}^* \in R^{n \times l_{(i)p}}$, $p = 1, 2, \dots, m+1$, $l_{(i)p}$ is the number of layers for the pth NN, and $S_{(i)p}(x(t)) = [S_1^p, S_2^p, \dots, S_{l_{(i)p}}^p] \in R^{l_{(i)p}}$ are the basis functions. $W_{(i)p}$ and $S_{(i)p}$ are the different NN structures used to approximate the original traffic network system. With this multiple-model-based online approximation design, it is more likely that we can find an ideal NN structure to approximate a traffic network system in (1).

It can be seen that the proposed multiple NN model in equation (24) consists of a number of simple NN with each being dependent on operating conditions. This would produce a simplified whole model for the traffic system rather than using a single NN which in general would be complicated in structure with a large number of weights to be trained.

B. Parameterization of $x_o(t)$

. With the multiple-model-based NN approximation, the original traffic delay vector $x_o(t)$ is expressed as

$$x_o(t) = x_{(i)}(t) + \delta_{(i)}(t), \tag{27}$$

where $\delta_{(i)}(t)$ is the approximation error of the ith NN approximation. With (24) – (26), we can further parameterize the system output $x_o(t)$ as follows:

$$x_o(t) = \theta_{(i)}^* \Phi_{(i)}(t-1) + \delta_{(i)}(t), \tag{28}$$

where

$$\theta_{(i)}^* = [A_{(i)}^*, W_{(i)1}^*, W_{(i)2}^*, \dots, W_{(i)m+1}^*] \tag{29}$$

$$\begin{aligned} \Phi_{(i)}(t) = & [x_{(i)}(t)^T, S_{(i)1}(x_{(i)}(t))^T, (S_{(i)2}(x_{(i)}(t)) u_{(i)}(t))^T, \\ & \dots, (S_{(i)m+1}(x_{(i)}(t)) u_{(i)}(t))^T]^T \end{aligned} \tag{30}$$

are the system parameter vectors and the regressor vectors, respectively, $i = 1, 2, \dots, N$.

With the nominal multiple-model NN approximation derived in this section, we can start the adaptive control design to meet the objective (i.e., to reduce travel delays under the unknown traffic network dynamics).

V. Adaptive Control Design

In this section, the adaptive control scheme is designed for the NN-approximation-based traffic network systems. First, the design of adaptive control for a single NN-based traffic network system approximation is presented. Next, the multiple-model NN-based adaptive control design is presented.

A. Single NN-Based Adaptive Control Design

Following the nominal control design structure, we first design the optimal controller based on the estimated parameters of the unknown system, and then we derive the parameter estimation law to guarantee the desired system performance.

Optimal Control Design. The NN approximation of the cost function is defined in (16). Because the NN parameters are unknown at time step t, assume that the system is observable, we can only obtain the estimate of $x(t+1)$ as $\hat{x}(t+1)$. The estimate optimization problem is defined as

$$\hat{J}(t) = \mathbb{P}^T \hat{x}(t+1) \tag{31}$$

$$u_j(t) \in [u_{min}, u_{max}], \quad j = 1, 2, \ldots, m, \tag{32}$$

where $\hat{J}(t)$ is the estimate of the cost function $J(t)$, $t = 1, 2, \ldots$. In addition, we define

$$d_j(t) = \mathbb{P} W_{j+1}(t) S_{j+1}(\hat{x}(t)) \tag{33}$$

$$W_g(t) = [W_2(t), W_3(t), \ldots, W_{m+1}(t)], \tag{34}$$

where $W_g(t)$ are the estimates of the nominal NN weight parameter W_g^*, $j = 1, \ldots, m$. Based on the certainty equivalence principle of using parameter estimates, we propose the optimal control design as

$$u_j(t) = \begin{cases} u_{max}, & d_j(t) < 0 \\ 0, & d_j(t) = 0 \\ u_{min}, & d_j(t) > 0, \end{cases} \tag{35}$$

where $W_{j+1}(t)$ are the estimates of the nominal NN weight parameter W_{j+1}^*.

Parameter Estimation. To obtain the state estimation $\hat{x}(t)$, we define the estimated model as

$$\hat{x}(t+1) = \theta(t)\Phi(t), \tag{36}$$

where $\theta(t)$ is the estimate of the system parameter θ^*. To develop the adaptive update laws for $\theta(t)$, we introduce the estimation error

$$\varepsilon(t) = \theta(t-1)\Phi(t-1) - x_o(t). \tag{37}$$

Based on a gradient algorithm, we choose the following adaptive parameter update laws as

$$\theta(t) = \theta(t-1) - \frac{\Gamma\Phi(t-1)\varepsilon(t)}{m^2(t)} + f(t), \tag{38}$$

with $\Gamma = \Gamma^T > 0$ as a gain matrix, $m(t) = \sqrt{1 + \alpha\Phi^T(t-1)\Phi(t-1)}$ with $\alpha > 0$ being a design parameter, $f(t)$ as a modification term for robustness with respect to $\delta(t)$, and $\theta(0) = \theta_0$ as the initial estimate of θ^*.

Parameter Projection. To guarantee robustness with respect to $\delta(t)$, we choose a parameter projection design for $f(t)$. We assume the jth element of $\theta^* \in R^n$ belongs to a known interval $\theta_j^* \in [\theta_j^a, \theta_j^b]$, $j = 1, \ldots, n$, and choose the gain matrix in (38) as

$$\Gamma = \text{diag}\{\gamma_1, \ldots, \gamma_n\}, \ \gamma_j \in (0, 2), \ j = 1, \ldots, n. \tag{39}$$

We then define the base adaptation vector as

$$g(t) = -\frac{\Gamma\varepsilon(t)\Phi(t-1)}{m^2(t)}. \tag{40}$$

Denoting the jth component of $\theta(t)$, $f(t)$, $g(t)$ as $\theta_j(t)$, $f_j(t)$, $g_j(t)$, respectively, we choose the initial parameter estimate as $\theta_j(0) \in [\theta_j^a, \theta_j^b]$. We set the projection function components as

$$f_j(t) = \begin{cases} 0 & \text{if } \theta_j(t-1) + g_j(t) \in [\theta_j^a, \theta_j^b], \\ \theta_j^b - \theta_j(t-1) - g_j(t), & \text{if } \theta_j(t-1) + g_j(t) > \theta_j^b, \\ \theta_j^a - \theta_j(t-1) - g_j(t), & \text{if } \theta_j(t-1) + g_j(t) < \theta_j^a. \end{cases} \tag{41}$$

Then, with the above choice of $f(t)$, the adaptive law (38) has the desired properties:

Lemma 1: The adaptive parameter law (38) for the system (1) ensures that $\theta(t) \in L^\infty$, $\frac{\varepsilon(t)}{m(t)} \in L^\infty$, and

$$\sum_{t=t_1}^{t_2} \frac{\varepsilon^2(t)}{m^2(t)} \leq a_1 + b_1 \sum_{t=t_1}^{t_2} \frac{\delta^2(t)}{m^2(t)} \tag{42}$$

for some constant $a_1 > 0$, $b_1 > 0$, and all $t_2 > t_1 \geq 0$.

Proof: We define the positive definite function $V(\tilde{\theta}) = \tilde{\theta}^T\Gamma^{-1}\tilde{\theta}$. With adaptive law (38) and the estimation definition (37) and (12), we have

$$\varepsilon(t) = \tilde{\theta}(t-1)\Phi(t-1) - \delta(t), \tag{43}$$

where $\tilde{\theta}(t) = \theta(t) - \theta^*$ and calculate the increment of $V(\tilde{\theta})$ as

$$\begin{aligned} & V(\tilde{\theta}(t)) - V(\tilde{\theta}(t-1)) \\ =& -(2 - \frac{\Phi^T(t-1)\Gamma\Phi(t-1)}{m^2(t)})\frac{\varepsilon^2(t)}{m^2(t)} + 2\frac{\varepsilon(t)\delta(t)}{m^2(t)} \\ &+ 2f^T\Gamma^{-1}(\tilde{\theta}(t-1) - \frac{\Gamma\varepsilon(t)\Phi(t-1)}{m^2(t)} + f(t)) - f^T(t)\Gamma^{-1}f(t). \end{aligned} \tag{44}$$

From condition (15), we can see that the following condition is satisfied for some constant $c_1 > 0$ and $c_2 > 0$.

$$\frac{|\delta(t)|}{m(t)} < c_1 + \frac{c_2}{m(t)}. \tag{45}$$

With the parameter projection design of $f(t)$ given in (41), the following property holds

$$f_j(t)(\theta_j(t-1) - \theta_j^* + g_j(t) + f_j(t)) \leq 0, \ j = 1, 2, \ldots, n. \tag{46}$$

Using the inequality below [40] and (44), (46)

$$\begin{aligned} & -(2 - \frac{\Phi^T(t-1)\Gamma\Phi(t-1)}{m^2(t)})\frac{\varepsilon^2(t)}{m^2(t)} + 2\frac{\varepsilon(t)\delta(t)}{m^2(t)} \\ \leq & \frac{\alpha_1\varepsilon^2(t)}{2m^2(t)} - \frac{\alpha_1}{2}(\frac{|\varepsilon(t)|}{m(t)} - \frac{2|\delta(t)|}{\alpha_1 m(t)})^2 + \frac{2\delta^2(t)}{\alpha_1 m^2(t)}, \end{aligned} \tag{47}$$

where $\alpha_1 = 2 - \lambda_{max}[\Gamma] > 0$, with $\lambda_{max}[\Gamma] \in [0, 2]$ being the maximum eigenvalue of Γ, we have

$$\begin{aligned} & V(\tilde{\theta}(t)) - V(\tilde{\theta}(t-1)) \\ \leq & \frac{\alpha_1\varepsilon^2(t)}{2m^2(t)} - \frac{\alpha_1}{2}(\frac{|\varepsilon(t)|}{m(t)} - \frac{2|\delta(t)|}{\alpha_1 m(t)})^2 + \frac{2\delta^2(t)}{\alpha_1 m^2(t)} \\ & - f^T(t)\Gamma^{-1}f(t). \end{aligned} \tag{48}$$

We then see (42) holds, and $\theta(t) \in L^\infty$, $\frac{\varepsilon(t)}{m(t)} \in L^\infty$. □

Remark 1: The above adaptive scheme can guarantee the boundedness of the parameter estimation and estimation error.

When modeling error $\delta(t) = 0$, we have $\theta(t) \in L^\infty$, $\frac{\varepsilon(t)}{m(t)} \in L^2 \cap L^\infty$, $\theta(t) - \theta(t-1) \in L^2$.

The parameter projection algorithm (41) not only guarantees the desired robustness properties with respect to modeling error but also ensures that the parameter estimation $\theta(t)$ stays in the same intervals with their true values. This feature can improve the control design performance and ensure the stability of the traffic network system.

From (43), we can see that the performance of the designed adaptive control scheme depends on the nominal approximation error $\delta(t)$ defined in (12) as does that of the nominal control design (22) (which uses the nominal system parameter θ^* defined in (13)). The nominal approximation error $\delta(t)$ depends on the choice of the NN approximator (6) and is a fixed error signal with a fixed upper bound.

To reduce the effect of such an approximation error, in the next subsection we propose a multiple-model-based adaptive NN control scheme that employs a bank of approximators and a bank of adaptive estimation schemes. A switching mechanism is used to select the best approximator with the minimal estimation error to match the approximator with the smallest approximation error.

B. Multiple-Model NN-Based Adaptive Control Design

While using a single NN-based adaptive control design given in the previous section can achieve the desired objective, the performance can be further improved by using multiple-model NN-based adaptive control design. Using multiple-model NNs, we can select the NN that best represents the current unknown system dynamic on line and achieve better estimation results.

Bank of Control Designs. The multiple-model-based approximation of the cost function (3) – (4) using NNs in (24) is defined as

$$J_{(i)}(t) = \mathbb{P}^T x_{(i)}(t+1) \tag{49}$$

$$u_{(i)j}(t) \in [u_{min}, u_{max}], \quad t = 1, 2, \ldots, \quad j = 1, 2, \ldots, m, \tag{50}$$

where $J_{(i)}$, $x_{(i)}$ are the ith NN approximation of the cost function (3) – (4) and traffic delay vector x_o, $i = 1, 2, \ldots, N$. The estimate of the above cost function is

$$\hat{J}_{(i)}(t) = \mathbb{P}^T \hat{x}_{(i)}(t+1) \tag{51}$$

$$u_{(i)j}(t) \in [u_{min}, u_{max}], \quad t = 1, 2, \ldots, \quad j = 1, 2, \ldots, m, \tag{52}$$

where $\hat{J}_{(i)}$, $\hat{x}_{(i)}$ are the estimates of the ith NN approximation to the cost function (51) – (52) and traffic delay vector $x_{(i)}$. We define $d_{(i)j}(t) = \mathbb{P} W_{(i)j+1}(t) S_{(i)j+1}(\hat{x}_{(i)}(t))$, and based on the derivation in Section V-A, we come up with the following optimal control design:

$$u_{(i)j}(t) = \begin{cases} u_{max}, & d_{(i)j}(t) < 0 \\ 0, & d_{(i)j}(t) = 0 \\ u_{min}, & d_{(i)j}(t) > 0, \end{cases} \tag{53}$$

where $W_{(i)j+1}(t)$ is the jth estimate of the nominal parameter W^*_{j+1}.

Bank of Parameter Estimators. To obtain the state estimation $\hat{x}_{(i)}(t)$, we define the estimated model as

$$\hat{x}_{(i)}(t+1) = \theta_{(i)}(t) \Phi_{(i)}(t), \tag{54}$$

where $\theta_{(i)}(t)$ is the ith NN estimate of the system parameter θ^*. To develop the adaptive update laws for $\theta_{(i)}(t)$, we introduce the estimation error

$$\varepsilon_{(i)}(t) = \theta_{(i)}(t-1) \Phi_{(i)}(t-1) - x_o(t). \tag{55}$$

From (28), we have

$$\varepsilon_{(i)} = \tilde{\theta}_{(i)}(t-1) \Phi_{(i)}(t-1) - \delta_{(i)}(t). \tag{56}$$

Based on a gradient algorithm, we choose the adaptive parameter update laws for $\theta_{(i)}(t)$ as

$$\theta_{(i)}(t) = \theta_{(i)}(t-1) - \frac{\Gamma_{(i)} \Phi_{(i)}(t-1) \varepsilon_{(i)}(t)}{m^2(t)} + f_{(i)}(t), \tag{57}$$

with $\Gamma_{(i)} = \Gamma_{(i)}^T > 0$ as a gain matrix,$\theta_{(i)}(0) = \theta_{(i)0}$, $m(t) = \sqrt{1 + \alpha \Phi_{(i)}^T(t-1) \Phi_{(i)}(t-1)}$ with $\alpha > 0$ being a design parameter, and $\theta_{(i)0}$ as the initial estimate of $\theta^*_{(i)}$, $i = 1, 2, \ldots, N$.

Parameter Projection. To guarantee the robustness with respect to $\delta_{(i)}(t)$, which is the modeling error of the ith NN estimate, we choose a parameter projection design for $f_{(i)}(t)$, $i = 1, 2, \ldots, N$. Assume that the jth element of $\theta^*_{(i)} \in R^n$ belongs to a known interval $\theta^*_{(i)j} \in [\theta^a_{(i)j}, \theta^b_{(i)j}]$, $j = 1, \ldots, n$, and choose the gain matrix in (38)

$$\Gamma_{(i)} = \text{diag}\{\gamma_{(i)1}, \ldots, \gamma_{(i)n}\}, \ \gamma_{(i)j} \in (0, 2), \ j = 1, \ldots, n. \tag{58}$$

We then define the base adaptation vector as

$$g_{(i)}(t) = -\frac{\Gamma_{(i)} \varepsilon_{(i)}(t) \Phi_{(i)}(t-1)}{m^2(t)}. \tag{59}$$

Denoting the jth component of $\theta_{(i)}(t)$, $f_{(i)}(t)$, $g_{(i)}(t)$ as $\theta_{(i)j}(t)$, $f_{(i)j}(t)$, $g_{(i)j}(t)$, we choose the initial parameter estimate as $\theta_{(i)j}(0) \in [\theta^a_{(i)j}, \theta^b_{(i)j}]$. We set the projection function components as

$$f_{(i)j}(t) = \begin{cases} 0 & \text{if } \theta_{(i)j}(t-1) + g_{(i)j}(t) \\ & \quad \in [\theta^a_{(i)j}, \theta^b_{(i)j}], \\ \theta^b_{(i)j} - \theta_{(i)j}(t-1) & \text{if } \theta_{(i)j}(t-1) + g_{(i)j}(t) \\ \quad -g_{(i)j}(t), & \quad > \theta^b_{(i)j}, \\ \theta^a_{(i)j} - \theta_{(i)j}(t-1) & \text{if } \theta_{(i)j}(t-1) + g_{(i)j}(t) \\ \quad -g_{(i)j}(t), & \quad < \theta^a_{(i)j}. \end{cases} \tag{60}$$

Then, with the above choice of $f_{(i)}(t)$, using a derivation similar to Lemma 1, the adaptive law (57) has the desired properties:

Lemma 2: The adaptive parameter law (57) for system (1) ensures that $\theta_{(i)}(t) \in L^\infty$, $\frac{\varepsilon_{(i)}(t)}{m(t)} \in L^\infty$ and

$$\sum_{t=t_1}^{t_2} \frac{\varepsilon^2_{(i)}(t)}{m^2(t)} \leq a_2 + b_2 \sum_{t=t_1}^{t_2} \frac{\delta^2_{(i)}(t)}{m^2(t)}, \quad i = 1, 2, \ldots, N, \tag{61}$$

for some constant $a_2 > 0$, $b_2 > 0$, and all $t_2 > t_1 \geq 0$.

Control Switching Scheme. According to the above design, N adaptive controllers are obtained, and a control switching scheme is needed to choose the current controller. The switching scheme in this study is designed based on the estimation cost with different NN models. We define the estimation cost as

$$J_{(i)2}(t) = \sum_{j=1}^{t} \varepsilon_{(i)}^{T}(j) Q \varepsilon_{(i)}(j) \tag{62}$$

for $i = 1, 2, \ldots, N$, where $Q = Q^T > 0$ is the constant design parameter matrix, which is chosen as the identity matrix in our control design. Different choices of Q can be used to set estimation priorities for each states. The control signal $u(t)$ is obtained by

$$u(t) = u_{(j)}(t),\ j = \underset{j=1,2,\ldots,N}{\arg\min}\ J_{(j)2}(t). \tag{63}$$

To prevent arbitrarily fast switching, we introduce a nonzero waiting time $T_{min} > 0$ between two switches. This mechanism guarantees the switching time between two switches is always greater than T_{min}.

The control switching algorithm (63) selects the adaptive estimation scheme with the minimal $J_{(i)2}$ corresponding to a minimal estimation error $\epsilon_{(i)}(t)$, which satisfies (61), to match the best of the approximators in (28), with the minimal approximation error $\delta_{(i)}(t)$. This explains the potential advantage of the multiple-model-based adaptive NN control scheme over its single-model counterpart developed in the last subsection.

Computation Complexity. Note that although the proposed algorithm uses NNs, the computation time is relatively efficient. For each time step, the time complexity is $O(m_{nn})$, with the modeling complexity $m_{nn} = M_1 + \cdots + M_N$, M_1 being the number of total nodes in the ith candidate NN. Note that with the NN models preselected, the time complexity for each time step can be further reduced to $O(1)$. The time complexity is equivalent to performing one NN prediction. In comparison, with dynamic programming approaches [41, 42], the time complexity at each time step is $O(n * m)$, with n being the historic time step needing to be calculated per time step and m being the modeling complexity at each iteration. Moreover, the conventional offline learning methods [43] have a time complexity within the range $(O(n^2) \sim O(n^3))$. As a result, the proposed multiple-model NN-based control scheme is computationally effective compared with other smart traffic control approaches.

C. Summary

In this section, the adaptive control scheme is designed for the NN-approximation-based traffic network systems. The single NN-based adaptive control design is first derived. The single NN-based adaptive control design algorithm is summarized in Algorithm 1,
where T_{end} is the end of simulation time.

Then, to achieve a better control performance in the presence of an online-learning mechanism, a multiple-model NN-based adaptive control with a control switching module is designed. The multiple-model NN-based adaptive control design algorithm is summarized in Algorithm 2.

Algorithm 1 Single NN-based adaptive control

```
initialization: θ(0) = θ_0, u(0) = 0, x̂(0) = 0
for t = 0,1,2,...,T_end do
    obtain state estimation from (36)
    update parameter estimation using (37) – (41)
    apply optimal control design (31) – (35) to traffic system,
end for
```

Algorithm 2 Multiple-model NN-based adaptive control

```
initialization: θ_(i)(0) = θ_(i)0, u_(i)(0) = 0, x̂_(i)(0) = 0
for t = 0,1,2,...,T_end do
    obtain state estimations of different NNs from (54)
    update parameter estimation using (55) - (60)
    calculate the bank of optimal control design using (51) - (53)
    use (62) and (63) to find the best controller
    apply optimal control to original traffic system
end for
```

In summary, the proposed multiple-model NN-based control scheme can well identify the unknown traffic network system and reduce traffic delays.

VI. Simulation Study

In this section, a simulation study for the proposed multiple-model NN-based optimal control is presented. First, the simulation system is introduced, and then the simulation results are presented. Finally, further discussion is provided based on the simulation results.

A. The Simulation System

In this section, the simulation system is introduced. In this simulation study, we consider a simple, two-intersection traffic system case. First, the traffic system model is introduced, and then the NN models used to approximate the original traffic system model are provided.

1) Traffic System Model: For this study, we consider a two-intersection traffic system. The traffic system structure is shown in Fig. 3. The roads in this traffic system are “Road 1”, “Road 2”, and “Road 12”, where ”Road 12” is considered the main road, and ”Road 1” and ”Road 2” are side roads. $v_1 = 30$ mph, $v_2 = 30$ mph, and $v_{12} = 45$ mph are the vehicle speed limits on the roads. N_1 and N_2 are the two intersections, u_1 and u_2 are the traffic control signals at the intersections, and l_{12} is the distance between N_1 and N_2. We assume vehicles randomly enter Road 1, Road 2, and Road 12 every second with the probability s_1, s_2, s_{12}, respectively. The traffic signal cycle length is chosen as $T_{cyc} = 80s$. Note that for this traffic system, we assume that the average vehicle speed and the traffic volume on the two directions of each road are the same and that the turning signals are fixed in each phase for simplicity. For this traffic system, we consider three traffic flow scenarios for our simulation study:

Case 1: sparse traffic flow. The average traffic in flow on each road is $s_1 = 100$ vehicles per hour, $s_2 = 180$ vehicles per hour, and $s_{12} = 360$ vehicles per hour.
Case 2: normal traffic flow. The average traffic in flow on each road is $s_1 = 180$ vehicles per hour, $s_2 = 360$ vehicles per hour, and $s_{12} = 540$ vehicles per hour.
Case 3: dense traffic flow. The average traffic in flow on each road is $s_1 = 360$ vehicles per hour, $s_2 = 540$ vehicles per hour, and $s_{12} = 900$ vehicles per hour.

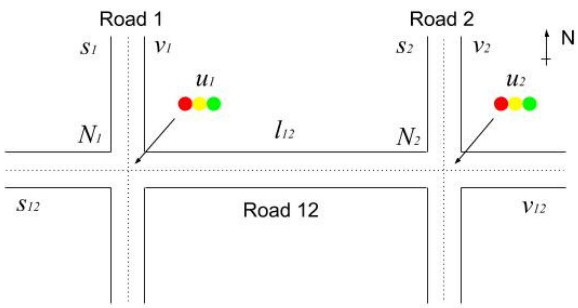


Fig. 3: Two-intersection traffic system example.

2) NN Approximation Model: The NN models used to approximate the above traffic system are shown below.

Nominal NN Approximation Model. The two-intersection traffic system described above can be modeled as

$$x_o(t+1) = \psi(x_o(t)) + h(x_o(t))u(t), \tag{64}$$

where $x_o = [x_{oNS1}, x_{oEW1}, x_{oNS2}, x_{oEW2}]^T \in R^4$ are the traffic delays of both directions at each intersection, and $u = [u_1, u_2]^T$ is the control signal, which is the green signal period of the E–W direction. This original traffic system can be approximated by a nominal single NN model in some traffic conditions as

$$x(t+1) = A^* x(t) + W_1^* S_1(x(t)) + W_g^* \Phi_g(x(t), u(t)) \tag{65}$$

$$= \theta^* \Phi(t), \tag{66}$$

where x is the approximation of x_o, A^*, and W_1^*, W_g^* are the nominal system parameters, whose values are listed as follows:

$$\theta^* = [A^*, W_1^*, W_2^*, W_3^*] \tag{67}$$

$$\Phi(t) = [x(t)^T, S_1(x(t))^T, (S_2(x(t))u_1(t))^T, (S_3(x(t))u_2(t))^T]^T, \tag{68}$$

$$A^* = \begin{bmatrix} 0.9884 & -0.0467 & 0.0058 & 0.0017 \\ -0.0071 & 0.9643 & 0.0104 & -0.0017 \\ -0.0044 & -0.0341 & 1.0047 & -0.0073 \\ -0.0071 & -0.02937 & 0.0122 & 1.0060 \end{bmatrix}, \tag{69}$$

$$W_1^* = \begin{bmatrix} 0.9286 & 0.5130 & 0.2319 & 0.3595 \\ 0.7592 & 0.4136 & 0.7967 & 0.5036 \\ 0.6176 & 0.4858 & 0.8501 & 0.2002 \\ 0.5503 & 0.2894 & 0.3402 & 0.9055 \end{bmatrix}, \tag{70}$$

$$W_g^* = [W_2^*, W_3^*] \tag{71}$$

$$W_2^* = \begin{bmatrix} 0.1949 & 0.3273 & 0.3026 & 0.0873 \\ 0.6075 & 0.2799 & 0.9113 & 0.5158 \\ 0.8831 & 0.3655 & 0.2541 & 0.9572 \\ 0.4195 & 0.7796 & 0.5760 & 0.1937 \end{bmatrix}, \tag{72}$$

$$W_3^* = \begin{bmatrix} 0.8512 & 0.8501 & 0.1778 & 0.6718 \\ 0.5600 & 0.0988 & 0.3279 & 0.4857 \\ 0.4690 & 0.7370 & 0.2108 & 0.4138 \\ 0.5000 & 0.1745 & 0.0159 & 0.3580 \end{bmatrix}. \tag{73}$$

Note that A^* and W^* matrices are determined based on the simulation of a static traffic flow scenario, where we give an initial estimate of the system parameters and use the online learning scheme to find the value for this static traffic flow scenario. These A* and W* are later served as the initial guess for dynamic traffic system estimation and control.

Based on (14), we have $\Phi_g(t) = [S_2(x(t))u_1(t), S_3(x(t))u_2(t)]^T$. $S_1(x(t))$, where $S_2(x(t))$, $S_3(x(t))$ are the activation functions, which have the following structures:

$$sig(x) = \frac{1}{1+e^{-x}} + 0.5 \tag{74}$$

$$S_1(x(t)) = \begin{bmatrix} sig(x_{NS1})sig(x_{NS2}) \\ sig(x_{EW1}) \\ sig(x_{EW2})sig(x_{NS2}) \\ sig(x_{EW1})sig(x_{EW2}) \end{bmatrix}, \tag{75}$$

$$S_2(x(t)) = \begin{bmatrix} sig(x_{NS1}) \\ sig(x_{NS1})sig(x_{EW1}) \\ sig(x_{NS2}) \\ sig(x_{EW2})sig(x_{NS2}) \end{bmatrix}, \tag{76}$$

$$S_3(x(t)) = \begin{bmatrix} sig(x_{NS1})sig(x_{EW1}) \\ sig(x_{EW1}) \\ sig(x_{EW2})sig(x_{NS2}) \\ sig(x_{EW2}) \end{bmatrix}. \tag{77}$$

Remark 2: Note that this NN approximation system (65) is only suitable for normal operating conditions of the benchmark traffic system described in Section VI-A1, Case 1. Even the best NN approximation may change over different operating conditions for the benchmark traffic system. As a result, to accurately capture the traffic system in Section VI-A1, multiple NN estimation structures are used to identify the dynamics of the benchmark system. The above nominal NN system provides initial guesses for the online-learning NN system parameters and NN structures. The online-learning and control simulation study directly estimates and controls the original traffic system in (64).

Single NN-Based Optimal Control Simulation. The single NN estimate of the model (65) is given as

$$\hat{x}(t+1) = A(t)\hat{x}(t) + W_1(t)S_1(\hat{x}(t)) + W_g(t)\Phi_g(\hat{x}(t), u(t)) = \theta(t)\Phi(t), \tag{78}$$

where $\theta(t)$ is the estimate of the nominal parameter θ^*.

Multiple-Model NN-Based Optimal Control Simulation. The multiple-model NN estimate of the model (65) is

$$\hat{x}_{(i)}(t+1) = A_{(i)}(t)\hat{x}_{(i)}(t) + W_{(i)1}(t)S_{(i)1}(\hat{x}_{(i)}(t)) + W_{(i)g}(t)\Phi_{(i)g}(\hat{x}_{(i)}(t), u_{(i)}(t)) \tag{79}$$

$$= \theta_{(i)}(t)\Phi(t),$$

where $\theta_{(i)}(t)$ is the ith NN estimate of the nominal parameter θ^*, and the initial parameter estimation $\theta_{(i)}(0)$ is chosen to be close to the nominal parameter $\theta_{(i)}(0) = 0.8\theta^*$ for $i = 1, \ldots, N$. We set $N = 3$, and the activation functions $S_{(i)1}(x(t))$, $S_{(i)2}(x(t))$, and $S_{(i)3}(x(t))$ are chosen as follows:

$$sig(x) = \frac{1}{1+e^{-x}} + 0.5 \tag{80}$$

$$S_{(1)1}(x(t)) = \begin{bmatrix} sig(x_{NS1})sig(x_{NS2}) \\ sig(x_{EW1}) \\ sig(x_{EW2})sig(x_{NS2}) \\ sig(x_{EW1})sig(x_{EW2}) \end{bmatrix}, \tag{81}$$

$$S_{(1)2}(x(t)) = \begin{bmatrix} sig(x_{NS1}) \\ sig(x_{NS1})sig(x_{EW1}) \\ sig(x_{NS2}) \\ sig(x_{EW2})sig(x_{NS2}) \end{bmatrix}, \tag{82}$$

$$S_{(1)3}(x(t)) = \begin{bmatrix} sig(x_{NS1})sig(x_{EW1}) \\ sig(x_{EW1}) \\ sig(x_{EW2})sig(x_{NS2}) \\ sig(x_{EW2}) \end{bmatrix}, \tag{83}$$

$$S_{(2)1}(x(t)) = \begin{bmatrix} sig(x_{NS1}) \\ sig(x_{EW1}) \\ sig(x_{NS2}) \\ sig(x_{EW2}) \end{bmatrix}, \tag{84}$$

$$S_{(2)2}(x(t)) = \begin{bmatrix} sig(x_{NS1}) \\ sig(x_{EW1}) \\ sig(x_{NS2}) \\ sig(x_{EW2}) \end{bmatrix}, \tag{85}$$

$$S_{(2)3}(x(t)) = \begin{bmatrix} sig(x_{NS1}) \\ sig(x_{EW1}) \\ sig(x_{NS2}) \\ sig(x_{EW2}) \end{bmatrix} \tag{86}$$

$$S_{(3)1}(x(t)) = \begin{bmatrix} sig(x_{NS1})sig(x_{NS2}) \\ sig(x_{EW1})sig(x_{EW2}) \\ sig(x_{NS1})sig(x_{NS2}) \\ sig(x_{NS2})sig(x_{EW2}) \end{bmatrix}, \tag{87}$$

$$S_{(3)2}(x(t)) = \begin{bmatrix} sig(x_{EW1})sig(x_{EW2}) \\ sig(x_{NS1})sig(x_{NS2}) \\ sig(x_{NS2})sig(x_{EW2}) \\ sig(x_{NS1})sig(x_{EW2}) \end{bmatrix}, \tag{88}$$

$$S_{(3)3}(x(t)) = \begin{bmatrix} sig(x_{NS1})sig(x_{NS2}) \\ sig(x_{NS1})sig(x_{NS2}) \\ sig(x_{EW1})sig(x_{NS2}) \\ sig(x_{NS1})sig(x_{EW2}) \end{bmatrix}, \tag{89}$$

where the first NN estimate, denoted as $NN_{(1)}$, shares the same NN structure as the nominal NN structure, which is also used in single NN simulation in Section VI-A2. $NN_{(2)}$ and $NN_{(3)}$ use different activation functions that formulate different neuron connections and build up new NN structures. These multiple models are selected with different NN structures, which are tested to have better performance under different traffic conditions. Moreover, switching between different models will affect the dynamic parameter estimation performance (e.g., convergence speed, overshoot). With the multiple-model design, the best estimation model is chosen based on the current traffic condition and the estimation performance.

B. Simulation Results

In this section, the simulation results are presented. First, the traffic delay optimization results are shown, and the traffic delay comparison between conventional traffic control algorithms is listed. The estimation results are then presented.

1) Optimal Control Result: To test the optimal control result, we compare our control algorithm with conventional traffic signal control schemes, including pre-timed control and actuated control. We choose the weight vector $\mathbb{P} = [s_1, s_{12}, s_2, s_{12}]^T$, which is determined based on the traffic flow on each road. The cycle length for pre-timed signal control is 80s. The minimum green signal length for actuated control is 20s. The testing scenario is shown in the bottom graph in Fig. 4, where we assume the traffic flow is initially normal flow (case 2) but increases to busy flow (case 3) after 100 signal cycles, and finally decrease to sparse flow (case 1) after 200 signal cycles. Fig. 4's top graph shows the average traffic delay using different traffic signal control methods. It indicates that with single NN control and multiple NN control, the traffic delay is relatively less compared to conventional control methods. Moreover, with the use of multiple NNs, the control scheme can quickly adapt to changing traffic scenarios, whereas single NN control will not perform as well under heavy traffic conditions. This result shows the effectiveness of the proposed adaptive control scheme and further reveals the advantage of the multiple NN design structure. We then run this test case for 20 runs to reduce the effect of random traffic flow and collect the average traffic delay data. Table I shows the average traffic delay using different methods. From these data we can see that by using our optimal control method, the average traffic delay is significantly reduced under all the initial green-time options. Compared with a single NN-based optimal control scheme, the scheme with multiple NNs results in a better reduction on average traffic delay.

TABLE I: Travel delays with different initial green times and control methods

Control Method	Avg. Veh. Delay (s)	Std. Dev.
Pre-timed control (init. green 24s)	25.23	11.12
Pre-timed control (init. green 40s)	16.44	5.23
Pre-timed control (init. green 56s)	19.87	5.48
Actuated control	17.58	4.62
Multiple NN control	**9.85**	**1.72**
Single NN control	**14.35**	**7.56**

2) Estimation Result: We test the state estimation of the proposed schemes, in which we choose the initial estimate $\theta(0)$ and $\theta_{(i)(0)}, i = 1, 2, 3$ are $0.8\theta^*$. Fig. 5 shows the estimation error of the average traffic delay increment, and the bottom figure shows the switching signal for the multiple NN design. Fig. 6 shows the designed control signal NN control and multiple NN control. Table II shows the maximum absolute error and the standard error comparison of using single NN and multiple NN estimations. Based on the results above, we can see that multiple NNs generally give a better

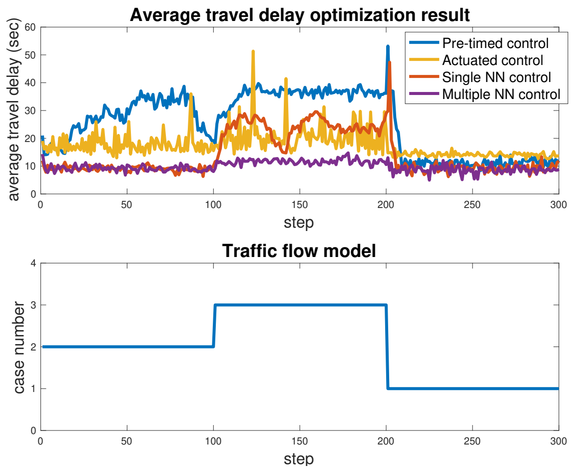


Fig. 4: Optimization results with multiple NNs.

estimation of the original traffic system, especially in changing traffic flow scenarios. As we can see at around 200 steps, where the traffic flow is changing from case 3 to case 1, the estimation using the single NN has some large errors. The proposed multiple NN control can give a better estimation of the unknown and changing dynamics and lead to better control performance.

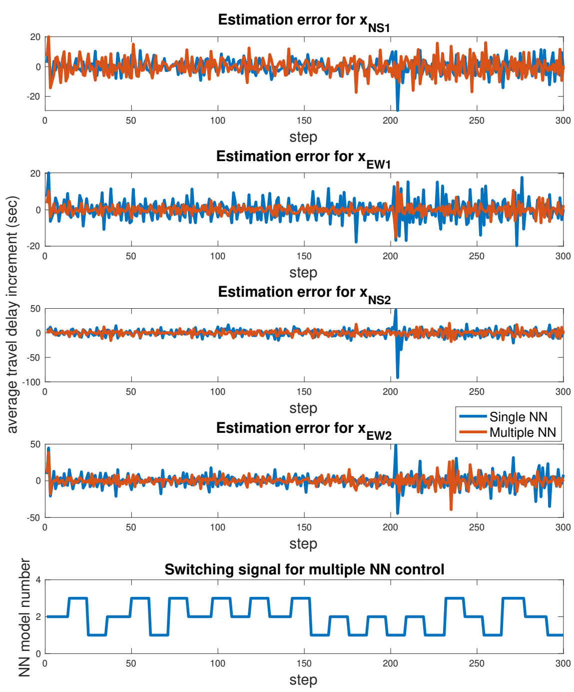


Fig. 5: State estimation error for x_o.

3) Discussion: From this simulation study, the effectiveness of the proposed adaptive online-learning-based optimal control design is verified. Based on the estimation results for the system state x_o, we can see that with both single NN and multiple NN models, a good estimation result can be obtained. However, with the multiple NN models estimation results

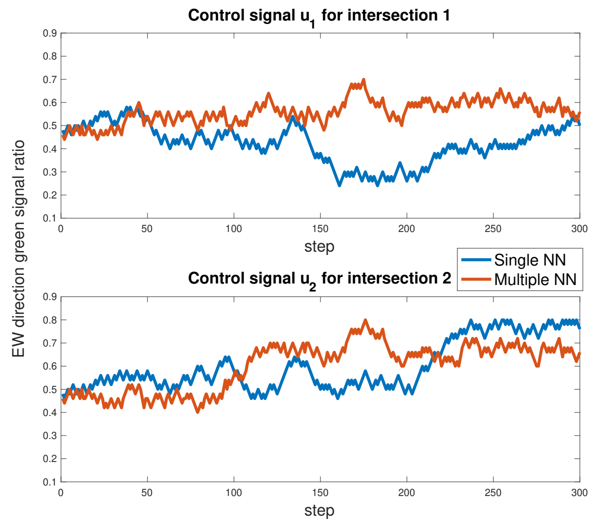


Fig. 6: Control signals.

TABLE II: Estimation error comparison

State Estimation	Single NN (s)		Multiple NNs (s)	
	Max Err.	Std. Err.	Max Err.	Std. Err.
x_{oNS1}	29.51	5.90	20.16	5.58
x_{oEW1}	20.39	5.90	18.15	2.77
x_{oNS2}	91.39	9.16	19.89	5.42
x_{oEW2}	48.79	9.74	38.67	6.97

shown in Fig. 5, an accurate estimation of the original traffic model with smaller estimation error and a greater ability to adapt to traffic flow model changes can be achieved. For the optimization results shown in Fig. 4, we can see that with the proposed optimal control scheme, the average traffic delay is significantly reduced compared with pre-timed or actuated signal control. The proposed algorithm is, therefore, suitable for all traffic flow scenarios and is capable of adapting to traffic flow variations.

Comparison Between Single NN and Multiple NNs. Based on the simulation studies, the proposed multiple NN design can achieve better estimation performance and better optimal control performance than can the single NN design. The advantage of using multiple NNs is that multiple NNs have more candidate NN selections and evaluate and switch between them during the real-time learning and control process, whereas the single NN will have only one candidate NN model. Unlike offline learning schemes in which the estimation performance of different models is judged before system operation, online learning will require all the decisions to be made along with system operation. For this purpose, during system operation, the system dynamic may change, and the preselected NN for a single NN may not be the best estimation of the current system operating condition. Multiple NNs, on the other hand, will provide more candidate estimations and serve as a model evaluation process in the online-learning process.

While the stability and the effectiveness of the control are guaranteed with the design, for best control performance, the number and structure of the NNs in multiple-model NN design are selected based on our experiment experience. This process is similar to the model selection process for offline learning (NN-based control). However, for the design of a multiple-model adaptive controller of the multiple models, each model is chosen to correspond to a system operation that may lead to an essentially different dynamic function described by a unique NN with a determined structure and a set of unknown parameters. Thus, a multiple-model-based adaptive control design has expanded capacities in dealing with the structural and parametric uncertainties of traffic system dynamics.

VII. Conclusions and Future Work

In this paper, a traffic signal control scheme was designed with the use of an adaptive online-learning scheme using multiple-model NNs. First, NN-based traffic system modeling was presented for both single NN and multiple NNs. Then, the adaptive optimal control scheme was designed with an online-learning scheme used to identify the traffic system model. Finally, a simulation study was presented to show the effectiveness of the proposed scheme. Based on the simulation results, we can conclude that the proposed multiple-model-NN-based online estimation scheme can well identify the original traffic system. Moreover, the optimal control scheme designed based on the estimated traffic model can achieve improved traffic-delay minimization compared to pre-timed traffic signal control. The control design proposed in the paper is, therefore, suitable for network traffic signal control. For simulation simplicity and clarity, we conducted the two-intersection simulation study. In future work, we can move the simulation study to larger traffic networks. Another potential future work related to this paper is to test this algorithm in a real-world traffic network. In this case, more traffic phases such as left and right turn at the concerned intersections need to be considered in traffic modeling. The proposed scheme is capable of handling such tests with proper NN structures selected.

Acknowledgment

This paper is based on work supported by the US Department of Energy, Vehicle Technologies Office, Energy Efficient Mobility Systems (EEMS) program, Systems and Modeling for Accelerated Research in Transportation (SMART) Mobility Consortium. This support is gratefully acknowledged.

References

[1] K.-L. A. Yau, J. Qadir, H. L. Khoo, M. H. Ling, and P. Komisarczuk, "A survey on reinforcement learning models and algorithms for traffic signal control," *ACM Computing Surveys (CSUR)*, vol. 50, no. 3, pp. 1–38, 2017.

[2] G. Britain and T. A. Unit, *General Principles of Traffic Control by Light Signals*. Department for Transport London, UK, 2006.

[3] B. Yin, M. Dridi, and A. El Moudni, "Traffic network micro-simulation model and control algorithm based on approximate dynamic programming," *IET Intelligent Transport Systems*, vol. 10, no. 3, pp. 186–196, 2016.

[4] J. Luk, "Two traffic-responsive area traffic control methods: Scat and scoot," *Traffic engineering & control*, vol. 25, no. 1, 1984.

[5] C. G. S. Cools and B. D'Hooghe, "Self-organizing traffic lights: A realistic simulation," in *Advances in applied self-organizing systems*. Springer, 2013, pp. 45–55.

[6] R. P. Roess, E. S. Prassas, and W. R. McShane, *Traffic engineering*. Pearson/Prentice Hall, 2004.

[7] J. D. Little, M. D. Kelson, and N. H. Gartner, "Maxband: A versatile program for setting signals on arteries and triangular networks," 1981.

[8] N. H. Gartner, S. F. Assman, F. Lasaga, and D. L. Hou, "A multi-band approach to arterial traffic signal optimization," *Transportation Research Part B: Methodological*, vol. 25, no. 1, pp. 55–74, 1991.

[9] C. Stamatiadis and N. H. Gartner, "Multiband-96: a program for variable-bandwidth progression optimization of multiarterial traffic networks," *Transportation Research Record*, vol. 1554, no. 1, pp. 9–17, 1996.

[10] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, 2013.

[11] D. Teodorović, "Swarm intelligence systems for transportation engineering: Principles and applications," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 6, pp. 651–667, 2008.

[12] E. Mouhcine, K. Mansouri, and Y. Mohamed, "Solving traffic routing system using vanet strategy combined with a distributed swarm intelligence optimization," *J. Comput. Sci*, vol. 14, pp. 1499–1511, 2018.

[13] B. P. Gokulan and D. Srinivasan, "Distributed geometric fuzzy multiagent urban traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 714–727, 2010.

[14] D. Hartanti, R. N. Aziza, and P. C. Siswipraptini, "Optimization of smart traffic lights to prevent traffic congestion using fuzzy logic," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 17, no. 1, pp. 320–327, 2019.

[15] M. A. Salman, S. Ozdemir, and F. V. Celebi, "Fuzzy traffic control with vehicle-to-everything communication," *Sensors*, vol. 18, no. 2, p. 368, 2018.

[16] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (marlin-atsc): methodology and large-scale application on downtown toronto," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1140–1150, 2013.

[17] L. Kuyer, S. Whiteson, B. Bakker, and N. Vlassis, "Multiagent reinforcement learning for urban traffic control using coordination graphs," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2008, pp. 656–671.

[18] P. Mannion, J. Duggan, and E. Howley, "An experimental review of reinforcement learning algorithms for adaptive traffic signal control," in *Autonomic road transport support systems*. Springer, 2016, pp. 47–66.

[19] D. Liu, W. Yu, S. Baldi, J. Cao, and W. Huang, "A

switching-based adaptive dynamic programming method to optimal traffic signaling," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 11, pp. 4160–4170, 2019.

[20] D. Liu, S. Baldi, W. Yu, and G. Chen, "On distributed implementation of switch-based adaptive dynamic programming," *IEEE Transactions on Cybernetics*, 2020.

[21] S. Amini, I. Gerostathopoulos, and C. Prehofer, "Big data analytics architecture for real-time traffic control," in *2017 5th IEEE international conference on models and technologies for intelligent transportation systems (MT-ITS)*. IEEE, 2017, pp. 710–715.

[22] A. Ata, M. A. Khan, S. Abbas, G. Ahmad, and A. Fatima, "Modelling smart road traffic congestion control system using machine learning techniques," *Neural Network World*, vol. 29, no. 2, pp. 99–110, 2019.

[23] D. Nallaperuma, R. Nawaratne, T. Bandaragoda, A. Adikari, S. Nguyen, T. Kempitiya, D. De Silva, D. Alahakoon, and D. Pothuhera, "Online incremental machine learning platform for big data-driven smart traffic management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4679–4690, 2019.

[24] Q. Yang, S. Jagannathan, and Y. Sun, "Robust integral of neural network and error sign control of mimo nonlinear systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3278–3286, 2015.

[25] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani, "Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks," *IEEE Transactions on neural networks and learning systems*, vol. 24, no. 10, pp. 1513–1525, 2013.

[26] H. Zhang, Z. Wang, and D. Liu, "A comprehensive review of stability analysis of continuous-time recurrent neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 7, pp. 1229–1262, 2014.

[27] Z. Ni, H. He, and J. Wen, "Adaptive learning in tracking control based on the dual critic network design," *IEEE transactions on neural networks and learning systems*, vol. 24, no. 6, pp. 913–928, 2013.

[28] A. Sahoo, H. Xu, and S. Jagannathan, "Adaptive neural network-based event-triggered control of single-input single-output nonlinear discrete-time systems," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 1, pp. 151–164, 2015.

[29] W. Xu, X. Liu, H. Wang, and Y. Zhou, "Event-triggered adaptive nn tracking control for mimo nonlinear discrete-time systems," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[30] W. Meng, Q. Yang, and Y. Sun, "Adaptive neural control of nonlinear mimo systems with time-varying output constraints," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 5, pp. 1074–1085, 2014.

[31] L. Chen and K. S. Narendra, "Intelligent control using neural networks and multiple models," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 2. IEEE, 2002, pp. 1357–1362.

[32] K. S. Narendra, J. Balakrishnan, and M. K. Ciliz, "Adaptation and learning using multiple models, switching, and tuning," *IEEE control systems magazine*, vol. 15, no. 3, pp. 37–51, 1995.

[33] H. Wang, C. R. Wang, M. Zhu, and W. Hong, "Globalized modeling and signal timing control for large-scale networked intersections," in *Proceedings of the 2nd ACM/EIGSCC Symposium on Smart Cities and Communities*, 2019, pp. 1–7.

[34] H. Wang, M. Zhu, W. Hong, C. Wang, G. Tao, and Y. Wang, "Optimizing signal timing control for large urban traffic networks using an adaptive linear quadratic regulator control strategy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 333–343, 2022.

[35] W. Li, C. Wang, Y. Shao, H. Wang, J. Ringler, G. Zhang, T. Ma, and D. Chou, "Hybrid neural network modeling for multiple intersections along signalized arterials-current situation and some new results," in *VEHICULAR 2021, The Tenth International Conference on Advances in Vehicular Systems, Technologies and Applications*, 2021, pp. 10–20.

[36] O. Corrigan, "An investigation into machine learning solutions involving time series across different problem domains," Ph.D. dissertation, Dublin City University, 2018.

[37] J. Hu and J. Wang, "Global stability of complex-valued recurrent neural networks with time-delays," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 6, pp. 853–865, 2012.

[38] J. Sarangapani, *Neural network control of nonlinear discrete-time systems*. CRC press, 2018.

[39] E. B. Kosmatopoulos, M. M. Polycarpou, M. A. Christodoulou, and P. A. Ioannou, "High-order neural network structures for identification of dynamical systems," *IEEE transactions on Neural Networks*, vol. 6, no. 2, pp. 422–431, 1995.

[40] G. Tao, *Adaptive control design and analysis*. John Wiley & Sons, 2003, vol. 37.

[41] D. Liu and Q. Wei, "Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 3, pp. 621–634, 2013.

[42] X. Pang and P. Werbos, "Neural network design for j function approximation in dynamic programming," *arXiv preprint adap-org/9806001*, 1998.

[43] E. D. Karnin, "A simple procedure for pruning back-propagation trained neural networks," *IEEE transactions on neural networks*, vol. 1, no. 2, pp. 239–242, 1990.