

Research Article

Modelling and Prediction of Random Delays in NCSs Using Double-Chain HMMs

Yuan Ge ¹, Yan Zhang,¹ Wengen Gao,¹ Fanyong Cheng,¹ Nuo Yu,¹ and Jincenzi Wu²

¹School of Electrical Engineering, Anhui Polytechnic University, Wuhu 241000, China

²Tandon School of Engineering, New York University, New York, NY, USA

Correspondence should be addressed to Yuan Ge; ygetoby@mail.ustc.edu.cn

Received 31 July 2020; Revised 17 September 2020; Accepted 30 September 2020; Published 29 October 2020

Academic Editor: Licheng Wang

Copyright © 2020 Yuan Ge et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper is concerned with the modelling and prediction of random delays in networked control systems. The stochastic distribution of the random delay in the current sampling period is assumed to be affected by the network state in the current sampling period as well as the random delay in the previous sampling period. Based on this assumption, the double-chain hidden Markov model (DCHMM) is proposed in this paper to model the delays. There are two Markov chains in this model. One is the hidden Markov chain which consists of the network states and the other is the observable Markov chain which consists of the delays. Moreover, the delays are also affected by the hidden network states, which constructs the DCHMM-based delay model. The initialization and optimization problems of the model parameters are solved by using the segmental K-mean clustering algorithm and the expectation maximization algorithm, respectively. Based on the model, the prediction of the controller-to-actuator (CA) delay in the current sampling period is obtained. The prediction can be used to design a controller to compensate the CA delay in the future research. Some comparative experiments are carried out to demonstrate the effectiveness and superiority of the proposed method.

1. Introduction

In traditional control systems, system nodes (such as sensors, controllers, and actuators) are usually connected by the port to port wiring, which may cause many problems such as the difficult wiring and maintenance and the low flexibility and reliability. Such drawbacks appear in many automation systems due to the increasing complexity of controlled plants. With the advent and development of networks, traditional point-to-point control systems are being reshaped and redefined, which gives birth to networked control systems (NCSs) wherein feedback control loops are closed through communication networks [1]. The utilization of a multipurpose shared network to connect spatially distributed components endows control systems with several advantages such as decreased volume of wiring, low installation and maintenance costs, increased system flexibility, and high reliability [2]. Nowadays, NCSs have been extensively applied in many practical systems such as unmanned marine vehicles [3], intelligent manufacture [4],

transportation network [5], and haptics collaboration over the Internet [6].

Compared with traditional control systems, the integration of communication network in the control loops leads to network-induced delays, data dropouts, packet disorder, and congestion. Among them, the network-induced delay is the most significant phenomenon that degrades the system performance or even destabilizes the control systems. See [3] for an example of dealing with random delays in networked unmanned marine vehicles. Therefore, it is very necessary and urgent to investigate the random delays while implementing networked control. This is the initial research motivation of this paper. Over the past decade, considerable attention has been paid to the studying of NCSs with delays, and many significant results have been reported in the literature (see [7, 8] and references therein).

Based on the networked structure for the UMV (Figure 1 in [3]), we can draw a typical structure of NCS as shown in Figure 2. The sensor-to-controller delay (called SC delay) in the k th sampling period (denoted as τ_k^{sc}) is exposed when

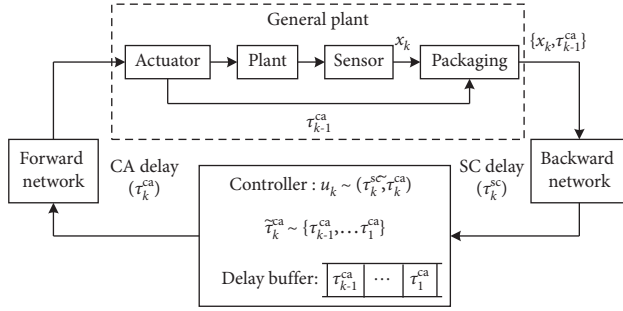


FIGURE 1: Refined diagram of Figure 2 with some other data.

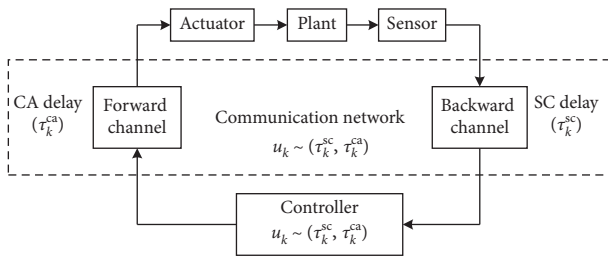


FIGURE 2: Typical structure of NCS.

the sensor measurement is transmitted from the sensor to the controller. Similarly, the controller-to-actuator delay (called CA delay) in the k th sampling period (denoted as τ_k^{ca}) is exposed when the control law is transmitted from the controller to the actuator. Regardless of the type of network (wired or wireless communication network), the performance of NCSs is always affected by these delays, which can be constant or time-varying according to their distribution characteristics. To compensate the delays, it is often necessary to establish the mathematical model of delays first. The last two decades have witnessed the flourishing evolution of this area in NCSs.

The constant delay occurs in NCSs when a buffer is used on the controller (or actuator) side, and the data in the buffer are read periodically by the controller (or actuator) [9]. For this case, the delay is simply modelled as a constant. As a result, the NCSs can be transformed into deterministic systems, and some conventional deterministic control methodologies have been applied to NCSs in [10, 11]. Nevertheless, the buffer makes the delay artificially increased because the data packet can be used only at some fixed time instant although it has already arrived for a while. The main limitation of the constant delay model leads to much design conservatism, or, even worse, the system stability margin decreases so much that the system becomes unstable.

On the other hand, the controller and actuator in NCSs are usually event-driven, which results in the time-varying delay in NCSs. For this case, two divisions can be found in the literature, i.e., the deterministically varying delay and the stochastically varying delay. In the former division, the variation of delays is not known a priori, but the instantaneous value is available to designers in real time. In the latter division, the variation of the delays is associated with some statistical descriptions: either the case of the current

delay being independent of the previous ones or the case of some correlation existing between the current delay and the previous ones.

For the deterministically varying delay, there are three main approaches to analyze the NCSs although the statistics of time-varying delay is not available. The first approach is to transform the network-induced delay into the input delay, and then the recent advances on the delay system approaches can be applied for the analysis and control of NCSs [12, 13]. The second approach is to treat the network-induced delay as a variation parameter of the system, and, more specifically, it is usually called as the exponential uncertainty [14, 15]. Then, the stability and stabilization problems can be solved based on the robust control approach. The third approach is to introduce a new working mode of the actuator, by which the delay can take finite values only, and ultimately the NCS is modelled as a discrete-time switched system with a finite number of subsystems [16, 17]. The main difference between them is that different approaches result in a different system model. The first one models the NCS as a delay system, the second one models the NCS as a parameter-varying system, and the third one models the NCS as a discrete-time switched system.

For the stochastically varying case, the statistics of delays are available for system design, and the stochastic system approach can be applied. To model the random delays, two methods are widely used: one is the independent identically distributed (i.i.d.) model and the other is the Markov chain model. The i.i.d. model is similar to the modelling of stochastic sampling problem, i.e., the delay is partitioned into multiple different time-varying delays and each delay has a certain bound. Then, a delay distribution-based analysis and synthesis approach can be applied for the NCSs with nonuniform distribution characteristics of delays [18, 19]. The same delay distribution-based approach has been extended to fuzzy NCSs in [20, 21]. Different from the i.i.d. model, the Markov chain model is proposed under the assumption that the delay is correlated and the transition of different delays has Markovian property. For the past few years, a lot of effort has been put on the research about Markov chain-based analysis and synthesis methods for random delays in NCSs. These methods can be generally divided into two categories: one considering only SC delays [22] (or CA delays [23]) and the other considering both SC delays and CA delays [24, 25]. In [24, 26], the SC delay and the CA delay were lumped together as a single delay, and then were modelled as a single Markov chain. In [25, 27], the SC delay and the CA delay were modelled as two different Markov chains. Under the Markov chain-based delay model, the NCS is often modelled as a Markovian jump linear system (MJLS), and then many control methodologies (e.g., robust control, predictive control, and fuzzy control) can be used to analyze and synthesize the NCSs with random delays.

The stochastic distribution of current delay is governed by the previous delay in the Markov chain-based delay model. However, some other researchers believe that the stochasticity of current delay should be due in large part to some stochastic network factors (e.g., network load, node competition, and network congestion) which can be described as an abstract random variable named network state. As a result, hidden

Markov models (HMMs) have been successfully used to model random delays of NCSs [28, 29]. In the HMM-based delay model, the stochastic distribution of current delay is only governed by the current network state. This is the main difference between the HMM-based delay model and the Markov chain-based delay model. Along with each packet transmission over the network, the network state will jump from one mode to another following a Markov chain. The network state determines the distribution of random delays. This kind of relationship between the network state and the random delay is referred to as an HMM. According to the delay characteristics, there are mainly three kinds of HMMs to model the random delays of NCSs: discrete-time HMM (DTHMM) [28, 30], continuous-time HMM (CTHMM) [29, 31], and semi-continuous HMM (SCHMM) [32]. Moreover, how to optimally initialize the parameters of HMM-based delay models has been discussed in [33].

Compared with the Markov chain-based delay model which describes the probabilistic relationship among random delays themselves, the HMM-based delay model reveals the essential generation mechanism of random delays and describes the probabilistic relationship between random delays and hidden network states. To a certain extent, the HMM-based delay model is closer to the real distribution of random delays. However, the probabilistic relationship objectively existing among random delays themselves is neglected in the HMM-based delay model. Therefore, to get a more accurate delay model, we prefer to believe that the probabilistic relationship exists not only between random delays and network states, but also between random delays themselves, which is the main motivation of this paper. That is to say, the current delay is governed by the current network state as well as the previous delay. Therefore, the so-called double-chain hidden Markov model (denoted as DCHMM) is introduced in this paper to model the random delays in NCSs. The contribution of this paper lies in the novel modelling method (i.e., DCHMM) that considers the relationship between the random delay and the network state as well as the relationship between the random delay and its previous value. This is the first time in the literature that both the network status and the delay itself are considered simultaneously in the delay modelling. Based on the model, the segmental K-mean clustering algorithm and the expectation maximization algorithm are presented to solve the initialization and optimization problems of the model parameters. Then, the optimized model is used to predict the CA delay and get more accurate delay prediction result than the discrete HMM.

Generally, the CA delay can arbitrarily take values from its acceptable interval, although only a certain discrete value is observed during each sampling period. This is the reason the continuous HMM [29, 31] and semicontinuous HMM [32] have been used to model the CA delays. In these two models, the mixture Gaussian density functions are used to describe the distribution of the CA delays, which improves the modelling accuracy compared with the discrete HMM. However, at the same time, the model parameters to be estimated are augmented and then the computational complexity is increased in these two models. Therefore, for simplicity, the CA delays are limited to a few discrete intervals by using a scalar quantizer, such as the uniform quantizer in [28, 34] and the K-means

clustering quantizer in [30], and then the discrete HMM can be used to model the CA delays. Similarly, in this paper, the CA delays will be also quantized into several finite different discrete observations, and more specifically, the observation process will be assumed to possess the Markovian property, which is much different from the discrete HMM in [28, 30] where the observations are independent of each other. Therefore, there are two Markov chains to be considered in this paper, one is the hidden network states and the other is the observable delay observations, which are modelled as the double-chain hidden Markov model (DCHMM). Since the DCHMM concerns the dependency relationship objectively existing among random CA delays, it will present higher precision of modelling and prediction than the discrete HMM, which will be demonstrated through the contrastive experiments in this paper. Nevertheless, the superiority of DCHMM over the continuous or semicontinuous HMM is not so clear unless some continuous stochastic processes other than the Markov chain are used to describe the dependency relationship among random CA delays, which will be investigated in our future work.

The rest of this paper is organized as follows. The problem formulation is given in Section 2. The DCHMM-based delay model and its parameter estimation are presented in Section 3. How to predict the random CA delays based on the DCHMM delay model is proposed in Section 4. The effectiveness and superiority of the proposed methods are illustrated by experimental examples in Section 5. Finally, some concluding remarks are given in Section 6.

2. Problem Formulation

To better present the problem, the block diagram of a typical NCS in Figure 2 is further refined in Figure 1, where some other data (e.g., sensor measurement, control law, and historical CA delays) are annotated. In Figure 1, the sensor is time-driven and samples the plant every h seconds to get the plant state (called sensor measurement and denoted by x_k). Both the controller and the actuator are event-driven, which means that the controller calculates the control law as soon as the sensor measurement arrives at the controller node and the actuator acts as soon as the control law arrives at the actuator node. If the network nodes (i.e., sensor, controller, and actuator) are clock-synchronous (this can be implemented using the method in [35]), the timestamp technology can be used to calculate the SC and CA delays. The sensor measurement is time-stamped before it is transmitted through the backward network, and when it arrives at the controller node, the SC delay can be calculated by comparing the timestamp of the sensor measurement with the local time of the controller node. Similarly, the control law is time-stamped before it is transmitted through the forward network, and when it arrives at the actuator node, the CA delay can be calculated by comparing the timestamp of the control law with the local time of the actuator node.

Both the SC and CA delays may degrade the system performance or even destabilize the control system. Therefore, many kinds of controllers are designed to compensate these delays. When designing the current control law u_k , the current SC delay τ_k^{SC} has occurred and can be measured by using the timestamp

technology (in this paper, the word “current” means the k th sampling period). However, at the same time, the current CA delay τ_k^{ca} has not occurred. In order to compensate the current CA delay τ_k^{ca} by the control law u_k , a feasible way is to predict the current CA delay before designing the control law u_k . Thus, both the predicted CA delay (denoted as $\tilde{\tau}_k^{\text{ca}}$) and the measured SC delay τ_k^{sc} are known to the controller and can be compensated only if they are considered into the design of the control law u_k . In this paper, the DCHMM is proposed to model the CA delay and obtain its current predicted value $\tilde{\tau}_k^{\text{ca}}$, which aims to improve the prediction accuracy compared with the DTHMM.

There are several parameters (the number of hidden states N , the number of observations M , the probability distribution of the first hidden state π , the transition matrix between hidden states \mathbf{A} , and the set of transition matrices between successive outputs given a particular hidden state \mathbf{B}) in the DCHMM-based CA delay model (denoted as λ , and $\lambda = (N, M, \pi, \mathbf{A}, \mathbf{B})$). To get the optimal estimation of these parameters, the historical CA delays are needed as the inputs of an iterative optimization process to train these parameters. Therefore, a delay buffer is set at the controller node to collect all the past CA delay data (i.e., $\tau_{k-1}^{\text{ca}}, \dots, \tau_1^{\text{ca}}$) as shown in Figure 1. As described above, the adjacent previous CA delay τ_{k-1}^{ca} is calculated at the actuator node and then can be packaged into the current sensor measurement x_k to generate a single packet. This packet will be transmitted to the controller through the backward network. Once receiving this packet, the controller will extract the previous CA delay data τ_{k-1}^{ca} from this packet and put it into the delay buffer. In this way, all the past CA delay data are collected in the delay buffer and then can be used to optimally estimate the parameters of the DCHMM-based CA delay model and predict the current CA delay ($\tilde{\tau}_k^{\text{ca}}$). Based on the predicted value ($\tilde{\tau}_k^{\text{ca}}$) and the real measured value (τ_k^{sc}), some control law can be designed to compensate both the CA delay and the SC delay in the current sampling period. This paper is focused on the modelling and prediction problem, leaving the compensation problem as the future work.

For simplicity, the sum of the SC and CA delay is assumed to be not more than one sampling period (i.e., $\tau_k^{\text{sc}} + \tau_k^{\text{ca}} \leq h$) in this paper, which means that there is no data packet dropout and disorder concerned in this paper.

3. DCHMM-Based Delay Modelling

3.1. Model Derivation. It should be noted that this paper will focus on the discrete-time, discrete-state, and discrete-observation DCHMM, where the state transition probability matrix and the observation transition probability matrix for each state are assumed time homogeneous. Similar to DTHMM, the forward network states are modelled as a time-homogeneous Markov chain with N different states that constitute a discrete finite state set $S = \{1, 2, \dots, N\}$. Generally, the network state (denoted as s) defines the whole working status of the network and is an abstract variable that comprehensively reflects the stochastic factors (e.g., network load, nodes competition, and network congestion) of the network. The network state in the k th sampling period is represented as s_k . Obviously, $s_k \in S(s)$ holds. Along with each packet transmission in the forward network, the network state may

transfer from one mode to another through Markovian property described as the following equation:

$$\begin{aligned} P(s_{k+1} = j | s_k = i, s_{k-1}, \dots, s_1) \\ = P(s_{k+1} = j | s_k = i) = a_{ij}, \quad (i, j \in S, \forall k \in \mathbb{Z}). \end{aligned} \quad (1)$$

The Markovian property tells us that the conditional probability distribution of the network state in the next sampling period (denoted as s_{k+1}) depends only upon the current network state s_k , not on the past network states $\{s_{k-1}, \dots, s_1\}$. In equation (1), $a_{ij} \in [0, 1]$ denotes the one-step transition probability of going from the network state i at time k to the network state j at time $k+1$, satisfying the constraint: $a_{ij} \geq 0$ and $\sum_{j=1}^N a_{ij} = 1$. The Markovian property of network states is illustrated in Figure 3(a). All one-step transition probabilities construct a one-step transition matrix $\mathbf{A} = \{a_{ij}\}_{1 \leq i, j \leq N}$ as shown in Figure 3(b). For the special case of time $k = 1$, a_{ij} is simplified into an initial state probability of the first hidden network state that is defined by π_j as follows:

$$\pi_j = P(s_1 = j). \quad (2)$$

Obviously, $\pi_j \geq 0$ and $\sum_{j=1}^N \pi_j = 1$ hold. All possible initial state probabilities construct an initial vector $\pi = \{\pi_j\}_{1 \leq j \leq N}$.

Now, these parameters (N , π , and \mathbf{A}) define a Markov chain of network states. As is known, the DCHMM describes a two-stage stochastic process. The defined Markov chain is exactly the first stage of the DCHMM-based delay model. In the second stage, for every state at time k additionally a CA delay τ_k^{ca} is generated. As mentioned before, the DCHMM proposed in this paper has finite discrete observations and the observations for each state construct a time-homogeneous Markov chain. Similar to the DTHMM [28, 30], the observations are also obtained by quantizing the CA delays, and then they can be used to estimate the parameters of the DCHMM-based delay model. The delay interval $((0, h])$ is assumed to consist of a complete set such as $(0, h) = (h_0, h_1) \cup (h_1, h_2) \cup \dots \cup (h_{M-2}, h_{M-1}) \cup (h_{M-1}, h_M]$ (where $h_0 = 0$ and $h_M = h$), and for $(\tau_k \in h_{m-1}, h_m]$ ($m = 1, \dots, M$), a new observation (o_k) is defined as $o_k = m$ with an observation space $O = \{1, 2, \dots, M\}$ and $o_k \in O$ holds. After $k-1$ sampling periods, one can get a set of network states: $\bar{s} = \{s_1, s_2, \dots, s_{k-1}\}$, a set of the CA delays: $\bar{\tau}^{\text{ca}} = \{\tau_1^{\text{ca}}, \tau_2^{\text{ca}}, \dots, \tau_{k-1}^{\text{ca}}\}$ and a corresponding set of observations: $\bar{o} = \{o_1, o_2, \dots, o_{k-1}\}$ by quantizing the CA delays, which lay a foundation for deriving the DCHMM-based delay model.

Under the previously mentioned time homogeneous assumption, the conditional probability of obtaining the observation o_k for the DCHMM is defined as follows:

$$\begin{aligned} b_{lm}^{(i)} = P(o_k = m | s_k = i, s_{k-1}, \dots, s_1, o_{k-1} = l, o_{k-2}, \dots, o_1) \\ = P(o_k = m | s_k = i, o_{k-1} = l), \quad (i \in S, l, m \in O, \forall k \in \mathbb{Z}). \end{aligned} \quad (3)$$

In equation (3), $b_{lm}^{(i)} \in [0, 1]$ denotes the conditional transition probability of going from the observation l at time

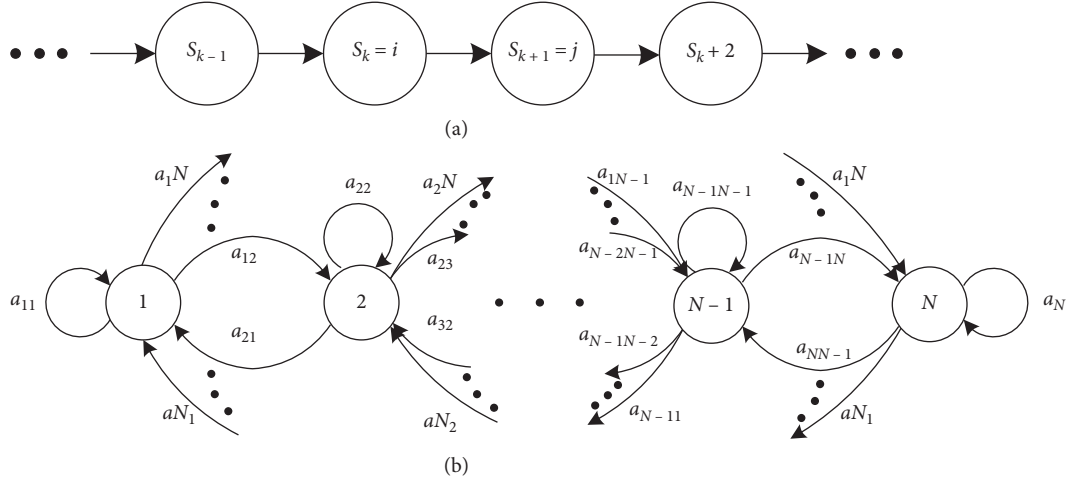


FIGURE 3: Markov chain for the network states: (a) Markovian property; (b) transition matrix.

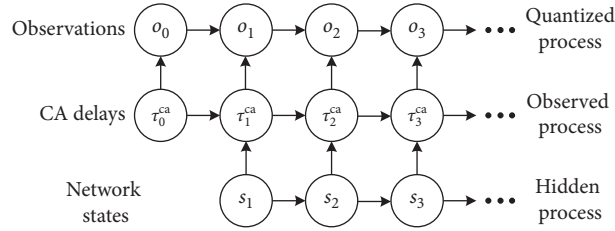


FIGURE 4: DCHMM-based delay model.

$k-1$ to the observation m at time k for the network state i at time k , satisfying the constraint: $b_{lm}^{(i)} \geq 0$ and $\sum_{m=1}^M b_{lm}^{(i)} = 1$. For a given state $i \in S$, let the matrix $\mathbf{B}^{(i)}$ define the conditional observation transition probabilities for the state i , where the (l, m) entry of $\mathbf{B}^{(i)}$ is $b_{lm}^{(i)}$. That is, if the DCHMM is currently in the state i , $\mathbf{B}^{(i)}$ will be the observation transition probability matrix used to determine the observation at the current time point given the observation which was obtained at the previous time point. From this, it can be seen that the observation of the DCHMM-based delay model can be viewed as a time inhomogeneous Markov chain, where the transition probability matrix used for the observations is dependent on the hidden network state of the DCHMM.

Based on the definitions in equations (1), (2), and (3), the DCHMM-based delay model, which is usually denoted as λ , can be completely described as follows:

- (1) A finite set of network states $S = \{1, 2, \dots, N\}$
- (2) A matrix \mathbf{A} of state transition probabilities:

$$\mathbf{A} = \{a_{ij} | a_{ij} = P(s_k = j | s_{k-1} = i)\}, \quad i, j \in S. \quad (4)$$

- (3) A vector π of initial state distribution:

$$\pi = \{\pi_j | \pi_j = P(s_1 = j)\}, \quad j \in S. \quad (5)$$

- (4) A set of the transition matrix between successive observations given a particular state i ($i \in S$):

$$\mathbf{B} = \{\mathbf{B}^{(i)}\} \text{ with } \mathbf{B}^{(i)} = [b_{lm}^{(i)}], \quad l, m \in O. \quad (6)$$

From the above, $\lambda = (N, M, \pi, \mathbf{A}, \mathbf{B})$. In general, N and M are known in advance, so λ can be rewritten as follows for simplicity:

$$\lambda = (\pi, \mathbf{A}, \mathbf{B}). \quad (7)$$

Equation (7) gives the definition of the DCHMM-based delay model which is represented in Figure 4.

Remark 1. The DCHMM is a generalization of both a visible Markov chain and a hidden Markov chain. When there is only one hidden state ($N = 1$), the DCHMM reduces to an homogeneous Markov chain with transition matrix $\mathbf{B}^{(1)}$. On the other hand, when there are $N > 1$ hidden states but each matrix $\mathbf{B}^{(i)}$ ($i \in S$) has identical rows, the model reduces to a DTHMM.

In Figure 4, the observed process is governed by a hidden Markov chain in which successive observations are directly correlated through the Markov property. The DCHMM presented in this paper combines characteristics of both visible and hidden models. It is called “double” since it can be viewed as the superposition of two interlinked Markov chains: the hidden Markov chain governing the relation between states of a nonobservable variable (i.e., network state) and the visible Markov chain governing (together with the hidden state process) the relation between outputs of an

observed variable (i.e., CA delay). That is, the DCHMM has a similar stochastic framework to the HMM, but now it is assumed that, for a given time point k , the current CA delay τ_k^{ca} is not only dependent on the current hidden state s_k but also dependent (through the Markov property) on the previous CA delay τ_{k-1}^{ca} .

In Figure 4, since the value of τ_1^{ca} depends on the past, we consider an initial CA delay value (τ_0^{ca}) at time 0 with no corresponding hidden state, and its corresponding observation is o_0 . Based on this consideration, the set of the past CA delays $\bar{\tau}^{\text{ca}}$ is redefined as $\bar{\tau}^{\text{ca}} = \{\tau_0^{\text{ca}}, \tau_1^{\text{ca}}, \tau_2^{\text{ca}}, \dots, \tau_{k-1}^{\text{ca}}\}$, and the corresponding set of the observations \bar{o} is redefined as $\bar{o} = \{o_0, o_1, o_2, \dots, o_{k-1}\}$.

Remark 2. The dependence of the current CA delay on both the current network state and the previous CA delay can be explained as follows. The CA delay process can be considered as a Markov chain, where the transition probability matrix is dependent on the current network state occupied. That is, the transition probability matrix of the CA delay process is associated with each state in the network state space, and each time the DCHMM enters a new state, the transition probability matrix of the CA delay process for that network state is used to determine which observation (given the previous CA delay) will be emitted for that time point. The CA delay process can thus be viewed as a time inhomogeneous Markov chain, where the transition probability matrix used for the observation process is driven by the hidden state process of the DCHMM.

One of the benefits to introduce the DCHMM-based delay model is that the advantages of both the Markov chain and the HMM are conserved. That is, the system is driven by an unobserved process, and the successive observations are also directly correlated.

3.2. Model Estimation. In order to get the accurate prediction of the current CA delay (τ_k^{ca}) by using the DCHMM-based delay model as described in equation (7), three different estimation problems have to be considered as follows:

- Q1: the estimation of the likelihood of a sequence of observations $\{o_0, o_1, \dots, o_k\}$ given a model λ
- Q2: the estimation of the parameters π , \mathbf{A} and \mathbf{B} given a sequence of observations
- Q3: The estimation of the optimal sequence of hidden network states given a model and a sequence of outputs

These three problems are similar to the problems occurring in DTHMM theory and can be solved using similar methods. Q1 is solved using a forward iterative algorithm. Q2 is achieved with an Expectation Maximization (EM) algorithm. Q3 is obtained through the Viterbi algorithm.

Remark 3. In this section, only the resulting formulas for each algorithm are given. The complete derivation is provided in Appendix A. Moreover, as presented here, the three algorithms can lead to numerical problems since they involve the calculation of infinitesimal values. A good solution is to normalize the intermediary results at each step of the

calculation. The practical implementation of this method is discussed in Appendix B.

3.2.1. Likelihood of the Observation. The likelihood of the observation \bar{o} ($\bar{o} = \{o_0, o_1, \dots, o_{k-1}\}$) given the DCHMM-based delay model λ is defined as follows:

$$L = P(o_0, o_1, \dots, o_{k-1} | \lambda). \quad (8)$$

An iterative procedure similar to the forward procedure developed in [32] for the estimation of the SCHMM is proposed to get the likelihood in equation (8). A forward variable is defined as follows:

$$\alpha_r(j) = P(o_0, \dots, o_r, s_r = j | \lambda). \quad (9)$$

For $r = 1$, equation (9) becomes

$$\alpha_1(j) = b_{o_0 o_1}^{(j)} \pi_j. \quad (10)$$

In the general case, for $r = 2, \dots, k-1$,

$$\alpha_r(j) = b_{o_{r-1} o_r}^{(j)} \sum_{i=1}^N a_{ij} \alpha_{r-1}(i), \quad i, j \in S. \quad (11)$$

By summing $\alpha_{k-1}(j)$ over j , one can obtain the likelihood of the entire sequence of observation \bar{o} :

$$L(o_0, o_1, \dots, o_{k-1}) = \sum_{j=1}^N P(o_0, o_1, \dots, o_{k-1}, s_{k-1} = j) = \sum_{j=1}^N \alpha_{k-1}(j). \quad (12)$$

The iterative computation of $\alpha_r(j)$ is sufficient to obtain the likelihood. Moreover, another iterative algorithm similar to the backward procedure appearing in [32] is defined here. It will be used later for the estimation of the parameters of the DCHMM-based delay model. The backward variable is defined as follows.

$$\beta_r(i) = P(o_{r+1}, \dots, o_{k-1} | o_r, s_r = i, \lambda). \quad (13)$$

For $r = k-1$, we obtain

$$\beta_{k-1}(i) = 1, \quad (14)$$

and for $r = 1, \dots, k-2$,

$$\beta_r(i) = \sum_{j=1}^N a_{ij} b_{o_r o_{r+1}}^{(j)} \beta_{r+1}(j). \quad (15)$$

With this result, the likelihood in equation (12) can be rewritten as

$$L(o_0, o_1, \dots, o_{k-1}) = \sum_{i=1}^N \alpha_r(i) \beta_r(i), \quad r = 1, \dots, k-1. \quad (16)$$

Equation (12) corresponds to $r = k-1$.

3.2.2. *Estimation of π , \mathbf{A} , and \mathbf{B} .* The complete identification of the DCHMM-based delay model requires the estimation of three sets of probabilities: π , \mathbf{A} , and \mathbf{B} . We use an expectation maximization (EM) algorithm for the parameter estimation as we have done in [32].

First, we define the joint probability of two successive hidden network states as follows:

$$\xi_r(i) = P(s_r = i | o_0, \dots, o_{k-1}) = \sum_{j=1}^N \varepsilon_r(i, j) = \frac{\alpha_r(i) \beta_r(i)}{L(o_0, \dots, o_{k-1})}, \quad i \in S, r = 1, \dots, k-2. \quad (18)$$

Using ε_r and ξ_r , we can write the reestimation formula for π , \mathbf{A} , and \mathbf{B} as follows:

$$\hat{\pi}_i = P(s_1 = i | o_0, \dots, o_{k-1}) = \xi_1(i), \quad (19)$$

$$\hat{a}_{ij} = \sum_{r=1}^{k-2} P(s_{r+1} = j | s_r = i, o_0, \dots, o_{k-1}) = \frac{\sum_{r=1}^{k-2} \varepsilon_r(i, j)}{\sum_{r=1}^{k-2} \xi_r(i)}, \quad (20)$$

$$\begin{aligned} \hat{b}_{lm}^{(i)}(r) &= P(o_r = m | o_0, \dots, o_{r-1} = l, \dots, o_{k-1}, s_r = i) \\ &= P(o_r = m | o_{r-1} = l, s_r = i) = \frac{\sum_{r=1}^{k-2} \xi_r(i)}{\sum_{r=1}^{k-2} \xi_r(i)}, \\ &\quad o_{r-1} = l, \end{aligned} \quad (21)$$

In practice, the parameter estimation of the DCHMM-based delay model is achieved using iteratively the forward-backward procedure and the reestimation formulas in (19)~(21). Generally, after dozens of iteration, one can obtain the optimum model parameters λ^* ($\lambda^* = (\pi^*, \mathbf{A}^*, \mathbf{B}^*)$, where $\pi^* = \{\pi_i^*\}$, $\mathbf{A}^* = \{a_{ij}^*\}$, and $\mathbf{B}^* = \{b_{lm}^{(i)*}\}$).

3.2.3. *Optimal Sequence of Hidden States.* Once we have an estimation of the DCHMM-based delay model, we can search the optimal sequence of hidden network states which maximizes the following conditional probability:

$$P(s_1, \dots, s_{k-1} | o_0, o_1, \dots, o_{k-1}), \quad (22)$$

or the joint probability

$$P(s_1, \dots, s_{k-1}, o_0, o_1, \dots, o_{k-1}). \quad (23)$$

This problem can be solved through an iterative dynamic procedure called the Viterbi algorithm. For $r = 1$ and $j = 1, \dots, N$, we define

$$\delta_1(j) = P(o_0, o_1, s_1 = j) = \pi_j b_{o_0 o_1}^{(j)}, \quad (24)$$

and for $r = 2, \dots, k-1$,

$$\begin{aligned} \varepsilon_r(i, j) &= P(s_r = i, s_{r+1} = j | o_0, \dots, o_{k-1}) = \frac{\alpha_r(i) a_{ij} b_{o_r o_{r+1}}^{(j)} \beta_{r+1}(j)}{L(o_0, \dots, o_{k-1})}, \\ &\quad i, j \in S, r = 1, \dots, k-2. \end{aligned} \quad (17)$$

Then, we define the marginal distribution of the hidden network states as follows:

$$\begin{aligned} \delta_r(j) &= \max_{i_1, \dots, i_{r-1}} P(o_0, \dots, o_r, s_1 = i_1, \dots, s_{r-1} = i_{r-1}, s_r = j) \\ &= \left[\max_{i_1, \dots, i_{r-1}} \delta_{r-1}(i_{r-1}) a_{i_{r-1} j} \right] b_{o_{r-1} o_r}^{(j)}. \end{aligned} \quad (25)$$

The optimal network state at time $k-1$ is then predicted as

$$\tilde{s}_{k-1} = \arg \max_{j=1, \dots, N} \delta_{k-1}(j), \quad (26)$$

and we obtain recursively for $r = k-2, \dots, 1$:

$$\tilde{s}_r = \arg \max_{j=1, \dots, N} \delta_{r+1}(j) a_{\tilde{s}_r j}. \quad (27)$$

Finally, the joint probability of the sequence of hidden network states and the sequence of observed CA delay outputs is equal to

$$P(s_1, \dots, s_{k-1}, o_0, o_1, \dots, o_{k-1}) = \max_{j=1, \dots, N} \delta_{k-1}(j). \quad (28)$$

As a dynamic programming method, every decision in the computation of the $\delta_r(j)$ is only locally optimal in the Viterbi algorithm. The globally optimum of the state sequence \tilde{s} is only known after the evaluation of the $\delta_{k-1}(j)$, i.e., after the CA delay sequence has been considered in its entire length. The state \tilde{s}_{k-1} maximizing $\delta_{k-1}(j)$ denotes the end of the optimal state sequence. The predecessor states can be determined by means of backtracking in (27).

4. Delay Prediction

Based on the optimal prediction of the network state \tilde{s}_{k-1} at time $k-1$ and the optimal estimation of the parameters \mathbf{A} and \mathbf{B} , the method to predict the current CA delay ($\tilde{\tau}_k^{\text{ca}}$) is proposed as follows.

Step 1. Predict the optimal network state in the current (i.e., k th) sampling period, which is denoted as \tilde{s}_k :

$$\tilde{s}_k = \arg \max_{j \in S} a_{\tilde{s}_{k-1} j}^*. \quad (29)$$

Step 2. Predict the optimal observation of the current CA delay, which is denoted as \tilde{o}_k :

$$\bar{o}_k = \arg \max_{m \in O} \left(b_{o_{k-1}m}^{(\bar{s}_k)^*} \right). \quad (30)$$

Step 3. Predict the optimal CA delay in the current (i.e., k th) sampling period, which is denoted as $\tilde{\tau}_k^{ca}$:

$$\tilde{\tau}_k^{ca} = \frac{1}{2} \left(h_{o_k}^- - h_{o_{k-1}}^- \right). \quad (31)$$

Remark 4. Once we get the optimal network state in the previous sampling period (\bar{s}_{k-1}) by using (26), the probability of different network state j ($j \in S$) in the current sampling period can be calculated based on the Markovian property of network states in (1), and the state that maximizes the probability is the optimal prediction of the current network state (\bar{s}_k). Based on the prediction (\bar{s}_k) and the previous observation of CA delay (o_{k-1}), the probability of different delay observation m ($m \in O$) in the current sampling period can be calculated according to (3), and the observation that maximizes the probability is the optimal prediction of the current CA delay observation (\bar{o}_k). For the predicted observation (\bar{o}_k), how to calculate the prediction of the current CA delay depends on the delay quantization method. As discussed in [30], there are two kinds of delay quantization methods to obtain the observations of historical CA delays, namely, uniform quantization and K-means clustering quantization. For the uniform quantization, the prediction of the current CA delay ($\tilde{\tau}_k^{ca}$) is set to the midpoint of the subinterval which the current observation (\bar{o}_k) falls in. Based on the definition of the complete subintervals, the subinterval which the observation \bar{o}_k falls in is $(h_{o_{k-1}}^-, h_{o_k}^-]$, and then the prediction of the current CA delay is given in (31). For the K-means clustering quantization, the prediction of the current CA delay ($\tilde{\tau}_k^{ca}$) is set to the centroid of the cluster which the current observation (\bar{o}_k) falls in, and the corresponding calculating method is given as following step 3.

Step 4. Predict the optimal CA delay in the current (i.e., k th) sampling period, which is denoted as $\tilde{\tau}_k^{ca}$:

$$\tilde{\tau}_k^{ca} = c_{o_k}^-. \quad (32)$$

In (32), $c_{o_k}^-$ is the centroid of the cluster which the current observation (\bar{o}_k) falls in. The detailed K-means clustering quantization method and the definition of the cluster centroid have been given in [30]. Actually, we have proved that the K-means clustering quantization is better than the uniform quantization in improving the prediction accuracy of CA delays. Therefore, we will only use equation (32) to predict the current CA delay in the experiments of this paper.

5. Illustrative Example

Some comparative experiments are carried out in this section to demonstrate the effectiveness and superiority of the methods proposed in this paper. For the sake of comparison, the experimental context is designed as the same as that in

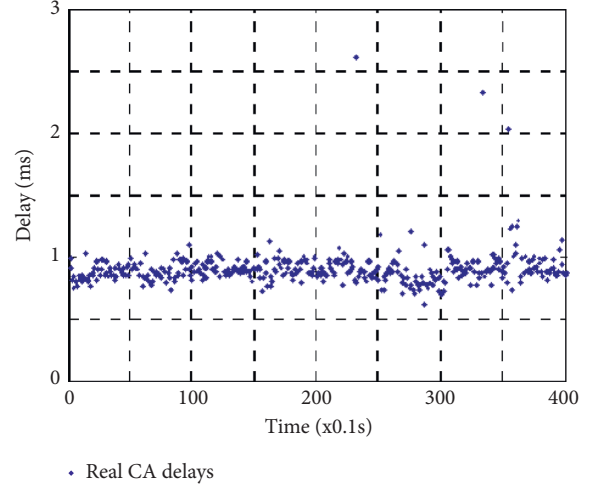


FIGURE 5: Real CA delay values.

[32], and the CA delay values shown in Figure 4 of [32] are used again here to derive the DCHMM-based delay model and validate the prediction of CA delays. For the convenience of reading, Figure 4 of [32] is redrawn as follows. In Figure 5, the first 200 CA delay values (from 1 to 200 on the x -coordinate) are used for the parameter estimation of the DCHMM-based delay model, and the last 200 CA delay values (from 201 to 400 on the x -coordinate) are used to evaluate the predictive effect based on the model. The main difference between the experiments in this paper and that in [32] is that this paper not only considers the relationship between the CA delays and the network states but also considers the relationship between the adjacent CA delays. That is to say, the distribution of the current CA delay in this paper is not only dependent on the current network states but also related to the previous CA delay.

Firstly, we need to initialize the DCHMM-based delay model λ^0 ($\lambda^0 = (N, M, \pi^0, \mathbf{A}^0, \mathbf{B}^0)$). By using the optimal stabilization methods in [33], we can obtain that the optimal initial values of N and M are 4 and 6, respectively, which means that there are four different network states (i.e., $S = \{1, 2, 3, 4\}$), and the delay interval $((0, h])$ can be divided into six complete subinterval (i.e., $M = 6$, and $O = \{1, 2, 3, 4, 5, 6\}$). The vector of initial state distribution (π^0) and the matrix of state transition probabilities (\mathbf{A}^0) can be almost uniformly initialized as follows, since they have little effect on the DCHMM-based delay model:

$$\begin{aligned} \pi^0 &= [0.25 \ 0.25 \ 0.25 \ 0.25], \\ \mathbf{A}^0 &= \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}. \end{aligned} \quad (33)$$

Compared with π^0 and \mathbf{A}^0 , the initial value of the observation matrix (\mathbf{B}^0) is very crucial to the parameter

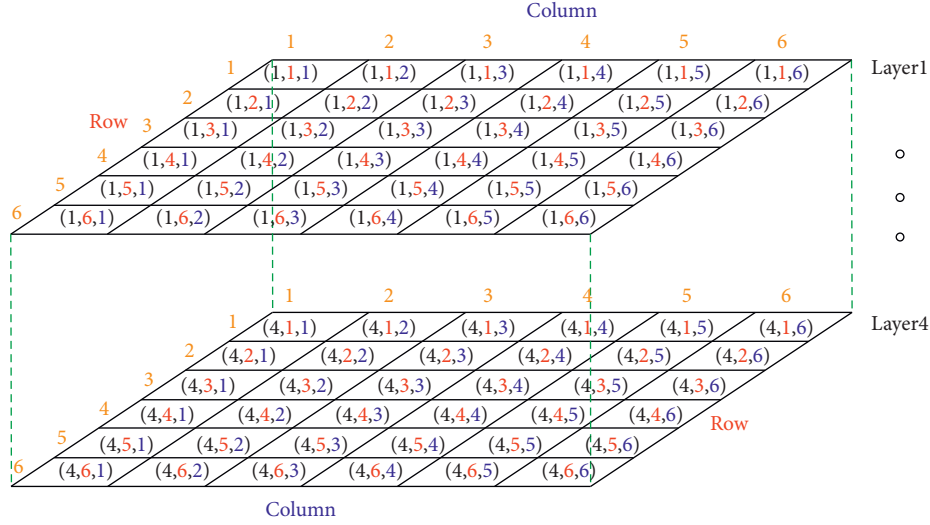


FIGURE 6: The detailed structure of the observation matrix.

estimation of the DCHMM-based delay model. The uniform initialization method is no longer suitable for obtaining the optimal initial value of the observation matrix (\mathbf{B}^0). In this section, the segmental K-means clustering algorithm proposed in [33] is used to optimally initialize the observation matrix (\mathbf{B}^0) as follows based on the first 200 CA delay values in Figure 5. With $N = 4$ and $M = 6$, we can rewrite the entry of the observation matrix \mathbf{B} as follows according to equation (3):

$$i \in S = \{1, 2, 3, 4\} \text{ with } l \in S = \{1, 2, 3, 4\}, l, m \in O = \{1, 2, 3, 4, 5, 6\}. \quad (34)$$

Obviously, the matrix \mathbf{B} is a three-dimensional matrix, in which i denotes the layer of the matrix, l denotes the row of each layer, and m denotes the column of each layer. The detailed structure of the matrix \mathbf{B} is given in Figure 6. There are 4 layers in the matrix \mathbf{B} , and every layer has 36 elements, which makes the matrix \mathbf{B} have a total of 144 elements. In Figure 6, the black number corresponds to the number of layers, the red number to the number of rows, and the blue number to the number of columns. For simplicity, the matrix elements of the second and third layers are omitted here since they are easy to figure out.

Based on the definition in equation (3), all the elements of the matrix \mathbf{B} satisfy the following constraint relation. Each element is greater than or equal to 0 and less than or equal to 1, and the sum of each row in each layer is equal to 1:

$$0 \leq b_{lm}^{(i)} \leq 1, \quad \sum_{m=1}^6 b_{lm}^{(i)} = 1. \quad (35)$$

Based on the first 200 CA delay values in Figure 5, the optimal initial values (\mathbf{B}^0) of the observation matrix can be obtained as shown in Figure 7 by using the segmental K-means clustering algorithm proposed in [33].

In general, the network status is good and the large numeric elements have relatively small row and column numbers in Figure 7. In order to illustrate this point, we

transform layer 1 in Figure 7 into Figure 8 that is a matrix plot with M rows and M columns. In the same way, the other three layers (layer 2 to 4) in Figure 7 can be transformed into their corresponding matrix plots, and the transformed result is omitted here for simplicity. In the matrix plot of Figure 8, each element is a colored dial. The color value (referring to the right color bar) and the size of the colored sector in one rectangle block (i, j) ($1 \leq i, j \leq M$) are determined by the initial value corresponding to the same (i, j) entry of layer 1 in Figure 7. In Figure 8, the region on the left shows the relatively large color values as well as the size of the colored sectors, while the region on the right shows the relatively small color values as well as the size of the colored sectors. Figure 8 illustrates, from another perspective, that the network always shifts to the state with short delay and presents a good status.

Secondly, we can use the EM algorithm to train the DCHMM-based delay model based on the initialized model λ^0 and the first 200 delay values in Figure 5. During this experiment, the maximum iterative number of the training procedure is set to 40 and the threshold ε to terminate the procedure is set to 5×10^{-4} . Under these conditions, it is found that the training procedure can always converge after dozens of iterations. At the end of this procedure, the optimized parameters $(\pi^*, \mathbf{A}^*, \mathbf{B}^*)$ of the DCHMM-based delay model are obtained as follows. Considering that the matrix \mathbf{B}^* contains too many elements (144 in total), which makes it unrepresentable, we present it in the form of a one-dimensional curve as shown in Figure 9 where every 36 coordinates on the horizontal axis corresponds to one layer of \mathbf{B}^* :

$$\pi^* = [0.1377 \quad 0.3956 \quad 0.3012 \quad 0.1655],$$

$$\mathbf{A}^* = \begin{bmatrix} 0.6308 & 0.1317 & 0.1292 & 0.1083 \\ 0.2116 & 0.5719 & 0.1125 & 0.1040 \\ 0.1125 & 0.1726 & 0.5122 & 0.2027 \\ 0.1047 & 0.2108 & 0.2014 & 0.4831 \end{bmatrix}. \quad (36)$$

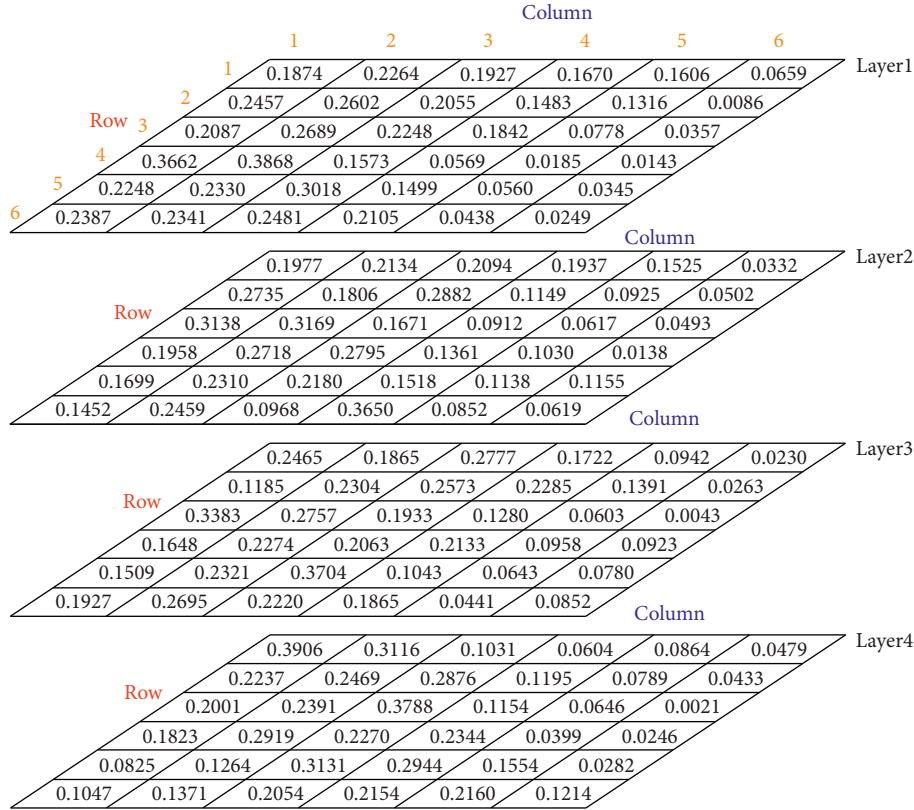


FIGURE 7: The optimal initial values of the observation matrix.

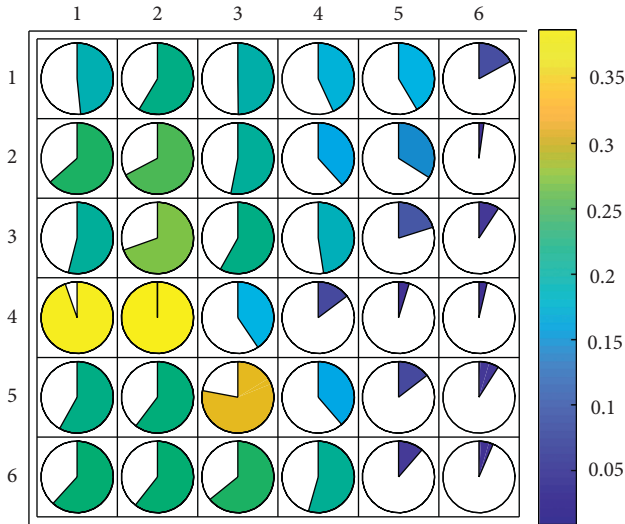


FIGURE 8: Distribution of the initial values on layer 1 of \mathbf{B}^0 .

Thirdly, we can use the Viterbi algorithm to get the optimal previous network state and then predict the current CA delay through three steps. The first step uses equation (29) to estimate the optimal current network state. The second step uses equation (30) to estimate the optimal observation of the current CA delay. The third step uses equation (32) to predict the current CA delay. Taking the three steps, the predicted values of CA delays can be obtained. For example, during the 201st period, the predicted

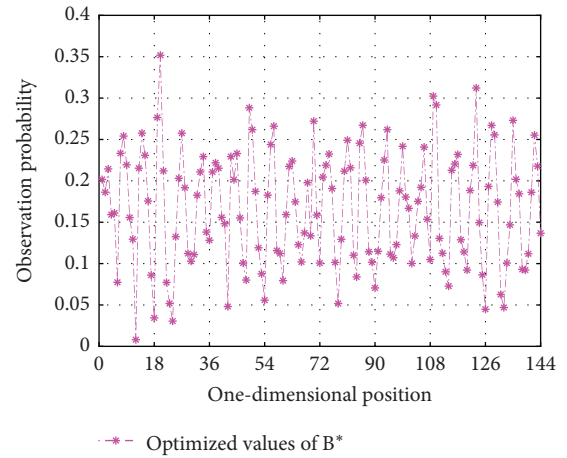


FIGURE 9: One-dimensional distribution of the optimized values of \mathbf{B}^* .

value is 0.8591 ms that is close to the real value 0.8432 ms in Figure 5, and the relative error is 1.89%. By recursively executing the three steps in the following periods, we can get other predicted CA delay values from the 202nd to the 400th period as shown in Figure 10.

To judge the predictive precision of the DCHMM-based delay model, the mean-squared error (MSE) is defined as follows in this experiment:

$$MSE = \frac{1}{200} \sum_{r=201}^{400} (\tilde{\tau}_r^{ca} - \tau_r^{ca})^2. \quad (37)$$

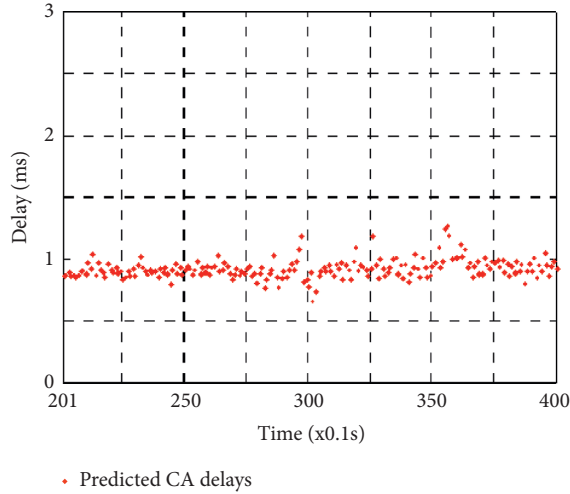


FIGURE 10: Predicted CA delay values.

According to the predicted values in Figure 10 and the real values in Figure 5, we can calculate that the MSE is 0.0032 by using equation (37), which is denoted as MSE_{DCHMM} (i.e., $MSE_{DCHMM} = 0.0032$). Generally, such prediction accuracy has been very high and is acceptable in NCS.

To illustrate the superiority of the DCHMM-based delay model over the DTHMM-based delay model in the prediction accuracy, we carry out again the experiment under the DTHMM-based delay model in [33]. Without considering the dependence between adjacent CA delays, the observation matrix \mathbf{B}^0 degenerates into a two-dimensional matrix, in which the rows represent the network states and the columns represent the intervals of CA delays. Under the same experiment condition, the predicted value is 0.8706 ms during the 201st period for the DTHMM-based delay model, and the relative error is 3.25% ($>1.89\%$ for the DCHMM-based delay model). Therefore, the DCHMM-based delay model proposed in this paper gives a better prediction of the CA delay during the 201st period than the DTHMM-based delay model. Moreover, by the end of this experiment, we got that the MSE for the DTHMM-based delay model is 0.0043 ($>0.0032 = MSE_{DCHMM}$), which is denoted as MSE_{DTHMM} (i.e., $MSE_{DTHMM} = 0.0043$). Obviously, we have $MSE_{DTHMM} > MSE_{DCHMM}$, which demonstrates the superiority of the DCHMM-based delay model over the DTHMM-based delay model in the prediction of CA delays.

Besides, we try to compare the DCHMM-based delay model over the SCHMM-based delay model proposed in [32] and get some interesting results. In the experiment of [33], the predicted value is 0.8615 ms during the 201st period for the SCHMM-based delay model, and the relative error is 2.17% ($>1.89\%$ for the DCHMM-based delay model). However, we cannot simply assume that the DCHMM-based delay model is superior to the SCHMM-based delay model. Actually, by the end of the experiment, the MSE for the SCHMM-based delay model is 0.0029

($<0.0032 = MSE_{DCHMM}$), which is denoted here as MSE_{SCHMM} (i.e., $MSE_{SCHMM} = 0.0029$). It is easy to find that $MSE_{SCHMM} < MSE_{DCHMM} < MSE_{DTHMM}$. So, from this point of view, the DCHMM-based delay model is not superior to the SCHMM-based delay model in the prediction of CA delays. This is mainly because that the delays are treated as discrete values and quantized into some limited subintervals in the DCHMM-based delay model. If the delays are treated as continuous values and can take any value in its allowable interval or if the dependency relationship among random CA delays is taken into account in the SCHMM-based delay model, the derived DCHMM-based delay model will give more accurate prediction and show the superiority over the SCHMM-based delay model, which will be demonstrated in our future work.

6. Conclusions

In networked control systems, random delays are the main cause that degrades the system performance and even causes the system instability. A feasible method to compensate the random delays is to take the random delays into the design of the controller. Considering the CA delay has not occurred when the controller is designed, we need to predict the CA delay before designing the controller. To get a high-precision prediction value, it is necessary to establish a high-precision delay model first. Different from the existing modelling methods, this paper considers the dependency between the network states and the CA delays as well as the interdependence between the CA delays. The dependency between the network states and the CA delays is modelled as a hidden Markov model, and the interdependence between the CA delays is modelled as a Markov chain. As a result, the double-chain hidden Markov model (DCHMM) is proposed in this paper to model the random CA delays for the first time in the networked control systems. As the name implies, there are two Markov chains in this model. One is the hidden Markov chain which consists of the network states, and the other is the observable Markov chain which consists of the random CA delays. Moreover, the random CA delays are also affected by the hidden network states, which constructs the DCHMM-based delay model. The DCHMM-based delay model is highly consistent with the distribution of the real network delays. Therefore, the DCHMM, compared with the DTHMM, gets much closer to the real network, which renders the DCHMM-based delay model with higher accuracy than the DTHMM-based delay model. However, it is worth noting that the DCHMM is more complex than the DTHMM in model parameter estimation and will consume more computing time. As a result, the DCHMM has a relatively poor real-time performance when directly applied to actual systems. A feasible solution is to determine the parameter reestimation frequency according to the prediction accuracy. A threshold is defined as in [32], and the event-triggering mechanism proposed in [36, 37] can be

used to trigger the parameter reestimation when the prediction error exceeds the threshold.

The expectation maximization algorithm is presented to obtain the optimal estimation of the model parameters after they are initialized by using the K-mean clustering algorithm. The Viterbi algorithm is presented to obtain the optimal estimation of the network states, and then the prediction of the CA delay in the current sampling period is obtained. Finally, some comparative experiments are carried out to demonstrate the superiority of the DCHMM-based delay model over the DTHMM-based delay model. Nevertheless, the random CA delays considered in this paper are discrete, and the DCHMM-based delay model is derived through quantizing the delays. If the CA delays are treated as continuous values and can take any value in its allowable interval or if the dependency relationship among random CA delays is taken into account in the SCHMM-based delay model, the derived DCHMM-based delay model will give more accurate

prediction and show the superiority over the SCHMM-based delay model. In our future research, the predicted CA delay will be used to design the controller in the current sampling period to compensate the imminent real CA delay. Furthermore, we will investigate the adaption of these methods to other systems such as the cyber-physical systems and the multiagent systems.

Appendix

A. Derivation of the Algorithm

In this appendix, we provide the complete derivation of the algorithms of Section 3.

A.1. Likelihood of the Observed CA Delay Sequence. For $r = 1$, equation (9) becomes

$$\alpha_1(j) = P(o_0, o_1, s_1 = j) = P(o_1 | o_0, s_1 = j)P(o_0, s_1 = j) = b_{o_0 o_1}^{(j)} P(o_0, s_1 = j). \quad (\text{A.1})$$

Since o_0 and s_1 are independent and the value of o_0 is known, we have

$$\alpha_1(j) = b_{o_0 o_1}^{(j)} P(o_0)P(s_1 = j) = b_{o_0 o_1}^{(j)} P(s_1 = j) = b_{o_0 o_1}^{(j)} \pi_j. \quad (\text{A.2})$$

For $r > 1$, equation (11) becomes

$$\begin{aligned} \alpha_r(j) &= P(o_0, \dots, o_r, s_r = j) \\ &= P(o_r | o_0, \dots, o_{r-1}, s_r = j)P(o_0, \dots, o_{r-1}, s_r = j) \\ &= P(o_r | o_{r-1}, s_r = j) \sum_{i=1}^N P(o_0, \dots, o_{r-1}, s_{r-1} = i, s_r = j) \\ &= b_{o_{r-1} o_r}^{(j)} \sum_{i=1}^N P(s_r = j | o_0, \dots, o_{r-1}, s_{r-1} = i)P(o_0, \dots, o_{r-1}, s_{r-1} = i) \\ &= b_{o_{r-1} o_r}^{(j)} \sum_{i=1}^N P(s_r = j | s_{r-1} = i)P(o_0, \dots, o_{r-1}, s_{r-1} = i) \\ &= b_{o_{r-1} o_r}^{(j)} \sum_{i=1}^N a_{ij} \alpha_{r-1}(i). \end{aligned} \quad (\text{A.3})$$

For β_r , $r = 1, \dots, k-2$, we have

$$\begin{aligned}
\beta_r(i) &= P(o_{r+1}, \dots, o_{k-1} | o_r, s_r = i) \\
&= \frac{P(o_r, o_{r+1}, \dots, o_{k-1}, s_r = i)}{P(o_r, s_r = i)} \\
&= \frac{1}{P(o_r, s_r = i)} \sum_{j=1}^N P(o_r, o_{r+1}, \dots, o_{k-1}, s_r = i, s_{r+1} = j) \\
&= \frac{1}{P(o_r, s_r = i)} \sum_{j=1}^N P(o_r, s_r = i) P(s_{r+1} = j | o_r, s_r = i) P(o_{r+1} | o_r, s_r = i, s_{r+1} = j) \cdot P(o_{r+2}, \dots, o_{k-1} | o_r, o_{r+1}, s_r = i, s_{r+1} = j) \\
&= \sum_{j=1}^N P(s_{r+1} = j | s_r = i) P(o_{r+1} | o_r, s_{r+1} = j) P(o_{r+2}, \dots, o_{k-1} | o_r, o_{r+1}, s_r = i, s_{r+1} = j) \\
&= \sum_{j=1}^N a_{ij} b_{o_r o_{r+1}}^{(j)} \beta_{r+1}(j).
\end{aligned} \tag{A.4}$$

A.2. Estimation of π , **A**, and **B**. Equation (17) for the calculation of $\varepsilon_r(i, j)$ is

$$\begin{aligned}
\varepsilon_r(i, j) &= P(s_r = i, s_{r+1} = j | o_0, \dots, o_{k-1}) \\
&= \frac{1}{P(o_0, \dots, o_{k-1})} P(o_0, \dots, o_{k-1}, s_r = i, s_{r+1} = j) \\
&= \frac{1}{P(o_0, \dots, o_{k-1})} P(o_0, \dots, o_r, s_r = i) P(s_{r+1} = j | o_0, \dots, o_r, s_r = i) \\
&\quad \cdot P(o_{r+1} | o_0, \dots, o_r, s_r = i, s_{r+1} = j) P(o_{r+2}, \dots, o_{k-1} | o_0, \dots, o_{r+1}, s_r = i, s_{r+1} = j) \\
&= \frac{1}{P(o_0, \dots, o_{k-1})} P(o_0, \dots, o_r, s_r = i) P(s_{r+1} = j | s_r = i) \\
&\quad \cdot P(o_{r+1} | o_r, s_{r+1} = j) P(o_{r+2}, \dots, o_{k-1} | o_{r+1}, s_{r+1} = j) \\
&= \frac{\alpha_r(i) a_{ij} b_{o_r o_{r+1}}^{(j)} \beta_{r+1}(j)}{L(o_0, \dots, o_{k-1})}.
\end{aligned} \tag{A.5}$$

and, for $\xi_r(i)$, we have

$$\begin{aligned}
\xi_r(i) &= P(s_r = i | o_0, \dots, o_{k-1}) \\
&= \frac{P(o_0, \dots, o_{k-1}, s_r = i)}{P(o_0, \dots, o_{k-1})} \\
&= \frac{P(o_0, \dots, o_r, s_r = i) P(o_{r+1}, \dots, o_{k-1} | o_0, \dots, o_r, s_r = i)}{P(o_0, \dots, o_{k-1})} \\
&= \frac{P(o_0, \dots, o_r, s_r = i) P(o_{r+1}, \dots, o_{k-1} | o_r, s_r = i)}{P(o_0, \dots, o_{k-1})} \\
&= \frac{\alpha_r(i) \beta_r(i)}{L(o_0, \dots, o_{k-1})}.
\end{aligned} \tag{A.6}$$

$$\begin{aligned}
\hat{a}_{ij} &= \sum_{r=1}^{k-2} P(s_{r+1} = j | s_r = i, o_0, \dots, o_{k-1}) \\
&= \frac{\sum_{r=1}^{k-2} P(s_r = i, s_{r+1} = j, o_0, \dots, o_{k-1})}{\sum_{r=1}^{k-2} P(s_r = i, o_0, \dots, o_{k-1})} \\
&= \frac{\sum_{r=1}^{k-2} P(s_r = i, s_{r+1} = j | o_0, \dots, o_{k-1}) P(o_0, \dots, o_{k-1})}{\sum_{r=1}^{k-2} P(s_r = i | o_0, \dots, o_{k-1}) P(o_0, \dots, o_{k-1})} \\
&= \frac{\sum_{r=1}^{k-2} P(s_r = i, s_{r+1} = j | o_0, \dots, o_{k-1})}{\sum_{r=1}^{k-2} P(s_r = i | o_0, \dots, o_{k-1})} \\
&= \frac{\sum_{r=1}^{k-2} \varepsilon_r(i, j)}{\sum_{r=1}^{k-2} \xi_r(i)},
\end{aligned}$$

Then, we can get the reestimation formula:

$$\begin{aligned}
\widehat{b}_{lm}^{(i)}(r) &= P(o_r = m | o_0, \dots, o_{r-2}, o_{r-1} = l, o_{r+1}, \dots, o_{k-1}, s_r = i) \\
&= \frac{P(o_0, \dots, o_{r-2}, o_{r-1} = l, o_r = m, o_{r+1}, \dots, o_{k-1}, s_r = i)}{P(o_0, \dots, o_{r-2}, o_{r-1} = l, o_{r+1}, \dots, o_{k-1}, s_r = i)} \\
&\quad \frac{\sum_{r=1}^{k-2} \xi_r(i)}{\sum_{r=1}^{k-2} \xi_r(i)} \\
&= \frac{o_{r-1} = l, o_r = m}{\sum_{r=1}^{k-2} \xi_r(i)} \\
&\quad o_{r-1} = l,
\end{aligned} \tag{A.7}$$

A.3. *Optimal sequence of hidden network states.* Equation (24) can be obtained as

$$\begin{aligned}
\delta_1(j) &= P(o_0, o_1, s_1 = j) \\
&= P(o_0, s_1 = j)P(o_1 | o_0, s_1 = j) \\
&= P(s_1 = j)P(o_1 | o_0, s_1 = j) \\
&= \pi_j b_{o_0 o_1}^{(j)},
\end{aligned} \tag{A.8}$$

and, for $r = 2, \dots, k-1$ and $j = 1, \dots, N$,

$$\begin{aligned}
\delta_r(j) &= \max_{i_1, \dots, i_{r-1}} P(o_0, \dots, o_r, s_1 = i_1, \dots, s_{r-1} = i_{r-1}, s_r = j) \\
&= \max_{i_1, \dots, i_{r-1}} P(o_0, \dots, o_{r-1}, s_1 = i_1, \dots, s_{r-1} = i_{r-1}) \\
&\quad \cdot P(s_r = j | o_0, \dots, o_{r-1}, s_1 = i_1, \dots, s_{r-1} = i_{r-1}) \\
&\quad \cdot P(o_r | o_0, \dots, o_{r-1}, s_1 = i_1, \dots, s_{r-1} = i_{r-1}, s_r = j) \\
&= \max_{i_1, \dots, i_{r-1}} P(o_0, \dots, o_{r-1}, s_1 = i_1, \dots, s_{r-1} = i_{r-1}) \\
&\quad P(s_r = j | s_{r-1} = i_{r-1}) P(o_r | o_{r-1}, s_r = j) \\
&= \left[\max_{i_1, \dots, i_{r-1}} \delta_{r-1}(i_{r-1}) a_{i_{r-1} j} \right] b_{o_{r-1} o_r}^{(j)}.
\end{aligned} \tag{A.9}$$

B. Practical Computation of the Algorithms

Some algorithms in this paper may lead to numerical errors. A good solution is to normalize the intermediary results at each step of the calculation. A computable version of these algorithms is provided in this appendix.

B.1. *Computation of the Forward Procedure.* The forward variable $\alpha_r(j)$ may easily take values too small to be handled

by a computer. To avoid this problem, one solution is to normalize $\alpha_r(j)$ at each step r .

For $r = 1$ and $j = 1, \dots, N$, we define

$$\widetilde{\alpha}_1(j) = \frac{b_{o_0 o_1}^{(j)} \pi_j}{\bar{\alpha}_1}, \tag{B.1}$$

where $\bar{\alpha}_1 = \sum_{j=1}^N b_{o_0 o_1}^{(j)} \pi_j / N$ is the average value of the $\alpha_1(j)$.

Then, for $r = 2, \dots, k-1$, we have

$$\widetilde{\alpha}_r(j) = \frac{b_{o_{r-1} o_r}^{(j)} \sum_{i=1}^N a_{ij} \widetilde{\alpha}_{r-1}(i)}{\bar{\alpha}_r}, \tag{B.2}$$

where $\bar{\alpha}_r = b_{o_{r-1} o_r}^{(j)} \sum_{i=1}^N a_{ij} \widetilde{\alpha}_{r-1}(i) / N$ is the average value of the $\alpha_r(j)$.

As a result, the log-likelihood of the sequence of observed CA delays can be obtained as follows:

$$\log\{L(o_0, o_1, \dots, o_{k-1})\} = \log \left\{ \sum_{j=1}^N \widetilde{\alpha}_{k-1}(j) \right\} + \sum_{r=1}^{k-1} \log(\bar{\alpha}_r). \tag{B.3}$$

B.2. *Computation of the Backward Procedure.* The computation of the backward procedure leads to the same kind of problems as the forward procedure and requires the same type of solution.

For $r = k-1$ and $i = 1, \dots, N$, we define $\widetilde{\beta}_{k-1}(i) = 1$ and $\bar{\beta}_{k-1}(i) = 1$. Then, for $r = k-2, \dots, 1$, we have

$$\widetilde{\beta}_r(i) = \frac{\sum_{j=1}^N b_{o_r o_{r+1}}^{(j)} \widetilde{\beta}_{r+1}(j) a_{ij}}{\bar{\beta}_r}, \tag{B.4}$$

where $\bar{\beta}_r = \sum_{j=1}^N b_{o_r o_{r+1}}^{(j)} \widetilde{\beta}_{r+1}(j) a_{ij} / N$ is the average value of the $\beta_r(i)$.

Therefore, the log-likelihood of the sequence of observed CA delays can be rewritten as follows:

$$\log\{L(o_0, o_1, \dots, o_{k-1})\} = \log \left\{ \sum_{i=1}^N \widetilde{\alpha}_r(i) \widetilde{\beta}_r(i) \right\} + \sum_{t=1}^r \log(\bar{\alpha}_t) + \sum_{t=r}^{k-1} \log(\bar{\beta}_t). \tag{B.5}$$

B.3. *Computation of $\varepsilon_r(i, j)$ and $\xi_r(i)$.* Based on the scaled version $\widetilde{\alpha}_r$ and $\widetilde{\beta}_r$ of α_r and β_r , the computation of $\varepsilon_r(i, j)$ and $\xi_r(i)$ is achieved as follows.

For $r = 1, \dots, k-2$,

$$\varepsilon_r(i, j) = \exp \left\{ \log(\widetilde{\alpha}_r(i)) + \sum_{t=1}^r \log(\bar{\alpha}_t) + \log(a_{ij}) + \log(b_{o_r o_{r+1}}^{(j)}) \right\} + \log(\widetilde{\beta}_r(i)) + \sum_{t=r}^{k-2} \log(\bar{\beta}_t) - \log(L(o_0, o_1, \dots, o_{k-1})), \tag{B.6}$$

and, for $r = 1, \dots, k-1$,

$$\xi_r(i) = \exp \left\{ \log(\tilde{\alpha}_r(i)) + \sum_{t=1}^r \log(\tilde{\alpha}_t) + \log(\tilde{\beta}_r(i)) \right\} + \sum_{t=r}^{k-1} \log(\tilde{\beta}_t) - \log(L(o_0, o_1, \dots, o_{k-1})). \quad (\text{B.7})$$

For $r = 1, \dots, k-2$, the relation (18) holds. The reestimation of π , \mathbf{A} , and \mathbf{B} is obtained through formulas (19), (20), and (21):

B.4. Computation of the Optimal Sequence of Hidden Network States. In order to avoid numerical errors, it is necessary to scale the quantity δ used in the Viterbi algorithm.

For $r = 1$ and $j = 1, \dots, N$, we define

$$\tilde{\delta}_1(j) = \frac{\pi_j b_{o_0 o_1}^{(j)}}{\delta_1}, \quad (\text{B.8})$$

where

$$\tilde{\delta}_1 = \frac{\sum_{i=1}^N \pi(i) b_{o_0 o_1}^{(j)}}{N}. \quad (\text{B.9})$$

For $r = 2, \dots, k-1$ and $j = 1, \dots, N$, we compute iteratively:

$$\tilde{\delta}_r(j) = \frac{[\max_{i_1, \dots, i_{r-1}} \delta_{r-1}(i_{r-1}) a_{i_{r-1} j}] b_{o_{r-1} o_r}^{(j)}}{\tilde{\delta}_r}, \quad (\text{B.10})$$

$$\tilde{\delta}_r = \frac{\sum_{j=1}^N [\max_{i_1, \dots, i_{r-1}} \delta_{r-1}(i_{r-1}) a_{i_{r-1} j}] b_{o_{r-1} o_r}^{(j)}}{N}.$$

The optimal hidden state at time $k-1$ is then

$$\hat{s}_{k-1} = \arg \max_{j=1, \dots, N} \tilde{\delta}_{k-1}(j), \quad (\text{B.11})$$

and we obtain recursively for $r = k-2, \dots, 1$,

$$\hat{s}_r = \arg \max_{j=1, \dots, N} \tilde{\delta}_{r+1}(j) a_{j \hat{s}_{r+1}}. \quad (\text{B.12})$$

Finally, the joint probability of the sequence of hidden network states and the sequence of observed CA delay outputs is equal to

$$P(s_1, \dots, s_{k-1}, o_0, o_1, \dots, o_{k-1}) = \left[\max_{j=1, \dots, N} \tilde{\delta}_{k-1}(j) \right] \prod_{r=1}^{k-1} \tilde{\delta}_r. \quad (\text{B.13})$$

Data Availability

All data sets used in this study are provided within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported partially by the National Natural Science Foundation of China (61572032) and the Discipline (Major) Top-notch Talent Project of Anhui Provincial Universities (gxbjZD20).

References

- [1] G. C. Walsh, H. Hong Ye, and L. G. Bushnell, "Stability analysis of networked control systems," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 3, pp. 438–446, 2002.
- [2] M. Lješnjanić, D. Nešić, and D. E. Quevedo, "Robust stability of a class of networked control systems," *Automatica*, vol. 73, pp. 117–124, 2016.
- [3] Y.-L. Wang and Q.-L. Han, "Network-based modelling and dynamic output feedback control for unmanned marine vehicles in network environments," *Automatica*, vol. 91, pp. 43–53, 2018.
- [4] M. E. M. B. Gaid, A. Cela, and Y. Hamam, "Optimal integrated control and scheduling of networked control systems with communication constraints: application to a car suspension system," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 4, pp. 776–787, 2006.
- [5] P. Park, H. Khadilkar, H. Balakrishnan, and C. J. Tomlin, "High confidence networked control for next generation air transportation systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3357–3372, 2014.
- [6] A. Maciel, G. Sankaranarayanan, T. Halic, V. S. Arikatla, Z. Lu, and S. De, "Surgical model-view-controller simulation software framework for local and collaborative applications," *International Journal of Computer Assisted Radiology and Surgery*, vol. 6, no. 4, pp. 457–471, 2011.
- [7] M. Li and Y. Chen, "Challenging research for networked control systems: a survey," *Transactions of the Institute of Measurement and Control*, vol. 41, no. 9, pp. 2400–2418, 2019.
- [8] X. M. Zhang, Q. L. Han, X. H. Ge et al., "Networked control systems: a survey of trends and techniques," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 1–17, 2020.
- [9] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 84–99, 2001.
- [10] D. K. Kim, J. W. Ko, and P. Park, "Stabilization of the asymmetric network control system using a deterministic switching system approach," in *Proceedings of the Forty First IEEE Conference on Decision and Control*, pp. 1638–1642, Las Vegas, NV, USA, December 2002.
- [11] L. A. Montestruque and P. J. Antsaklis, "On the model-based control of networked systems," *Automatica*, vol. 39, no. 10, pp. 1837–1843, 2003.
- [12] D. Yue, Q. L. Han, and J. Lam, "Network-based robust H_∞ control of systems with uncertainty," *Automatica*, vol. 41, no. 6, pp. 999–1007, 2005.
- [13] A. Farnam and R. M. Esfanjani, "Improved stabilization method for networked control systems with variable transmission delays and packet dropout," *ISA Transactions*, vol. 53, no. 6, pp. 1746–1753, 2016.
- [14] M. B. G. Cloosterman, N. van de Wouw, W. P. M. H. Heemels, and H. Nijmeijer, "Stability of networked control systems with

- uncertain time-varying delays," *IEEE Transactions on Automatic Control*, vol. 54, no. 7, pp. 1575–1580, 2009.
- [15] H. Zhang, Z. Zhang, Z. Wang, and Q. Shan, "New results on stability and stabilization of networked control systems with short time-varying delay," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2772–2781, 2016.
- [16] Y.-L. Wang and G.-H. Yang, " H_∞ control of networked control systems with time delay and packet disordering," *IET Control Theory & Applications*, vol. 1, no. 5, pp. 1344–1354, 2007.
- [17] H. Song, G.-P. Liu, and L. Yu, "Networked predictive control of uncertain systems with multiple feedback channels," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 11, pp. 5228–5238, 2013.
- [18] D. Yue, E. G. Tian, Z. D. Wang et al., "Stabilization of systems with probabilistic interval input delays and its applications to networked control systems," *IEEE Transactions on Systems, Man and Cybernetics - Part A*, vol. 39, no. 4, pp. 939–945, 2009.
- [19] C. Peng, D. Yue, E. Tian, and Z. Gu, "A delay distribution based stability analysis and synthesis approach for networked control systems," *Journal of the Franklin Institute*, vol. 346, no. 4, pp. 349–365, 2009.
- [20] D. Yue, E. G. Tian, Y. J. Zhang et al., "Delay-distribution-dependent stability and stabilization of T-S fuzzy systems with probabilistic interval delay," *IEEE Transactions on Systems, Man and Cybernetics - Part B*, vol. 39, no. 2, pp. 503–516, 2009.
- [21] C. Peng and T. C. Yang, "Communication-delay-distribution-dependent networked control for a class of T-S fuzzy systems," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 2, pp. 326–335, 2010.
- [22] X. He, Z. Wang, and D. H. Zhou, "Robust fault detection for networked systems with communication delay and data missing," *Automatica*, vol. 45, no. 11, pp. 2634–2639, 2009.
- [23] W. Wu, "Fault-tolerant control of uncertain non-linear networked control systems with time-varying delay, packet dropout and packet disordering," *IET Control Theory & Applications*, vol. 11, no. 7, pp. 973–984, 2017.
- [24] D. Sauter, S. Li, and C. Aubrun, "Robust fault diagnosis of networked control systems," *International Journal of Adaptive Control and Signal Processing*, vol. 23, no. 8, pp. 722–736, 2009.
- [25] H. Zhang and J. Wang, "Robust two-mode-dependent controller design for networked control systems with random delays modelled by Markov chains," *International Journal of Control*, vol. 88, no. 12, pp. 2499–2509, 2015.
- [26] C. Yang, Z.-H. Guan, and J. Huang, "Design of stochastic switching controller of networked control systems based on greedy algorithm," *IET Control Theory & Applications*, vol. 4, no. 1, pp. 164–172, 2010.
- [27] L. Zhang, Y. Shi, T. W. Chen et al., "A new method for stabilization of networked control systems with random delays," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1177–1181, 2005.
- [28] S. Cong, Y. Ge, Q. Chen, M. Jiang, and W. Shang, "DTHMM based delay modeling and prediction for networked control systems," *Journal of Systems Engineering and Electronics*, vol. 21, no. 6, pp. 1014–1024, 2010.
- [29] Y. Liu and D. Sun, "Delay-dependent H_∞ stabilisation criterion for continuous-time networked control systems with random delays," *International Journal of Systems Science*, vol. 41, no. 11, pp. 1399–1410, 2010.
- [30] Y. Ge, S. Cong, and W. Shang, "K-means based delay quantization and prediction in networked control systems," in *Proceedings of the 31st Chinese Control Conference*, pp. 5921–5926, Hefei, China, July 2012.
- [31] D. Huang and S. K. Nguang, "State feedback control of uncertain networked control systems with random time delays," *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 829–834, 2008.
- [32] Y. Ge, Q. Chen, M. Jiang, and Y. Huang, "SCHMM-based modeling and prediction of random delays in networked control systems," *Journal of the Franklin Institute*, vol. 351, no. 5, pp. 2430–2453, 2014.
- [33] Y. Ge, X. Zhang, Q. Chen, and M. Jiang, "Initialization of the HMM-based delay model in networked control systems," *Information Sciences*, vol. 364–365, pp. 1–15, 2016.
- [34] Y. Ge, S. Cong, and W. Shang, "DTHMM based optimal controller design for networked control systems," *Journal of University of Science and Technology of China*, vol. 42, no. 2, pp. 161–169, 2012.
- [35] T. Liu, D. J. Hill, and J. Zhao, "Synchronization of dynamical networks by network control," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1574–1580, 2012.
- [36] Q. Li, B. Shen, Z. D. Wang et al., "Recursive distributed filtering over sensor networks on Gilbert-Elliott channels: a dynamic event-triggered approach," *Automatica*, vol. 113, Article ID 108681, 2020.
- [37] Q. Li, Z. Wang, W. Sheng, F. E. Alsaadi, and F. E. Alsaadi, "Dynamic event-triggered mechanism for H_∞ non-fragile state estimation of complex networks under randomly occurring sensor saturations," *Information Sciences*, vol. 509, pp. 304–316, 2020.