1. Introduction

As a kind of control system [1–6], NCSs have motivated a lot of research studies in the control field during recent years [7–10] by many advantages it shares; for example, reduced weight, high flexibility, simple installation and maintenance, and low cost. Because of those practical characteristics, more and more efforts have been devoted to these systems [11–13]. Admired for past achievements, NCSs have been found in widespread applications such as feedback control systems [14, 15], stabilization of linear systems [16], control of nonlinear systems [17], and adaptive tracking control of nonlinear multiagent systems [18–20]. Compared with traditional point-to-point hardwiring system, typical NCSs is a young generation of control architectures which has a feedback control structure consisting with the controller, sensor, and actuator through network communication. The insertion of this performance condition, with a finite bandwidth, brings serial challenging and undesirable issues on account of data packet loss, scheduling, and latency in the process of communication signal’s transmission to remote analog input/output unit. Affected by these cases, the communication channel, which takes the place of traditional signal transmission technologies, brings hard-to-solve problems in the stability analysis and controller synthesis because of transmission time delays. As a result, this issue received increasing interests in this field [21–23]. Traditional time-delay analysis usually assumes that the time delay is constant, time-varying or obeys some random distribution, and rarely analyzes it from the perspective of system operation mechanism. Modeling from the perspective of operation mechanism can effectively show various state changes that may occur in the system and the relationship between changes. This prompts us to adopt HCPN to carry out structural reservation modeling and then analyze the time-delay characteristics.

In 1992, Petri nets were introduced by Petri [24] firstly as a net-theoretic approach to implement a particular purpose. The relationships between departments could be represented by a net, and it is a good approximation to imitate the appearance and character of asynchronous and concurrent operation in discrete event systems. Petri nets is a kind of mathematical structures which is a bipartite-directed graph consisting of two kinds of compositions; one is drawn as a
circle which is called Place, and another drawn as a rectangles named Transition. These two nodes combine with each other with arcs which are drawn as arrows. Coloured Petri net (CPN) [25–28], in which groups of objects is thus distinguished by colour, inherits all the advantages of classical Petri net. The wide practical application distributes on the direction of distributed control systems [29] and domains environments [30]. Recently, a novel of theoretical results has been done on the application of Petri nets for NCSs [31, 32]. Regarding the execution of the NCSs practical operational fundamental principle, the CPN has been chosen to build a formalism model [31]. With the rapid development of scientific technologies, the process in large-scale systems is more and more complicated, which makes it hard to solve problems in modeling. Therefore, to solve the problem of state explosion during formal verification, HCPN is invented [33–35] which is a new type of Petri net for creating large-scale and complex systems. Its main purpose is to summarize the system model with simple network model and to expand and fill it with substitution change. This method is not only beneficial to the excessive number of reservoirs, transitions, and arcs in the model but also beneficial to avoid the explosion of system state space and simplify the analysis of the model. In [31], a mapping from the established hypothetical delays in Ethernet of NCSs to entities of a CPN model was defined, and the simulation analysis of network induced delays was tested and verified in a simple way. Following a similar way, in [32], a two-step approach was included in the estimation of delays in the modeling of NCSs. However, the Ethernet model in the proposed model was studied in a hypothesis, in which time delays were not interrelated with network bandwidth-limited bit limit and packet dropouts. Further investigation and analysis is in [36, 37] and the references therein.

The primary contributions of this paper are as follows:

(i) First, considering the potential cause for the deteriorating performance or instability on NCSs, this paper sets network band delay in an interval instead of the definite transmission delay in [32] to make the model more realistic. In this method, it is supported by the HCPN model to simulate some challenging issues on account of network information propagation.

(ii) Second, the CPN model in [31, 32] is not considered data packet latency and dropout in the Ethernet CPN model of NCSs. Focusing on those challenges in network transmission, this paper has some alterations on the base of the model in [32]; close to the reality, data packets’ dropout and packet out-of-order are taken into consideration in the proposed model of data transmission on the network.

(iii) Third, the exploration and research of random time delays existing in the sensor-to-controller and controller-to-actuator in discrete time networked control systems are acquiescence with random Markovian delays in recent literature [38–40].

However, it is not clear if this theoretical acquiescence can be put into practice. This paper makes a certification of equivalence between time delays and Markov model [41].

This paper is organized as follows. Section 2 is the conceptual framework of Petri net. Section 3 is the HCPN model of NCSs-PLC. Section 4 analyzes the time delay extracted from the HCPN model. Section 5 is the equivalence analysis of the time delay of both the Markov model and the HCPN model.

2. Basic Conception

Definition 1. A ordinary non-HCPN can be defined as a nine elements’ tuple [42],

\[ \text{Traditional CPN} = (\Sigma, P, T, A, N, C, G, E, I), \]

satisfying the requirements below:

(i) \( \Sigma \) is called colour set which is a finite set describing nonempty types.

(ii) \( P \) is Place representing a ellipse which interprets a passive component with discrete status.

(iii) \( T \) is Transition establishing with a rectangle which explains an active component; tokens can consume, produce, and change the carrying information in Transitions.

(iv) \( A \) is arcs which connects Places and Transitions in the model. It can be represented by arrows, and it is a finite set which meets the expectations with \( \forall P_i \in P; T_j \in T; P_i \cap T_j = P_i \cap A_j = T_j \cap A_i = \emptyset \).

(v) \( N \) is defined as a map into arcs. It is defined from \( A \) into \( P_i \times T_j \cup T_j \times P_i \) which has two elements, the first element means arcs’ source and the second element means arcs’ destination.

(vi) \( C \) is defined a map in places. It is defined from \( P_i \) into \( \Sigma \) which means that every token on every \( P_i \) has a corresponding colour set type.

(vii) \( G \) is the guard function which is defined a map in transitions. \( G(T_j) \) is the type appertain to \( \Sigma \), and the binding must perform every Boolean expressions. It is can be shown as \( \forall T_j \in T [ \text{Type} (\text{Guard}(T_j))) \subseteq \Sigma \land \text{Type} (\text{Guard}(T_j)) = \text{Boolean} \).

(viii) \( E \) is called arc expression which maps every element in \( A \) to an expression, and the type of it can be written as \( C(P_i)_{\text{MS}} \) it can be shown as

\[ \forall a = A \left[ \text{Type}(E_j) = C(P_j)_{\text{MS}} \wedge \text{Type}(\text{Var}(E_j)) \subseteq \Sigma \right], \]

where \( P_i \) is defined as the place of \( N_i \).

(ix) \( I \) illustrates a map between \( P_i \) and the type \( C_i \). It is the initialization function which can be shown as

\[ \forall P_i \in P: \left[ \text{Type}(I_i) = C_i \right]. \]
Definition 2. A HCPN can be defined as a nine-element tuple satisfying the requirements below [42]:

\[ HCPN = (S_s, S_r, N_s, S_r A, PLPN, PO_T, PO_A, PLPS, FT, PP). \]

(i) \( S_r \) is a congregations for nonhierarchical pages in the model, and each page does not have collaborative net elements. It can be shown as

\[ \{ s_0, s_1, \ldots, s_i \mid n \in N^* \land s_0 = s_n \land \forall k \in 1, \ldots, n: s_k \in S_r A \left( SN_{S_{(k-1)}} \right) \} = \emptyset. \]

(iv) \( PLPN \subseteq P \) is defined as a set of place nodes or transition nodes.

(v) \( POT \) is a function of port type, and it has four types, in type, out type, in/out type, and general.

\[ \forall t \in S_r N: PO_A (t) \subseteq X(t) \times PLPN - S_r A(t), \]

\[ \forall t \in S_r N: \forall \left( p_1, p_2 \right) \in PO_A: PO_T \left( p_2 \right) \neq \text{general} \Rightarrow ST \left( p_1, t \right) = PT \left( p_2 \right), \]

\[ \forall t \in S_r N: \forall \left( p_1, p_2 \right) \in PO_A: \left[ C \left( p_1 \right) = C \left( p_2 \right) \land I \left( p_1 \right) = I \left( p_2 \right) \right]. \]

(vii) \( PLFS \subseteq P \) is a finite set of fusion sets such that

\[ \forall f s \in PLFS: \forall p_1, p_2 \in f s: \left[ C \left( p_1 \right) = C \left( p_2 \right) \land I \left( p_1 \right) = I \left( p_2 \right) \right]. \]

(viii) \( FT \) is a function of the fusion type. It is defined from fusion into \{global, page, instance\}, such that

\[ \forall f s \in FS: \left[ FT \left( f s \right) \neq \text{global} \Rightarrow \exists s \in S: f s \subseteq Ps \right]. \]

(ix) \( PP \in S_{MS} \) is a multiset of the prime page.

3. The CPN Model for NCSs

3.1. The Top Model. The integrated model can be divided into two levels, one is top level which simulates the relationship between the controller, the sensor, and the actuator with a cursory method, as shown in Figure 1. The other level is the detailed function description of the controller, the sensor, and the actuator. In the proposed top-level model, the sensor and the actuator connect with PLC Remote IO. PLC-CPU sends packets from the controller to PLC Remote IO through Ethernet; then, PLC Remote IO sends acknowledgments from the sensor and the actuator back to PLC-CPU. There are three subpages and one top page in the whole model. After some introductions of time delay in PLC-CPU, those pages are explained one by one in detail.
Continuous time controlled object

Figure 1: The top level in the CPN model for NCSs.

Controller

Transmission queue

Sensor

Ethernet

Execution

Output Input

Communication module---I/O scan time

Figure 2: Network control system with PLC being the CPU controller.

Sensor node

Actuator node

PLC communication module

Sensor signal with periodic sampling

Input port receiving datapacket with IO scan Tc

PLC communication processing data packet with scan Tp

Output port sending data packet with IO scan Tc

Event actuator received data packets.

Data packet lost

Figure 3: Process of time delay in PLC-CPN.
3.3. The Controller Model. The module of controller is divided into two parts. One is the communication module part, which is used to sending and receiving data with scan cycles, such as left part in Figure 4. The another part is the PLC-CPU model, which simulates functions of CPU reading(input), CPU executing, and CPU writing(output), such as right part in Figure 4. The Send\_R transition and the Rec\_A transition complete data packets’ sending and receiving, respectively. PLC\_R transition, PLC\_E transition, and PLC\_W transition complete simulation of reading, executing, and writing in the PLC-CPU. The definition of token elements in the model is shown in Table 1.

3.4. The Ethernet Model. Figure 5 is the Ethernet model which is the primary part for whole NCSs’ system. There are two modified points contrasting with the model which has been designed by Ghanaim et al. [32]. When the data packets transmit in Ethernet, transmission will be influenced by a lot of uncertainties. In consideration of these cases, the transmission time is set into a random time with an interval time varying rather than a fix data. In the proposed model, time delays of this portion are discrete uniform distributions between 1000 and 2000 when packet data passed from the sensor to the controller or from the controller to the actuator. In addition, in the actual network transmission, the data packets’ transmission is unstable because of some uncertain factors, such as packets lost and packets out of order. In this model, packets will be lost in ten percent probability, and this function is realized by fun Ok(s; Ten0;r; Ten1) = (r < = s). Data packets will be retransmitted based on Transmission Control Protocol/Internet Protocol if those transmission mistakes occurred. The mathematic presentation of this model is a tuple:

\[ HCPN = (S, SN, SA, PN, PT, PA, FS, FT, PP). \]  

By the definition of HCPN, those elements in this tuple can be used to describe the Ethernet model, as shown in Figure 6.

\[ \Sigma \] is the Token colour set in Figure 5. \( P \) is finite set of places and \( T \) is finite set of transitions. \( A \) is arc, and \( C \) to Store1 means that this arc is from place \( C \) to transition Store1. \( N \) is the node function. \( G \) is guard, for example, \((\#A, p1) = N, T = Net1, Net2 means that when transition is Net1 or Net2, the guard of transition is \((\#A, p1) = N; \) otherwise, transition guards are true. \( C \) is the colour definition of Place. Place \( A, B, C, \) and \( D \) are mean the Packet colour. \( I(P) \) is the initialization set if the place is \( SP1 \) or \( SP2, \) and the initial token is integer nine. \( E(a) \) means the arc expression in the Ethernet model, \( p1, a \in \{A to Net1, C to Store1 \cdot Store1 to A\} \) means when the arc is from \( A \) to Net1, \( C \) to Store1, and Store1 to \( A, \) and the arc expression on those arc is \( p1. \)

4. Time-Delay Analysis

4.1. Transmission Delay in Ethernet. Network transmission is an important element in the analysis of \( Tsc \) and \( Tca. \) Besides, it also can help to analyze the system state and performance. Data packets may be lost at a probability such as ten percent in the model of network transmission. With reference to Transmission Control Protocol/Internet Protocol (TCP/IP), data packets would be retransmitted if it is lost in the network transmission. Figure 7 is network transmission latency which is collected in the proposed HCPN model running 5050 steps. From Figure 7, it can be clearly see that, in time delay \( Tsc \) and \( Tca, \) network transmission time delays are almost greater than 1000 and less than 2000; however, there are still ten percent points out of scope, Figure 7 shows several peak points, for example A, B, and C, which mean that, in this time, data packets are dropped out in network transmission. It will be retransmitted according to TCP/IP so that the value of it becomes very bigger than others. The network transmission latency was the transmission time and the retransmission time and other time spent on extracting data packets.

4.2. The Time Delay \( Tca \) and \( Tsc. \) There are three scan cycles which play important roles in the calculation of backward time delay and forward time delay. The first one is periodic scan \( Tp \) for CPU program, the second one is periodic IO scan \( Tc \) for the communication module, and the last one is the periodic sampling \( Tth \) for the sensor. There exists another important data to calculate: network transmission delay. Table 2 is the parameters defined in the NCSs model which can be used to simulate and analyze the efficiency of the difference system with the HCPN model.

In addition, in the communication module, IO periodic scan cycle \( Tc \) is a summation of reading time, executing time, and writing time. The periodic scan \( Tp \) for CPU program is set to 17,000 and the periodic sampling \( Tth \) for the sensor is 1000. When network transmission latency was obtained, such as in Figure 7, \( Tca \) and \( Tsc \) can be calculated under the rule of scan cycles, as shown in Figure 8. Before the calculation of \( Tca \) and \( Tsc, \) it is necessary to obtain the network latency, which are produced in network transmission and waiting time for the scan cycle. Contrasting with delays obtained from the HCPN model in [32], it can be clearly see that because of the indeterminacy transmission phenomenon and the random transmission time delay the periodicity of \( Tca \) and \( Tsc \) is broken.

5. Verification and Analysis

5.1. Markov Modeling. A Hidden Markov Model (HMM) is a statistical Markov lain with hidden states which cannot be directly visible, but the sequence of observations can give some information of HMM. The sequence of observations alphabet is set to \( O_{seq}, \) and a hidden states’ alphabet sequence \( S_{seq} \) can be cached from \( O_{seq}. \) \( O_{seq} \) and \( S_{seq} \) are shown below:

\[ O_{seq} = \{o_1, o_2, \ldots, o_k, \ldots\}, \]
\[ S_{seq} = \{s_1, s_2, \ldots, s_j, \ldots\}, \]  

where \( O_k \in O = \{o_1, o_2, \ldots, o_N\} \) is the observation state at time \( k \) and \( S_j \in S = \{s_1, s_2, \ldots, s_N\} \) is the corresponding state at observation \( O_k. \)
The formal definition of a hidden Markov model (HMM) is as follows:

\[ \lambda = \{A, B, \pi_0\} \]

where \( A \) is the state transition probability matrix form one state to another state, which can be written as

\[ A = \{a_{ij}\}, \]

\[ \{a_{ij}\} = P\{s_{k+1} = s_j | s_k = s_i\} \]

\( B \) is the observation probability matrix which can be written as

\[ B = \{b_{ij}\}, \]

\[ \{b_{ij}\} = P\{o_{t+1} = o_j | s_k = s_i\} \]

and \( \pi_0 \) is the initial state probability which can be written as

\[ \pi_0 = \{\pi_i\}, \]

\[ \{\pi_i\} = P\{s_1 = s_i\}. \]

5.2. Markov Model Results. By the definition of HMM and the Baum Welch algorithm, the algorithm steps are as follows.

Step 1 : for a given observation sequence and HMM model, the probability variables of the hidden state \( S_i \) are

\[ \delta_t(i) = \frac{\alpha_t(i)\beta(i)}{P(O|\lambda)} \]

\[ P(O|\lambda) = \sum_{i=1}^{N} \alpha_t(i)\beta(i), \]

where \( \alpha_t(i) \) is called forward probability, where

\[ \alpha_t(i) = P\{o_1, o_2, \ldots, o_t, q_t = S_i | \lambda\}, \]

and \( \beta(i) \) is called backward probability, where

\[ \beta_t(i) = P\{o_{t+1}, o_{t+2}, \ldots, o_{T-1}, o_T, q_t = S_i | \lambda\}, \]

Step 2 : define the probability between the hidden state \( S_i \) and the hidden state \( S_j \) as follows:

\[ \phi_t(i, j) = P\{q_t = S_i, q_{t+1} = S_j | O, \lambda\}, \]

\[ \phi_t(i, j) = \frac{\alpha_t(i)a_{ij}\beta_{t+1}(j)}{P(O|\lambda)}, \]

\[ P(O|\lambda) = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i)a_{ij}\beta_{t+1}(j). \]

Step 3 : the formula between \( \delta_t(i) \) and \( \phi_t(i, j) \) is
Step 4: find the corresponding expected values of the above two variables $\sum_{i=1}^{N} \delta_i(i)$ and $\sum_{i=1}^{T-1} \phi_i(i, j)$.

Step 5: the new HMM parameters are estimated by using the two variables defined above and their expected values:

$$\delta_i(i) = \frac{N}{\sum_{j=1}^{N} \phi_i(i, j)}.$$  (20)

$$\tilde{a}_{ij} = \frac{\sum_{t=1}^{T-1} \phi_t(i, j)}{\sum_{t=1}^{T-1} \delta_t(i)}.$$  (21)

$$\tilde{b}_j(k) = \frac{\sum_{t=1}^{T-1} \phi_t(i, j)}{\sum_{t=1}^{T-1} \delta_t(i)}.$$  (22)

The Markov model can be built based on time delay $Tsc$ and $Tcs$ obtained from HCPN. In order to make analyzing more clear, data will be managed with the rounding method. From Figure 8, it two time delays can be obtained as follows:

$$Tsc = [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17],$$

$$Tca = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 29].$$  (23)

$Tsc$ can be divided into two categories: one category is the low delay $Tsc_{low}$ which includes the value $Tsc_{low} = [3, 4, 5, 6, 7, 8, 9]$. The other category is high delay $Tsc_{high}$ which includes the value $Tsc_{high} = [10, 11, 12, 13, 14, 15, 16, 17]$.

The time delay $Tsc$ can be written as

$$Tsc = \begin{cases} Tsc_{low}, & s_k = s_1, \\ Tsc_{high}, & s_k = s_2. \end{cases}$$  (24)

The same procedure can be easily adapted to $Tca$ which can be divided into two categories: one category is the low delay $Tca_{low}$ which includes the value $Tca_{low} = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19]$ and the other category is high delay $Tca_{high}$ and $Tca_{high} = [20, 21, 22, 23, 24, 25, 26, 27, 29]$. The time delay $Tca$ can be written as

$$Tca = \begin{cases} Tca_{low}, & s_k = s_1, \\ Tca_{high}, & s_k = s_2. \end{cases}$$  (25)

Figures 9 and 10 show the probability distribute of time delay $Tsc$ and $Tca$ which are the experimental data in HCPN, respectively.

The parameters in the HMM model can be calculated with Baum–Welch algorithm and maximum likelihood estimate, and the system $\lambda_{sc} = \{A_{sc}, B_{sc}, \pi_{0sc}\}$ with 2-state and 15-observations, and the matrix $A_{sc}, B_{sc}$ can be calculated by Matlab.
\[ \Sigma = \{ \text{UNIT, BOOL, INT, TIME, STRING, Packet, Data, Ten0, Ten1, N, A, I, O} \} \]

\[ P = \{ C, A, K1, K2, SP1, SP2, avail, B, D \} \]

\[ T = \{ \text{Store1, Store2, Net1, Net2} \} \]

\[ A = \{ \]
\[ \text{CtoStore1} \]
\[ \text{BtoNet2} \]
\[ \text{Net1toB} \]
\[ \text{Net2toSP2toNet2} \]
\[ \}

\[ N = (\text{Source dest}) \]

\[ G = \begin{cases} 
(\#A, p1) = N & \text{true} \\
\text{otherwise} & 
\end{cases} \]

\[ T = \{ \text{Net1, Net2} \} \]

\[ C = \{ \]
\[ \text{Packet} \]
\[ \text{INT} \]
\[ \text{Ten0} \]
\[ \text{UNIT} \]
\[ \}

\[ k = \{ \]
\[ (\#A, p1) = N & 1 \\
\[ \text{otherwise} & 9 \\
\[ \}

\[ E(a) = \{ \]
\[ a \in \{ \text{AtoNet1, CtoStore1, Store1toA} \} \]
\[ a \in \{ \text{BtoNet2, DtoStore2, Store2toB} \} \]
\[ a \in \{ \text{K1toNet1, K2toNet2} \} \]
\[ a \in \{ \text{Net1toSP1, SP1toNet1, Net2toSP2, SP2toNet2} \} \]
\[ a \in \{ \text{Net1toC, Net2toD} \} \]
\[ a \in \{ \text{Net1toB} \} \]
\[ a \in \{ \text{Net2toA} \} \]
\[ a \in \{ \text{Net1toK1} \} \]
\[ a \in \{ \text{Net2toK2} \} \]
\[ \}

\[ \text{Network trans delay in time delay } T_{sc} \]

\[ \text{Network trans delay in time delay } T_{ca} \]

\[ \text{Figure 6: Mathematical description for HCPN.} \]

\[ \text{Figure 7: Network transmission latency in the HCPN model for NCSs.} \]

\[ T \text{able 2: Main parameters.} \]

<table>
<thead>
<tr>
<th>Module</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLC module</td>
<td>17000</td>
</tr>
<tr>
<td>Ethernet module</td>
<td>2000</td>
</tr>
<tr>
<td>Remote analog input/output</td>
<td>1000</td>
</tr>
</tbody>
</table>

\[ A_{sc} = \begin{bmatrix} 0.28 & 0.72 \\ 0.84 & 0.15 \end{bmatrix}, \]

\[ B_{sc} = \begin{bmatrix} B_{sc1} & B_{sc2} \end{bmatrix}, \]

\[ B_{sc1} = \begin{bmatrix} 0.15 & 0.10 & 0.15 & 0.15 & 0.13 & 0.17 \\
0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \]

\[ B_{sc2} = \begin{bmatrix} 0.20 & 0.20 & 0.18 & 0.20 & 0.12 & 0.03 & 0.04 & 0.03 \end{bmatrix}. \]
In the similar way, the system $\lambda_{ca} = \{A_{ca}, B_{ca}, \pi_{0_{ca}}\}$ with 2-state and 19-observations can be calculated with $A_{ca} = \begin{bmatrix} 0.78 & 0.22 \\ 0.25 & 0.75 \end{bmatrix}$, $B_{ca} = \left[ \begin{array}{c} B_{ca}^1, B_{ca}^2 \end{array} \right]$, $B_{ca}^1 = \begin{bmatrix} 0.06 & 0.10 & 0.08 & 0.11 & 0.11 & 0.11 & 0.08 & 0.19 & 0.10 & 0.06 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$, $B_{ca}^2 = \begin{bmatrix} 0.19 & 0.12 & 0.10 & 0.15 & 0.06 & 0.15 & 0.15 & 0.06 & 0.02 \end{bmatrix}$.

(26)

5.3. Validation Results. From the above section, the important matrixes can be obtained as

\[ A = \{A_{sc}, A_{ca}\}, \]
\[ B = \{B_{sc}, B_{ca}\}, \]

(27)

Using the data from those two matrices, the stationary distribution of the Markov model $\lambda_{sc}$ and $\lambda_{ca}$ can be estimated by those two time delays ($Tca$ and $Tsc$) with the Matlab statics toolbox. The probability density function (PDF) can be calculated by sample arrays which can be obtained from HCPN consecutive time delay sequence. The PDF is the distribution of the time delay which can be compared with the stationary distribution of the Markov model.

Figures 11 and 12 are the stationary distribution of the Hidden Markov Model building above, and every observation probability value is similar to the probability density function of the time-delay sequence. The figures show that the Markov stationary distribution of both the time delay $Tsc$ and the time delay $Tca$ are approximately equivalent to the probability density function of HCPN. It is clearly seen
that building a Markov modeling of NCSs is feasible and time delays conform to Markov law.

6. Conclusion

In this paper, we have investigated a novel HCPN model approach for the network control system with PLC. The modified structure-conserving model has been accomplished to calculate sequences of delays under the control of scan cycles for PLC-CPU. Besides, a series of special phenomenon has been taken into consideration based on the traditional mode, such as data packets drop and data packets out of order. Time delays in network transmission have been calculated to observe the transmission data packet state in Ethernet. Finally, Markov analytical models have been built for analyzing time delays which can be obtained from the forward step mathematically.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant nos. 61403278 and 61503280).

References


