Hindawi
Discrete Dynamics in Nature and Society
Volume 2021, Article ID 9948855, 11 pages
https://doi.org/10.1155/2021/9948855

Research Article

Modeling and Analysis of Network Control System Based on Hierarchical Coloured Petri Net and Markov Chain

Jingdong Li,[1,2] **Zhangang Wang**,[1,2] **Liankun Sun**,[1,2] **and Wanru Wang**[1,2]

[1]*School of Computer Science and Technology, Tiangong University, Tianjin 300387, China*
[2]*Tianjin Key Laboratory of Autonomous Intelligence Technology and Systems, Tiangong University, Tianjin 300387, China*

Correspondence should be addressed to Zhangang Wang; wangzhangang@tiangong.edu.cn

Received 8 March 2021; Revised 24 April 2021; Accepted 28 May 2021; Published 9 June 2021

Academic Editor: Zi-Peng Wang

This paper investigates a modified modeling of networked control systems (NCSs) with programmable logic controller (PLC). First, the controller-to-actuator and sensor-to-controller network-induced delays are investigated by a modeling tactics based on hierarchical coloured petri net (HCPN) in a structure-conserving way. Comparing with the recent result, the signal transmission delay is set in a random interval instead of a fixed mode; moreover, the data packet drop out and disorder are also taken into consideration. Second, delays captured form CPN tools are analyzed with a strategy based on Baum–Welch algorithm and statistics science. Besides, time delays are modeled as a Markov chain and the transition probabilities is calculated using the consequent from the previous operation. Finally, a comparison verification illustrates the equivalence property between proposed models.

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 drown 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 interrelating 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],

$$\text{Traditional CPN} = (\Sigma, P, T, A, N, C, G, E, I), \tag{1}$$

satisfying the requirements below:

(i) Σ 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 $\forall P_i \in P; T_i \in T$: $P_i \cap T_i = P_i \cap A_i = T_i \cap A_i = \varnothing$.

(v) N is defined as a map into arcs. It is defined from A into $P_i \times T_i \cup T_i \times P_i$ 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 Σ 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. $G(T_i)$ is the type appertain to Σ, and the binding must perform every Boolean expressions. It is can be shown as $\forall T_i \in T$ [Type (Var (Guard (T)))$\subseteq\Sigma\wedge$Type (Guard (T_i)) = 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 $C(P_i)_{\mathrm{MS}}$; it can be shown as

$$\forall a = A\left[\mathrm{Type}(E_i) = C\left(\widehat{P}_i\right)_{\mathrm{MS}} \wedge \mathrm{Type}\left(\mathrm{Var}(E_i)\right) \subseteq \Sigma\right], \tag{2}$$

where $\widehat{P}_i$ is defined as the place of N_i.

(ix) I illustrates a map between P_i and the type $\widehat{C}_i$ It is the initialization function which can be shown as

$$\forall P_i \in P\colon \left[\mathrm{Type}(I_i) = \widehat{C}_i\right]. \tag{3}$$

Definition 2. A HCPN can be defined as a nine elements-tuple satisfying the requirements below [42]:

$$\mathrm{HCPN} = (S_e, S_eN, S_eA, \mathrm{PL_{PN}}, \mathrm{PO}_T, \mathrm{PO}_A, \mathrm{PL_{FS}}, \mathrm{FT}, \mathrm{PP}). \tag{4}$$

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

$$\left[\Sigma_{S_i}, P_{S_i}, T_{S_i}, A_{S_i}, N_{S_i}, C_{S_i}, G_{S_i}, E_{S_i}, I_{S_i}\right], \quad \forall S_1, S_2 \in S_e: \left[S_1 \neq S_2 \Rightarrow \left(P_{S_1} \cup T_{S_1} \cup A_{S_1}\right) \cap \left(P_{S_2} \cup T_{S_2} \cup A_{S_2}\right) = \varnothing\right]. \tag{5}$$

(ii) $S_eN \in T$ is substitution nodes.

(iii) S_eA is called page assignment, and every page must not be the subpage of itself; it is satisfied by the following conditions:

$$\left\{\widehat{s}_0, \widehat{s}_1, \ldots, \widehat{s}_i \in \widehat{S}_i | n \in N^+ \wedge \widehat{s}_0 = \widehat{s}_n \wedge \forall k \in 1, \ldots, n\colon s_k \in S_eA\left(\mathrm{SN}_{S_{i(K-1)}}\right)\right\} = \varnothing. \tag{6}$$

(iv) $\mathrm{PL_{PN}} \subseteq P$ 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.

(vi) PO_A is port assignment. It is binary relations such that

$$\begin{aligned} &\forall t \in S_eN\colon \mathrm{PO}_A(t) \subseteq X(t) \times \mathrm{PL}_{\mathrm{PN}-S_eA_{(T)}}, \\ &\forall t \in S_eN\colon \forall\left(p_1, p_2\right) \in \mathrm{PO}_A\colon \mathrm{PO}_T(p_2) \neq \text{general} \Rightarrow \mathrm{ST}(p_1, t) = \mathrm{PT}(p_2), \\ &\forall t \in S_eN\colon \forall\left(p_1, p_2\right) \in \mathrm{PO}_A\colon [C(p_1) = C(p_2) \wedge I(p_1) = I(p_2)]. \end{aligned} \tag{7}$$

(vii) $\mathrm{PL_{FS}} \in P_s$ is a finite set of fusion sets such that

$$\begin{aligned} &\forall fs \in \mathrm{PL_{FS}}\colon \\ &\forall p_1, p_2 \in fs\colon [C(p_1) = C(p_2) \wedge I(p_1) = I(p_2)]. \end{aligned} \tag{8}$$

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

$$\begin{aligned} &\mathrm{FT}\colon \mathrm{FS} \longrightarrow \{\text{global, page, instance}\}, \\ &\forall fs \in \mathrm{FS}\colon [\mathrm{FT}(fs) \neq \text{global} \Rightarrow \exists s \in S\colon fs \subseteq Ps]. \end{aligned} \tag{9}$$

(ix) $\mathrm{PP} \in S_{\mathrm{MS}}$ 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.

3.2. The Analysis of the Time Delays in PLC-NCSs. Figure 2 is a representative of traditional NCSs with the PLC controller. The sensor and actuator are connected to digital/analog I/O section through Ethernet. Before entering or leaving network, data packets will enter into a transmission queue to wait for the scan cycle of I/O.

Figure 3 gives detailed instructions of time delays in NCSs with the PLC controller, in which different samples were shown as different length arrows. In the upper part of the whole figure, there are some samples which are received in the sensor with periodic sampling. In the middle part, there are three kinds of axis showing the PLC controller. Before data packets arrive at execute program, it has to wait for the scan cycle so that it can be read into PLC. Both of the reading scan cycle and writing scan cycle in PLC communication are Tc. Another primary scan cycle is Tp which means that only at Tp scan cycles data packets can be captured into PLC-CPU to execute the programme. After a few execution time (τ_c), the communication module sends result data to the actuator node. During the process of the data packets' transmission, two network-induced delays are produced. One is called backward time delay and another is called forward time delay. The execution time delay in the controller is included in Tsc backward time delay which is the time costing in the transmission from the controller to the sensor, and Tca is forward time delay which is from CPU which received the sample to the actuator which received the same sample.

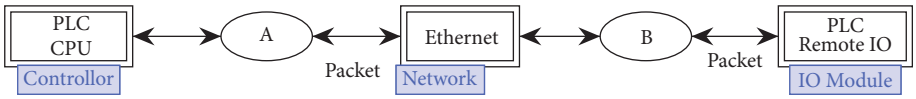

Figure 1: The top level in the CPN model for NCSs.

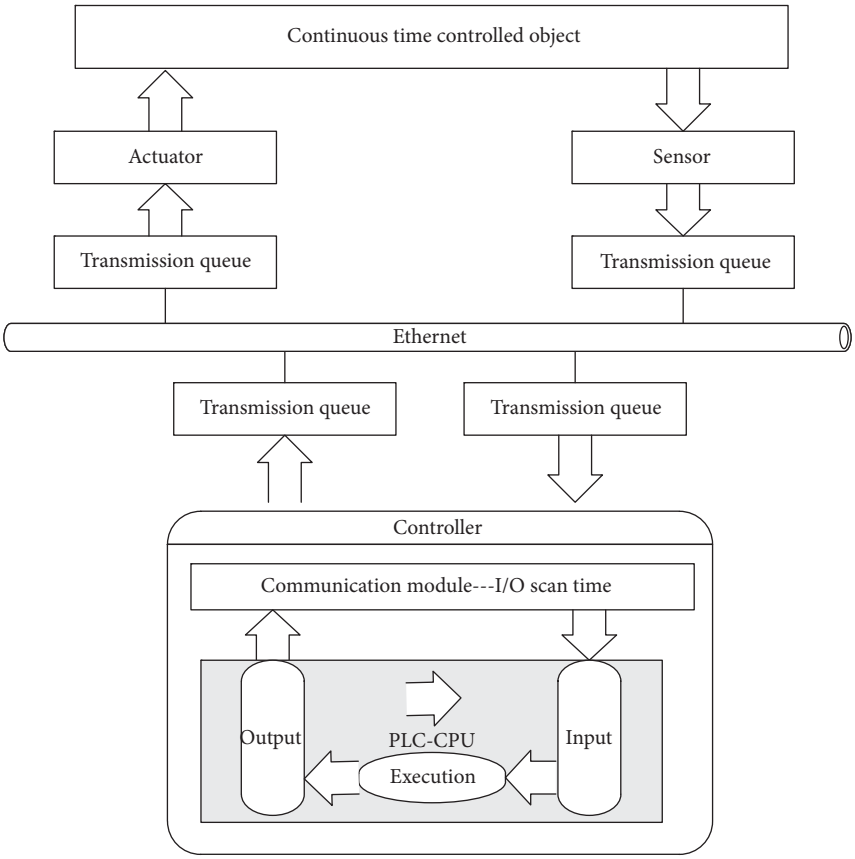

Figure 2: Network control system with PLC being the CPU controller.

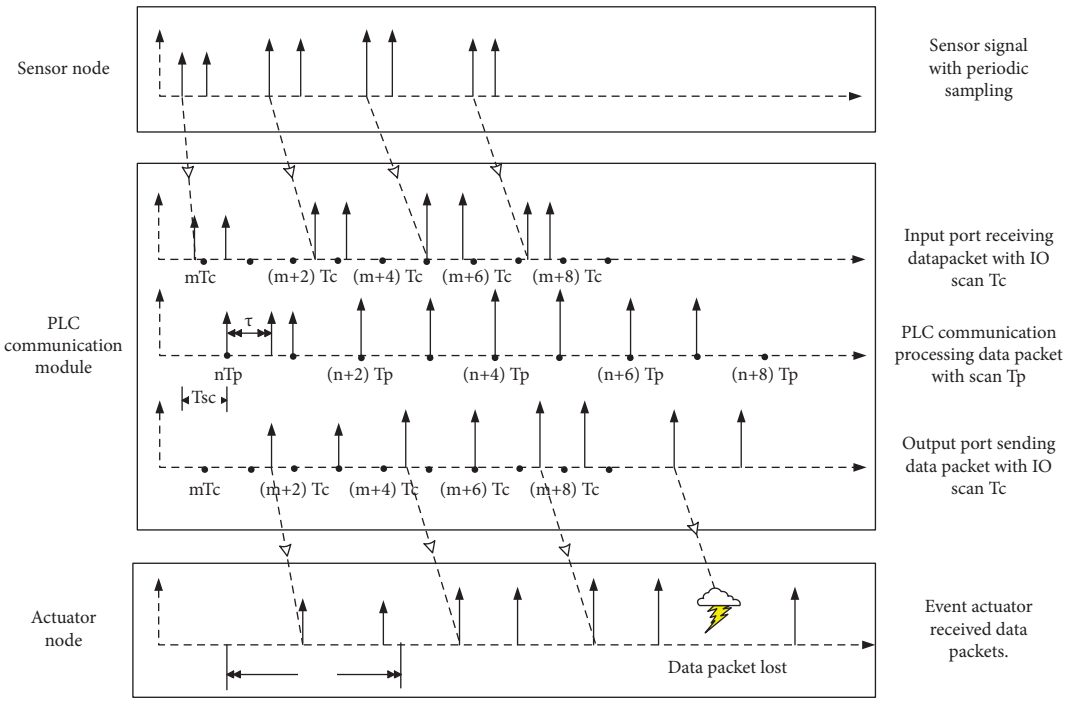

Figure 3: Process of time delay in PLC-CPN.

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 $(s\colon \text{Ten0}, r\colon \text{Ten1}) = (r <= s)$. 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:

$$\text{HCPN} = (S.SN.SA.PN, \text{PT}, \text{PA}, \text{FS}, \text{FT}, \text{PP}). \tag{10}$$

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

Σ 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, $(\#A, p1) = N, T = \text{Net1}, \text{Net2}$ 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, $p1, a \in [A \text{ to Net1}, C \text{ to Store1} \cdot \text{Store1 to } A]$ means when the arc is from A to Net1, C to Store 1, 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 T_h 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 Th 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:

$$\begin{aligned} O_{\text{seq}} &= \{o_1, o_2, \ldots, o_i, \ldots\}, \\ S_{\text{seq}} &= \{s_1, s_2, \ldots, s_i, \ldots\}, \end{aligned} \tag{11}$$

where $O_i \in O = \{o_1, o_2, \ldots, o_N\}$ is the observation state at time k and $S_i \in S = \{s_1, s_2, \ldots, s_N\}$ is the corresponding state at observation O_k.

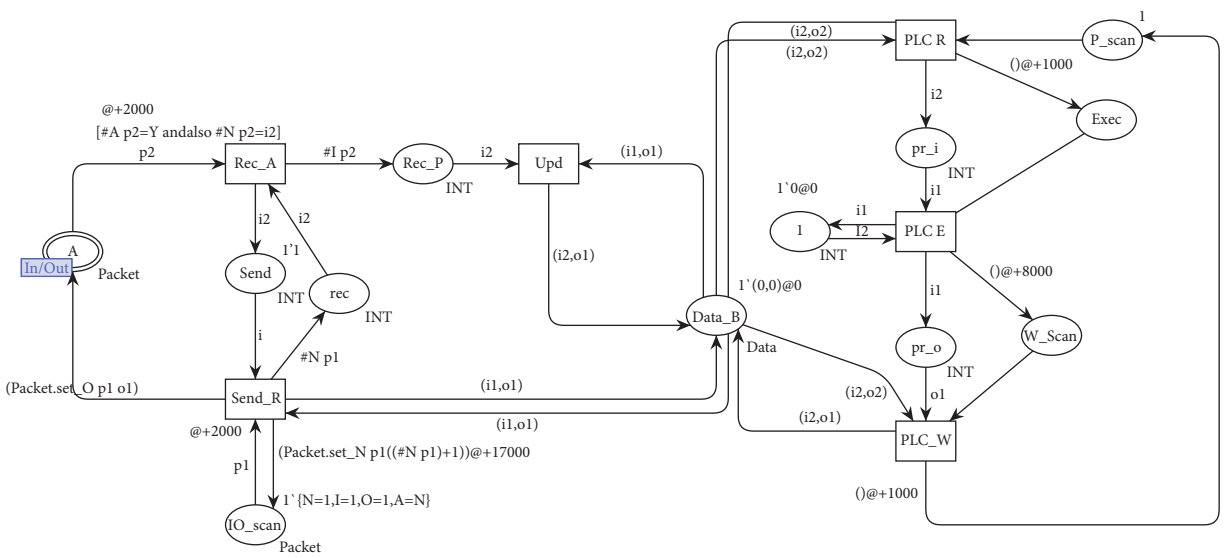

Figure 4: The controller module in the HCPN model for NCSs-PLC.

Table 1: Token definition of the HCPN model for NCSs.

```
colset UNIT = unit timed;
colset BOOL = bool;
colset INT = int timed;
colset TIME = time;
colset STRING = string;
colset N = INT;
colset A = with Y|N;
colset I, O = int;
colset Ten0 = int with 0, . . . , 10;
colset Ten1 = int with 1, . . . , 10;
fun Ok (s: Ten0, r: Ten1) = (r < = s);
colset Packet = record N: N ∗ I: I ∗ O: O ∗ A: A timed;
colset Data = product INT ∗ INT timed;;
var i, i1, i2, o1, o2: INT;
var p1, p2: Packet;;
```

The formal definition of a hidden Markov model (HMM) is as follows:

$$\lambda = \{A, B, \pi_0\}, \tag{12}$$

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

$$\begin{aligned} A &= \{a_{ij}\}, \\ \{a_{ij}\} &= P\{s_{k+1} = s_j | s_k = s_i\}, \end{aligned} \tag{13}$$

B is the observation probability matrix which can be written as

$$\begin{aligned} B &= \{b_{ij}\}, \\ \{b_{ij}\} &= P\{o_{k+1} = o_j | s_k = s_i\}, \end{aligned} \tag{14}$$

and π_0 is the initial state probability which can be written as

$$\begin{aligned} \pi_0 &= \{\pi_i\}, \\ \{\pi_i\} &= P\{s_1 = s_j\}. \end{aligned} \tag{15}$$

5.2. Markov Model Results. By the definition of HMM and the Baum Welth 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

$$\begin{aligned} \delta_t(i) &= \frac{\alpha_t(i)\beta(i)}{P(O|\lambda)}, \\ P(O|\lambda) &= \sum_{i=1}^{N} \alpha_t(i)\beta(i), \end{aligned} \tag{16}$$

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

$$\alpha_t(i) = P(o_1 \ o_2 \ o_3 \ \ldots \ o_{t-1} \ o_t \ , q_t = S_i | \lambda), \tag{17}$$

and $\beta_t(i)$ is called backward probability, where

$$\beta_t(i) = P(o_{t+1} \ o_{t+2} \ o_{t+3} \ \ldots \ o_{T-1} \ o_T \ , q_t = S_i | \lambda), \tag{18}$$

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

$$\begin{aligned} \phi_t(i, j) &= P(q_t = S_i, q_{t+1} = S_j | O, \lambda), \\ \phi_t(i, j) &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)}, \\ P(O|\lambda) &= \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j). \end{aligned} \tag{19}$$

Step 3 : the formula between $\delta_t(i)$ and $\phi_t(i, j)$ is

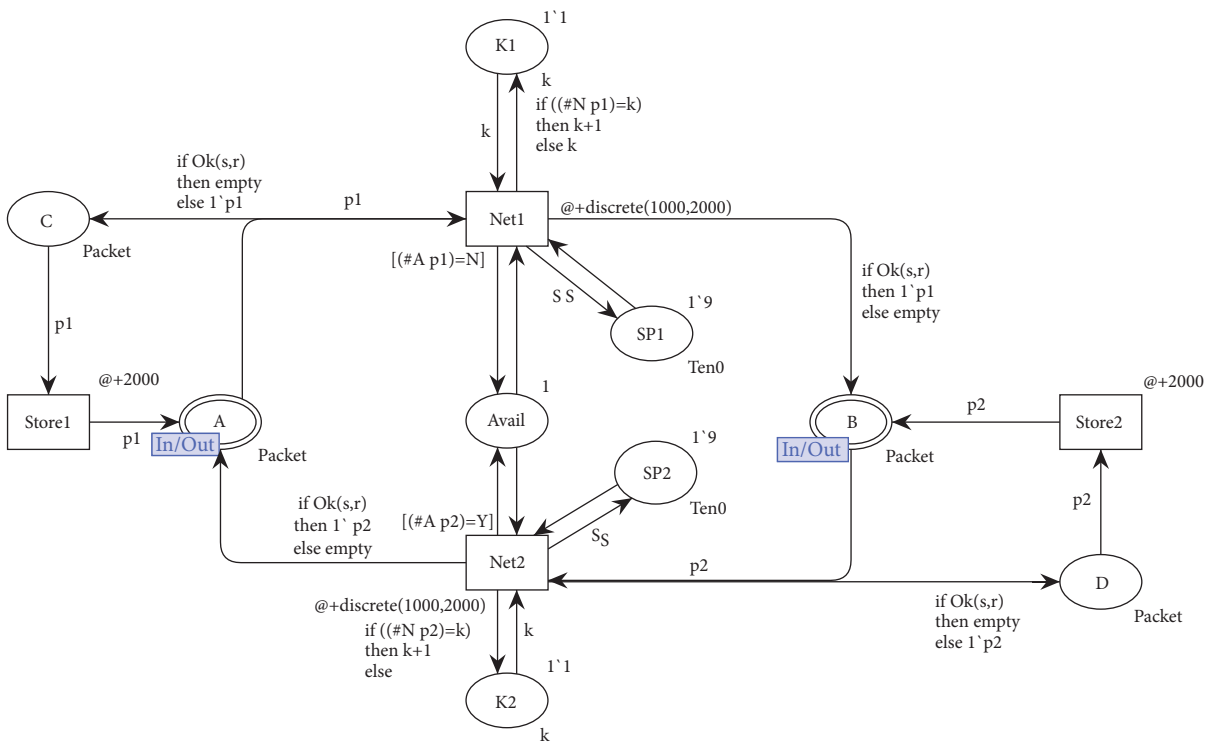

Figure 5: The Ethernet model for NCSs.

$$\delta_t(i) = \sum_{j=1}^{N} \phi_t(i, j). \tag{20}$$

Step 4 : find the corresponding expected values of the above two variables $\sum_{t=1}^{T-1} \delta_t(i)$ and $\sum_{t=1}^{T-1} \phi_t(i, j)$.

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

$$\begin{aligned} \hat{a}_{ij} &= \frac{\sum_{t=1}^{T-1} \phi_t(i, j)}{\sum_{t=1}^{T-1} \delta_t(i)}, \\ \hat{b}_j(k) &= \frac{\sum_{t=1}^{T-1} \phi_t(i, j)}{\sum_{t=1}^{T} \delta_t(i)}. \end{aligned} \tag{21}$$

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:

$$\begin{aligned} Tsc &= [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17], \\ Tca &= [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 29]. \end{aligned} \tag{22}$$

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

The time delay Tsc can be written as

$$Tsc = \begin{cases} Tsc_{\text{low}}, & s_k = s_1, \\ Tsc_{\text{high}}, & s_k = s_2. \end{cases} \tag{23}$$

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 $Tca_{\text{low}} = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19]$ and the other category is high delay Tca_{high}, and $Tca_{\text{high}} = [20, 21, 22, 23, 24, 25, 26, 27, 29]$. The time delay Tca can be written as

$$Tca = \begin{cases} Tca_{\text{low}}, & s_k = s_1, \\ Tca_{\text{high}}, & s_k = s_2. \end{cases} \tag{24}$$

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 $\lambda_{sc} = \{A_{sc}, B_{sc}, \pi_{0_{SC}}\}$ with 2-state and 15-observations, and the matrix A_{sc}, B_{sc} can be calculated by Matlab:

$$\Sigma = [UNIT, BOOL, INT, TIME, STRING, Packet, Data, Ten0, Ten1, N, A, I, O]$$

$$P = [C, A, K1, K2, SP1, SP2, avail, B, D] \quad T = [Store1, Store2, Net1, Net2]$$

$$A = \begin{cases} CtoStore1 & Store1toA & AtoNet1 & Net1toC \\ BtoNet2 & Net2toD & DtoStore2 & Store2toB \\ Net1toB & Net2toA & K1toNet1toK1 & Net1toSP1toNet1 \\ Net2toSP2toNet2 & K2toNet2toK1 & Net1toAvailtoNet1 & Net2toAvailtoNet2 \end{cases}$$

$$N = (Source\ dest) \qquad G = \begin{cases} (\#A, p1) = \mathrm{N} & T = Net1, Net2 \\ true & otherwise \end{cases}$$

$$C = \begin{cases} Packet & A, B, C, D \\ INT & K1, K2 \\ Ten0 & SP1, SP2 \\ UNIT & avail \end{cases} \qquad I(p) = \begin{cases} 1 & p = K1, K2 \\ 9 & P = SP1, SP2 \\ () & P = avail \\ \emptyset & other \end{cases}$$

$$E(a) = \begin{cases} p0 & a \in [AtoNet1, CtoStore1.Store1toA] \\ p2 & a \in [BtoNet2, DtoStore2.Store2toB] \\ k & a \in [K1toNet1, K2toNet2] \\ s & a \in [Net1toSP1, SP1tpNet1, Net2toSP2, SP2tpNet2] \\ \text{if } OK(s, r) \text{ then else } empty \text{ else } 1`P1 & a = Net1toC \\ \text{if } OK(s, r) \text{ then else } empty \text{ else } 1`P2 & a = Net2toD \\ \text{if } OK(s, r) \text{ then } 1`P1 \text{ else } empty & a = Net1toB \\ \text{if } OK(s, r) \text{ then } 1`P2 \text{ else } empty & a = Net2toA \\ \text{if } ((\#NP1) = K) \text{ then } k + 1 \text{ else } k & a = Net1toK1 \\ \text{if } ((\#NP2) = K) \text{ then } k + 1 \text{ else } k & a = Net2toK2 \\ () & other \end{cases}$$

Figure 6: Mathematical description for HCPN.

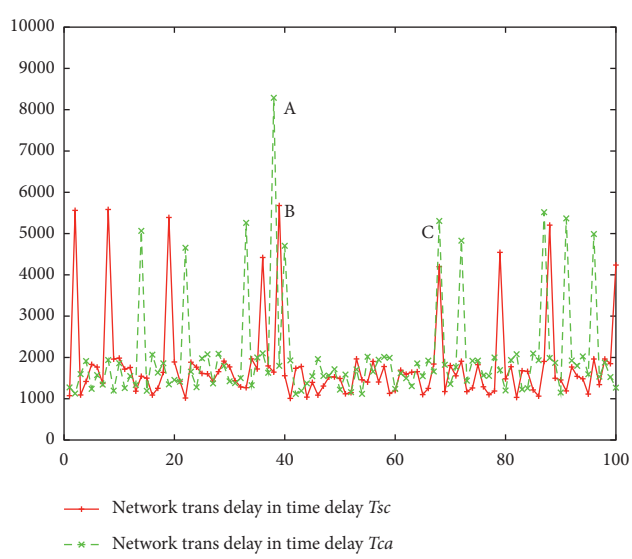

Figure 7: Network transmission latency in the HCPN model for NCSs.

Table 2: Main parameters.

Module	Time	
PLC module	IO scan cycle	17000
	Read	2000
	Execute	8000
	Write	2000
Ethernet module	2000	
Remote analog input/output	IO-read	1000
	IO-write	1000

$$A_{sc} = \begin{bmatrix} 0.28 & 0.72 \\ 0.84 & 0.15 \end{bmatrix},$$
$$B_{sc} = [B_{sc1} \quad B_{sc2}],$$
$$B_{sc1} = \begin{bmatrix} 0.15 & 0.10 & 0.15 & 0.15 & 0.15 & 0.13 & 0.17 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$
$$B_{sc2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.20 & 0.20 & 0.18 & 0.20 & 0.12 & 0.03 & 0.04 & 0.03 \end{bmatrix}. \tag{25}$$

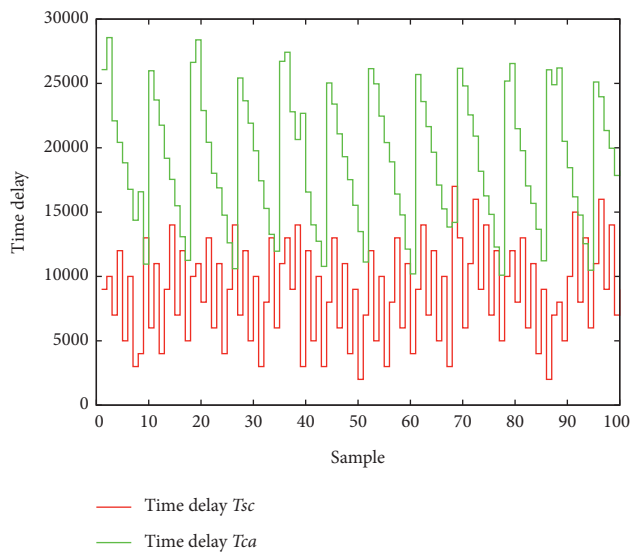

Figure 8: Tsc time delay and Tca time delay.

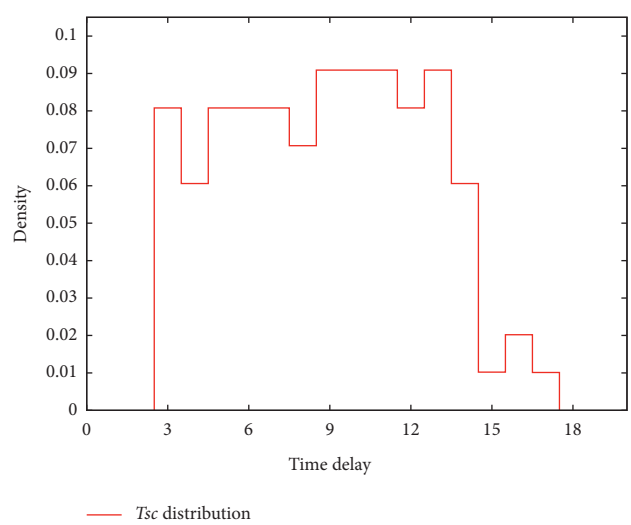

Figure 9: Distribution of *Tsc* in the HCPN model.

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

$$A_{ca} = \begin{bmatrix} 0.78 & 0.22 \\ 0.25 & 0.75 \end{bmatrix},$$
$$B_{ca} = [\, B_{ca1} \quad B_{ca2} \,],$$
$$B_{ca1} = \begin{bmatrix} 0.06 & 0.10 & 0.08 & 0.11 & 0.11 & 0.11 & 0.08 & 0.19 & 0.10 & 0.06 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$
$$B_{ca2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.19 & 0.12 & 0.10 & 0.15 & 0.06 & 0.15 & 0.15 & 0.06 & 0.02 \end{bmatrix}. \tag{26}$$

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

$$A = \{A_{sc}, A_{ca}\},$$
$$B = \{B_{sc}, B_{ca}\}. \tag{27}$$

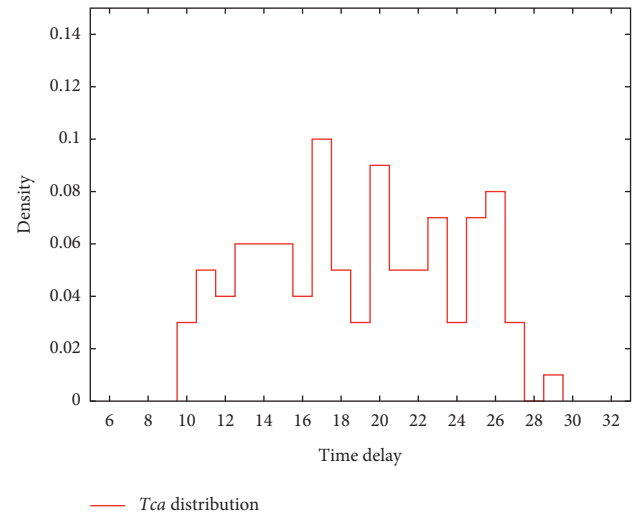

Figure 10: Distribution of *Tca* in the HCPN model.

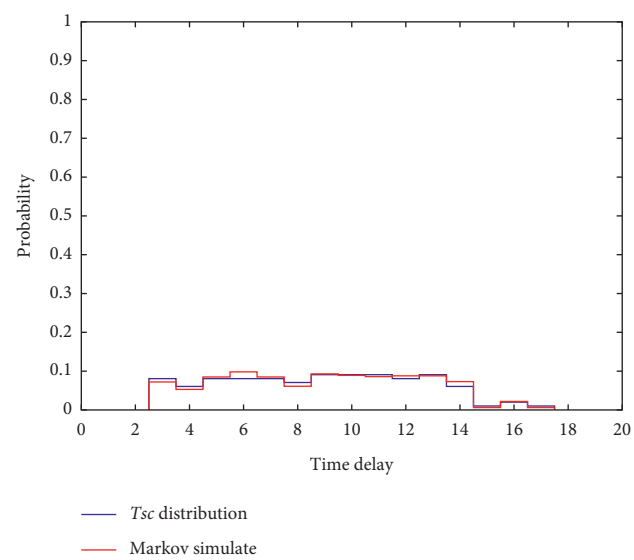

Figure 11: The comparison of *Tsc* probability distribution between the HCPN model and the Markov model.

Using the data from those two matrices, the stationary distribution of the Markov model λ_{sc} and λ_{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

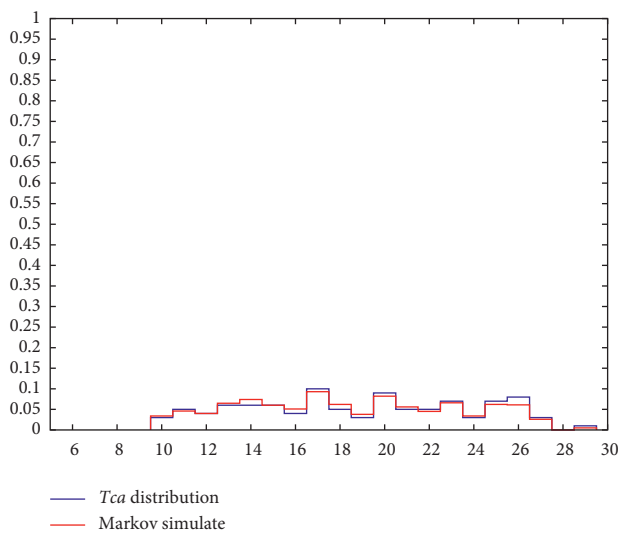

Figure 12: The comparison of *Tca* probability distribution between the HCPN model and the Markov model.

that building a Markov modeling of NCSs is feasible and time delays conform to Markov lain.

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

[1] H. Li, C. Wu, S. Yin, and H.-K. Lam, "Observer-based fuzzy control for nonlinear networked systems under unmeasurable premise variables," *IEEE Transactions on Fuzzy Systems*, vol. 24, pp. 1233–1245, 2015.

[2] H. Li, Z. Chen, L. Wu, H. K. Lam, and H. Du, "Event-triggered fault detection of nonlinear networked systems," *IEEE Transactions on Cybernetics*, vol. 47, pp. 1041–1052, 2016.

[3] S. Yin, X. Zhu, J. Qiu, and H. Gao, "State estimation in nonlinear system using sequential evolutionary filter," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 6, pp. 3786–3794, 2016.

[4] H. Gao and X. Li, "Filtering for discrete-time state-delayed systems with finite frequency specifications," *IEEE Transactions on Automatic Control*, vol. 56, no. 12, pp. 2935–2941, 2011.

[5] H. Li, Y. Gao, P. Shi, and H.-K. Lam, "Observer-based fault detection for nonlinear systems with sensor fault and limited communication capacity," *IEEE Transactions on Automatic Control*, vol. 61, pp. 2745–2751, 2015.

[6] H. Li, C. Wu, L. Wu, H.-K. Lam, and Y. Gao, "Filtering of interval type-2 fuzzy systems with intermittent measurements," *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 668–678, 2016.

[7] J. Zhang, C. Peng, D. Du, and M. Zheng, "Adaptive event-triggered communication scheme for networked control systems with randomly occurring nonlinearities and uncertainties," *Neurocomputing*, vol. 174, pp. 475–482, 2016.

[8] W. Wang, D. Nešić, and R. Postoyan, "Emulation-based stabilization of networked control systems implemented on flexray," *Automatica*, vol. 59, pp. 73–83, 2015.

[9] B. Liu, B. Qiu, Y. Cui, and J. Sun, "Fault-tolerant H ∞ control for networked control systems with randomly occurring missing measurements," *Neurocomputing*, vol. 175, pp. 459–465, 2016.

[10] G. P. Liu, "Design and analysis of networked non-linear predictive control systems," *IET Control Theory & Applications*, vol. 9, no. 11, pp. 1740–1745, 2015.

[11] X. Liu, X. Yu, G. Ma, and H. Xi, "On sliding mode control for networked control systems with semi-markovian switching and random sensor delays," *Information Sciences*, vol. 337-338, pp. 44–58, 2016.

[12] L. Schenato, "Optimal estimation in networked control systems subject to random delay and packet loss," in *Proceedings of the 45th IEEE Conference on Decision and Control, 2006*, pp. 5615–5620, IEEE, San Diego, CA, USA, December 2006.

[13] X.-S. Zhan, J. Wu, T. Jiang, and X.-W. Jiang, "Optimal performance of networked control systems under the packet dropouts and channel noise," *ISA Transactions*, vol. 58, pp. 214–221, 2015.

[14] D. Goswami, R. Schneider, and S. Chakraborty, "Re-engineering cyber-physical control applications for hybrid communication protocols," in *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1–6, IEEE, Paris, France, February 2011.

[15] M. Razeghi-Jahromi and A. Seyedi, "Stabilization of networked control systems with sparse observer-controller networks," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1686–1691, 2015.

[16] K. Liu, E. Fridman, and K. H. Johansson, "Dynamic quantization of uncertain linear networked control systems," *Automatica*, vol. 59, pp. 248–255, 2015.

[17] F. Cacace, F. Conte, A. Germani, and G. Palombo, "Optimal control of linear systems with large and variable input delays," *Systems & Control Letters*, vol. 89, pp. 1–7, 2016.

[18] L. Cao, H. Ren, W. Meng, H. Li, and R. Lu, "Distributed event triggering control for six-rotor UAV systems with asymmetric time-varying output constraints," *Science China Information Sciences*, vol. 64, no. 7, 2021.

[19] L. Qiu, Y. Shi, J. Pan, B. Xu, and H. Li, "Robust control for a networked direct-drive linear motion control system: design and experiments," *Information Sciences*, vol. 370-371, pp. 725–742, 2016.

[20] H. Arneson, N. Dousse, and C. Langbort, "A linear programming approach to routing control in networks of constrained nonlinear positive systems with concave flow rates," *Automatica*, vol. 68, pp. 357–368, 2016.

[21] H. Chen, J. Gao, T. Shi, and R. Lu, "H ∞ control for networked control systems with time delay, data packet dropout and disorder," *Neurocomputing*, vol. 179, pp. 211–218, 2016.

[22] X.-M. Sun, K.-Z. Liu, C. Wen, and W. Wang, "Predictive control of nonlinear continuous networked control systems with large time-varying transmission delays and transmission protocols," *Automatica*, vol. 64, pp. 76–85, 2016.

[23] S. Di Cairano, U. V. Kalabić, and I. V. Kolmanovsky, "Reference governor for network control systems subject to variable time-delay," *Automatica*, vol. 62, pp. 77–86, 2015.

[24] J. L. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice Hall PTR, Hoboken, NJ, USA, 1981.

[25] R. S. Cost, Y. Chen, T. Finin, Y. Labrou, and Y. Peng, "Using colored petri nets for conversation modeling," *Issues in Agent Communication*, vol. 19, pp. 178–192, 2000.

[26] W. Chun-jian, L. Yong-zhi, and X. Fan, "An improved modeling method based on colored petri net," *Physics Procedia*, vol. 24, pp. 1128–1132, 2012.

[27] N. Gharbi, C. Dutheillet, and M. Ioualalen, "Colored stochastic petri nets for modelling and analysis of multiclass retrial systems," *Mathematical and Computer Modelling*, vol. 49, no. 7, pp. 1436–1448, 2009.

[28] R. Bastide and E. Barboni, "Software components: a formal semantics based on coloured petri nets," *Electronic Notes in Theoretical Computer Science*, vol. 160, pp. 57–73, 2006.

[29] M. Jamro, D. Rzonca, and W. Rząsa, "Testing communication tasks in distributed control systems with sysml and timed colored petri nets model," *Computers in Industry*, vol. 71, pp. 77–87, 2015.

[30] H. Huang, H. Kirchner, Secure Interoperation in Heterogeneous Systems Based on Colored Petri Nets, Working Paper or Preprint (2009). URLhttps://hal.inria.fr/inria-00396952.

[31] A. Ghanaim and G. Frey, "Component based colored petri net model for ethernet based networked control systems," in *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 1100–1103, Berlin, Germany, September 2008.

[32] A. Ghanaim, G. A. Borges, and G. Frey, "Estimating delays in networked control systems using colored petri nets and Markov chain models," in *Proceedings of the IEEE Conference on Emerging Technologies & Factory Automation*, pp. 1–6, Berlin, Germany, September 2009.

[33] P. Huber, K. Jensen, and R. M. Shapiro, "Hierarchies in coloured petri nets," in *Advances in Petri Nets 1990*, pp. 313–341, Springer, Berlin, Germany, 1989.

[34] M. Elkoutbi and R. K. Keller, "Modeling interactive systems with hierarchical colored petri nets," in *Advanced Simulation Technologies Conference*, pp. 432–437, Citeseer, Princeton, NJ, USA, 1998.

[35] Y. Yang, Q. Tan, and Y. Xiao, "Verifying web services composition based on hierarchical colored petri nets," in *Proceedings of the First International Workshop on Interoperability of Heterogeneous Information Systems*, pp. 47–54, ACM, Bremen, Germany, November 2005.

[36] A. Ghanaim and G. Frey, "Markov modeling of delays in networked automation and control systems using colored petri net models simulation," in *Proceedings of the 18th IFAC World Congress, Anonymous IFAC Secretariat*, pp. 2731–2736, Milano, Italy, August 2011.

[37] A. Ghanaim and G. Frey, "Modeling and control of closed-loop networked plc-systems," in *Proceedings of the American Control Conference (ACC)*, pp. 502–508, Milwaukee, Wisconsin, USA, June 2011.

[38] G. Mustafa and T. Chen, "H filtering for nonuniformly sampled systems: a markovian jump systems approach," *Systems & Control Letters*, vol. 60, no. 10, pp. 871–876, 2011.

[39] Á. Cuenca, U. Ojha, J. Salt, and M.-Y. Chow, "A non-uniform multi-rate control strategy for a Markov chain-driven networked control system," *Information Sciences*, vol. 321, pp. 31–47, 2015.

[40] Y. Wang, S. Xu, and S. Zhang, "Fault detection for a class of nonlinear networked control systems with markov sensors assignment and random transmission delays," *Journal of the Franklin Institute*, vol. 351, no. 10, pp. 4653–4671, 2014.

[41] H. Gao, Z. Fei, J. Lam, and B. Du, "Further results on exponential estimates of markovian jump," *IEEE Transactions on Automatic Control*, vol. 56, no. 1, p. 223, 2011.

[42] K. Jensen, "Coloured petri nets," *EATCS Monographs on Theoretical Computer Science*, vol. 1.