2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Modeling and Control of Closed-Loop Networked PLC-systems

Abouelabbas Ghanaim, Georg Frey
Saarland University, Chair of Automation
D-66123, Saarbrücken, Germany
{abouelabbas.ghanaim, georg.frey}@aut.uni-saarland.de

Abstract

This paper presents a novel approach for modeling and control of Programmable Logic Controller based Networked Control Systems (PLC-based NCS). The approach allows control design of NCS using formal discrete modeling and simulation. This involves three steps. The first step is to design a structure-conserving hierarchal timed model for the whole PLC-based NCS using Colored Petri Nets (CPNs). The aim is to generate by simulation extensive sampled time delay data records that are used to analyze and model delay behavior. The second step is the bridge between formal modeling and simulation and NCS control design. It uses Direct and/or Hidden Markov Models (DMM, HMM) to describe mathematical models of NCS delays. The paper introduces a new concept of Mutual Markov modeling to analyze the interaction between the two induced delays, namely, the sensor-to-controller time delay τ_{sc} and the controller-to-actuator time delay τ_{ca}. In the third step, the proposed mutual Markov models are used to design a single mode-dependent state feedback controller using jump linear systems (JLS) approach. The control design solves a stability condition using Linear/Bilinear Matrix Inequalities (LMI/BMI). A numerical example is provided to demonstrate the proposed procedure.

1. Introduction

Networked Control Systems (NCS) are a type of distributed control systems where sensors, actuators and controllers are interconnected by real-time communication networks. Fig. 1 shows a schematic representation of a typical NCS with Programmable Logic Controller (PLC) as a system controller. Several advantages of these systems include: reduced systems wiring, increased system agility and ease of system diagnosis and maintenance.

Depending on the devices sharing the network and the volume of information interchanged, the sender waits a variable time until the medium is granted to it. The stochastic nature of the shared resource occupation means random access times. These times will be denoted as sensor-to-controller delay (τ_{sc}) which is the random access time in the sensor-controller (SC) link and controller-to-actuator delay (τ_{ca}) in the controller-actuator (CA) one. These delays are sources of potential instability. In addition, especially in time-driven platforms such as PLC-based NCS, as the controller-plant communication uses a non-exclusive medium; it is difficult for the control device to precisely determine the sampling and actuation instants. The lack of synchronization between controller and plant causes a significant worsening of the system response [5].

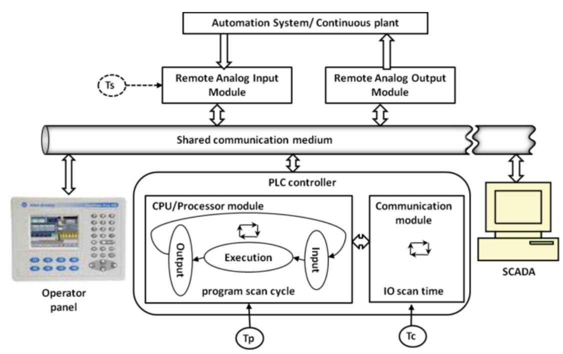

Fig. 1. Block diagram of PLC-based NCS.

The term Networked Control Systems (NCS) in recent literature refers to the interdisciplinary research area, combining both network and control theory, in order to guarantee the stability and performance of an NCS [2], [16]. In contrast Networked Automation System (NAS) combines network and formal modeling tools to guarantee certain time performances for time critical automation tasks [7],[10], and [13]. NAS-based NCS or without loss of generality PLC-based NCS perform not only open loop automation tasks but also closed loop control tasks for time critical tasks such as position and motion control. The main difference between open loop NAS and closed loop NCS lies in the fact that in NAS the occurrence of an event is assumed to happen at random time and the system response (delay) to such an event is determined independently of other events while in NCS consecutive delays of sampled sensor signals (*Ts* in the dashed circle connected to the remote analog input unit in Fig. 1) are of interest.

The NAS/NCS system controller as shown in the figure is composed of two modules: The first module is the CPU/processor module which executes the control tasks in time-based (cyclical) execution with a program scan time T_p. It reads input values from controller input buffer, executes the user control program, and finally writes control signals to controller output buffer. The second module is the input/output communication module that scans remote input/output units with T_c input/output scan time. The two scan cycles are not synchronized and the modules exchange data through an internal backplane bus.

In such PLC-based NCS, there is a time driven nature of all activities in the system such as sampling process signals, input/output scanning and execution of control program. Due to this nature, it is expected that network induced delays exhibit special periodic characteristics which is the main motivation to address this sub-class of NCSs.

Fig. 2 shows the three-step PLC-based NCS control design approach using formal modeling and simulation It is a simple place transition diagram showing the sequence of steps and the relations among them. First step uses Colored Petri Nets (CPN) for modeling and simulation of NCS setup. Simulation of CPN models of PLC-based NCS generate records of delays patterns which facilitate the process of modeling delays behavior.

The formal simulation is proposed as an alternative to experimental setups of delay measuring. Constructing an experimental setup to measure delays in NCS is a very complex and error prone process, especially, in industrial controller's case, where the controller may have neither sufficient memory space nor an efficient logging software tools. In addition, formal based modeling and simulation provides an efficient simulation environment from the view point of simulation speed in comparison to general purpose simulation software tools [13].

In the second step, the data generated from CPN model is used to identify delay Markov models to obtain a compact mathematical representation of delays behavior that can be used for the control design step. In this view, the second step can be considered as a bridge linking the two research areas: NAS formal modeling and simulation from one side and NCS control design from the other side. In this step, the paper introduces a new concept of Mutual Markov modeling to analyze interaction between the types of induced delays, namely, the sensor-to-controller time delay τ_{sc} and the controller-to-actuator time delay τ_{ca} using direct and/or Hidden Markov Models (DMM, HMM).

In the third step, the obtained mutual Markov transition matrices are used to design a state feedback controller for the NCS system using jump linear system (JLS). JLS control design is an offline controller design algorithm i.e. the controller parameters are calculated offline and stored in memory for online operation. The most recent control approach is the two mode-dependent control which designs a switching control according to the assumption of availability of both of the most recent delays $\tau_{sc}(k)$, $\tau_{ca}(k-1)$. In the single mode-dependent design method the controller assumes only that delays from sensor-to-controller $\tau_{sc}(k)$ are known at the time of calculation. In this paper, we propose a modified version of a single mode-dependent control that combines the benefits of the two control designs by using only τ_{sc} delay information and mutual Markov models to compensate for lack of τ_{ca} delay information.

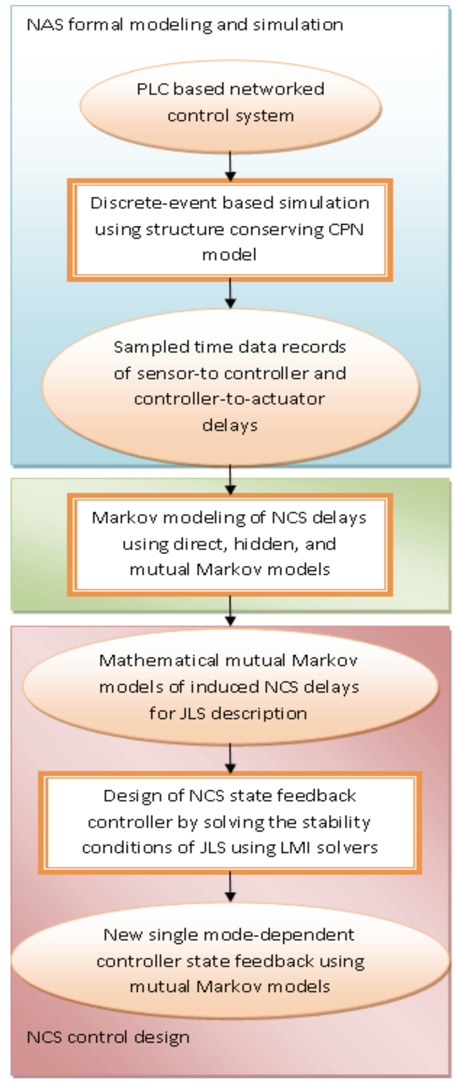

Fig. 2. Steps of NCS control design based formal modelling and simulation

This paper is organized as follows: The next section describes the PLC-based NCS induced delays time analysis. Section 3 presents the CPN modeling and simulation of PLC-based NCS. Section 4 presents the mutual Markov modeling approach of PLC-based NCS. Section 5 presents the jump linear systems control of PLC-based NCS and demonstrates results using a numerical example. The concluding section comments on results and summarizes main ideas.

2. Time analysis of PLC-based NCSs

Most NCS literature considers the setup with assumptions a) time driven sensors that sample the plant outputs periodically at sampling instances; b) an event driven controller which calculates the control signal as soon as the sensor data arrives; c) event driven actuators, which means that the plant inputs are changed as soon as the control signals become available [3], [14], [16]. In this paper, we also use assumptions a) and c) which agree with the sampled nature of the NCS systems, but assumption b) is modified to time-driven execution platforms, which constitutes an important category of industrial controllers such as PLCs.

Fig. 3 shows a detailed analysis of delay times in PLC-based NCS. The sensor and actuator nodes are of time and event nature respectively as previously assumed and they are connected to their digital/analog input/output (I/O) units. These components induce a small processing delay. The conversion time in the Analog Input (AI) unit has to be especially considered, since it delays sampled sensor values. The upper part of the figure shows four sensor samples (with different amplitudes and different colors) and the same four samples after being stored in the remote analog input unit with a processing delay(①).

The middle part shows the PLC controller. It is composed of two interacting scan cycle times *Tp*, *Tc*. In the middle is the CPU program scan cycle *Tp*, which executes the control program periodically. The PLC communication module - with scan cycle *Tc* - scans remote digital/analog IO units. The communication scan cycle surrounds the program scan cycle from senor data input side and from the control value output side, so that it appears twice in the figure. The communication module sends request packets (②) to remote IO units with new outputs and receives new sensor inputs via acknowledgment packets (③). In the general case, when the sensor and actuator are not connected to the same node, the request times are different as shown in the figure. The sampled sensor values appear as soon as acknowledgement packets are received and may be repeated depending on the sampling and IO scan times (see for example the three samples of the same amplitude). The control values appear after a computational delay τ_c in the communication module buffer. The controller calculation delay by definition is included in the delay from controller-to-sensor. The lower part of the figure shows the remote analog output module with the control values sent during IO scanning requests(④) and the actuator with the resulting delayed samples. The figure shows delays τ_{sc}, τ_{ca} and the (response time) total control delay τ_{tc}. It clearly shown in the figure that delays depend on the interaction among the three periodic times: sampling time T_s, program scan cycle T_p, and the input/output scan cycle T_c [5].

3. CPN modeling and simulation

Colored Petri nets (CPN) are high-level Petri nets with graphical form and well-known semantic which allows for formal analysis and fast simulation which is suitable to model concurrent and resource sharing systems as discussed in [11]. Applications of CPN to modeling and simulation of NASs and communication channels in estimation of response time in open loop schemes can be found for in [7], and [10]. CPN models of NCS can be built based on object oriented concepts as in [10] or in a structure-conserving component-based way as introduced in previous work in [15].

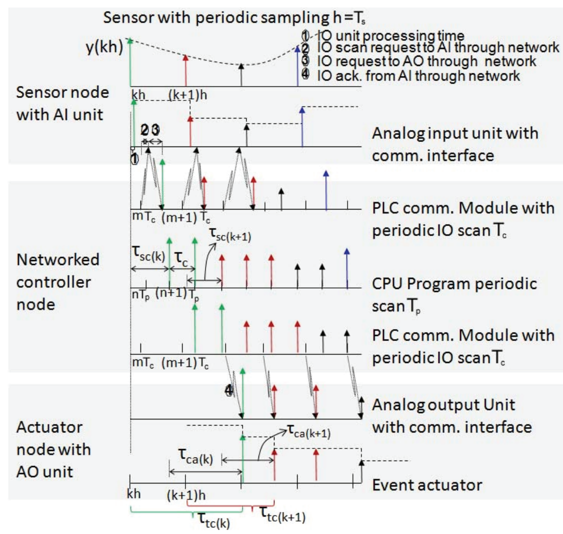

Fig. 3. Delay analysis in PLC based NCS

CPN models

To build the models, CPNtools developed by Kurt Jensen at University of Aarhus in Denmark 1982 is used with its existing simulation and monitoring features. The model explores the main features of PLC-based execution platforms, such as the interaction between client/server for input output scanning and the cyclical execution of control algorithms. The model is simulated using a dummy sampled signal generator. Sensor (input), controller (PLC), and actuator (output) signals are recorded for delays calculation.

In the proposed CPN models, a library of control and network components is constructed. The library includes: PLC with communication module, Ethernet switch, remote analog input/output unit with communication interface, and sample process. Each of these components is built in a separate window called a "Page". In addition, there is a main page for the overall system outline. The main page contains instances of the required components. The required control system configuration can be built by "cloning" the required instances, and connecting them in the same structure as the system to be modeled. Consequently, for a user of the library there is no need to access component pages.

In order to make the paper as a self-contained, we consider here the description of the system page of the model. A detailed description of the internal structure of each component is available in previous works [12], and [15]. Fig. 4 shows the main system layout of the CPN model for a typical PLC-based NCS system. Separate analog input/output units are used for sensor and actuator signals to model the general case of independent sensor/actuator communication nodes. The double line rectangles represent compound transitions with instance name and instance class (small single line rectangles under each transition). Ovals represent places to pass parameters and store data. The PLC component has four parameters: PLC address, CPU scan time, input/output scan time, and remote input/output scan list. The Ethernet switch (SW) has one parameter which is a list assigning each MAC address to a switch port number. The remote analog input/output unit (I/O) has one parameter, which is the address of the unit. The sample process (Process) has one parameter, which is the sample time of the process. The time resolution used here is 1µs, which means that 10ms scan cycle is entered as an integer of value 10000.

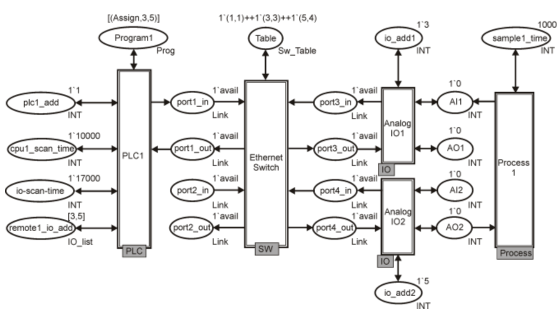

Fig. 4. CPN template with PLC, Ethernet switch, I/O-module, and sample process.

CPN models simulation

By definition, τ_{sc} is the time elapsed between reading a new data sample from the sensor (i.e. at the process) and the start of the execution phase of the controller using this sample. τ_{ca} is defined as the time elapsed between the start of the execution phase of the controller and the arrival of the new control output at the output port of the analog input/output unit, i.e. at the process. The data samples are stored with the execution time of each transition in an array of 10000 samples for processing.

Fig. 5 shows the resulting delay sequences for sampling times *Ts*=1ms, 40ms. In the case of 1ms (extremely small sampling time), delays of arrived samples to the analog output module are calculated whereas other missed samples are considered as *vacant sampling*. Delays can be represented directly against time or represented against the number of successfully transmitted samples (as shown in fig.). The fig. shows a sort of periodicity in the delay patterns. A correlation between τ_{sc} and τ_{ca} is also visible. These impressions will be justified using Markov modeling of delays independently for each delay and a composite model for both delays. Due to space limitation, just one sampling time of 1ms - which is the smallest sampling time - will be used in section 5 to model delays using Markov modeling.

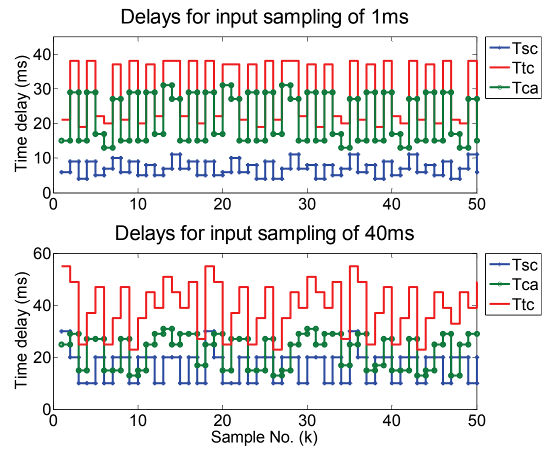

Fig. 5. Sensor-to-controller delays, controller-to-actuator delays, and total control delays for 1ms, 40ms input sampling for the system shown in Fig. 4.

4. Markov Modeling

In this paper, an approach is proposed for modeling induced network delays separately and to address the correlation between the two types of delays using mutual Markov models. We calculate the probability density function (*pdf*) of the CPN-model generated data, and then depending on the number of delay values and the *pdf* diagram, we select a direct or hidden Markov model using the number of distinct areas in the *pdf* diagram as the number of hidden states in the Markov model.

Proposed Markov modeling Procedure

A Markov model is a finite state model that describes a probability distribution over a number of possible sequences. The model states might correspond to network load states that lead to network induced delays. Each state emits observation and the states are connected by state transmission probabilities [4].

Given a sequence of observations O_{seq}, it's assumed a hidden sequence of states S_{seq} corresponding to this observations sequence as shown below:

$$O_{seq} = o_{(1)}, o_{(2)}, \ldots, \quad o_{(k-1)}, o_{(k)}, o_{(k+1)}, \ldots$$

$$S_{seq} = s_{(1)}, s_{(2)}, \ldots, \quad s_{(k-1)}, s_{(k)}, s_{(k+1)}, \ldots$$

Where O_{seq}, S_{seq} are the observation and its corresponding state respectively at sample time *k*,

$o_{(k)} \in O = \{o_1, o_2, ..., o_N\}$ a discrete observation set, and $s_{(k)} \in S = \{s_1, s_2, ..., s_M\}$ a discrete state set. The Markov model λ for the system can be written as: $\lambda = (A, B, \pi_0)$, with the state transition probability matrix $A = \{a_{ij}\} = \Pr(s_{(k+1)} = s_j \mid s_{(k)} = s_i)$, the state observations prob. matrix $B = \{b_{ij}\} = \Pr(o_{(k)} = o_j \mid s_{(k)} = s_i)$, and the initial state prob. vector $\pi_0 = \{\pi_i\} = \Pr(s_{(1)} = s_i)$.

Markov modeling results

The proposed approach is demonstrated using delay sequences shown in Fig. 5 separately. Fig. 6 shows the probability distribution of delays τ_{sc}, and τ_{ca} in the case of 1ms input sampling. τ_{sc} delays can be classified into two regions: small delays region {4, 5, 6, 7, 8, 9} with high probability and high delays region {10, 11} with small probability. Therefore, it is reasonable to model τ_{sc} with a 2-state, 8-observation HMM λsc model.

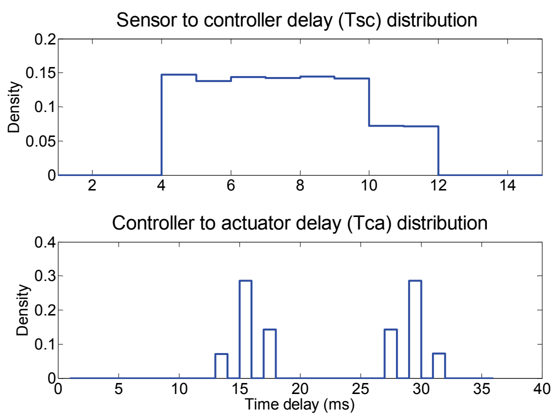

Fig. 6. Probability density distribution for τsc, and τca for 1 ms input sampling.

Using Maximum likelihood (ML) estimate and a Baum-Welch, the HMM λ_{sc} model parameters for the sensor-to-controller delay τ_{sc} are given by:

$$A_{\tau_{sc}} = \begin{bmatrix} 0.83 & 0.17 \\ 1 & 0 \end{bmatrix},$$

$$B_{\tau_{sc}} = \begin{bmatrix} 0.17 & 0.16 & 0.17 & 0.17 & 0.17 & 0.17 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 \end{bmatrix}$$

Where τ_{sc} delay equation can be written as (rect [] denotes a uniform distribution on this interval):

$$\tau_{sc} = \begin{cases} rect[4,9] & s_{(k)} = s_1 \\ rect[10,11] & s_{(k)} = s_2 \end{cases}$$

In a similar way, a 2-state, 6-observations HMM λ_{ca} for the controller-to-actuator delay τ_{ca} can bestimated as:

$$P_{\tau_{ca}} = \begin{bmatrix} 0.14 & 0.86 \\ 0.86 & 14 \end{bmatrix}, \quad Q_{\tau_{ca}} = \begin{bmatrix} 0.14 & 0.57 & 0.29 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.29 & 0.57 & 0.14 \end{bmatrix}$$

$$\tau_{ca} = \begin{cases} \{13,14,15\} & s_{(k)} = s_1 \\ \{27,29,31\} & s_{(k)} = s_2 \end{cases}$$

The important part of the proposed Markov modeling procedure is the ability to model the correlation between the two delays. The idea which is previously mentioned in [1]- which to the best of author's knowledge is not followed in literature - is used to estimate a composite/mutual model for the two delays. The idea is to interleave alternatively the two delay sequences; τ_{sc} τ_{ca} beginning with τ_{sc} to make a single composite/mutual new delay sequence, and to estimate a Markov model for the new sequence using the previous proposed method. As shown in Fig. 7, the number of states of the mutual model will be $M_m=M_{sc}+M_{ca}$ and state transitions will go from a state in the τ_{sc} model to a state in the τ_{ca} model and vice versa without inter-model state transitions. This model is called a 2-state advance by input sample transmission i.e. for each input sample the model evolves two times, one for τ_{sc} delay estimation and the other for τ_{ca} delay estimation. The HHM λ_m parameters are estimated for our case using the same ML and Baum-Welch to:

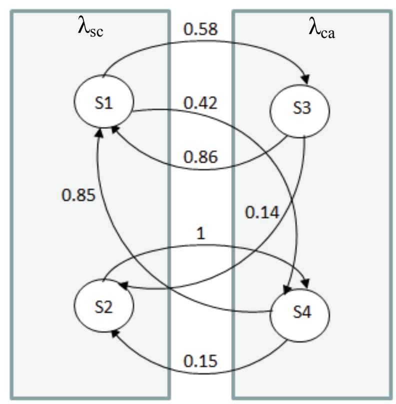

Fig. 7. State diagram of the composite HMM λcom for τsc and τca delays.

$$A_m = \begin{bmatrix} 0 & 0 & 0.58 & 0.42 \\ 0 & 0 & 0 & 1 \\ 0.86 & 0.14 & 0 & 0 \\ 0.85 & 0.15 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & \lambda'_{ca/cs} \\ \lambda'_{sc/ca} & 0 \end{bmatrix}$$

The Mutual model A_m can be decomposed into new separate Markov models $\lambda'_{ca/sc}$, $\lambda'_{sc/ca}$. The new models are interesting especially $\lambda'_{ca/sc}$ in estimating τ_{ca} given τ_{sc}. Since the τ_{ca} delay is unknown at the time of control value calculation, in contrary to the τ_{sc} delay, which could be calculated in the case of time stamped sensor signals.

For a slower input sampling, the same approach is used with exception that the number of delay elements is small, which suggests to use a direct Markov modeling (DMM) for τ_{sc} or τ_{ca}. The mutual model in this case can be composed of two different Markov models i.e. DMM can be used to model τ_{sc} delays and HMM to model τ_{ca} delays or vice versa.

5. Jump Linear Model of PLC-based NCSs

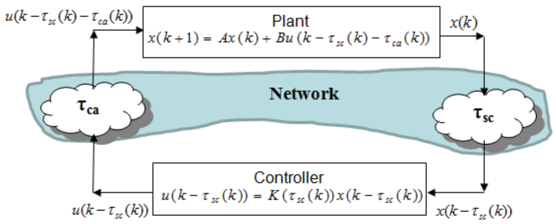

Fig. 8 Control system setup

First consider the control system setup in. Fig. 8. The discrete-time linear time invariant plant model is

$$x(k+1) = A(x) + Bu(k)$$

Where $x(k) \in \boldsymbol{R}^n$,$u(k) \in \boldsymbol{R}^m$, A, B are known real matrices with appropriate dimension. It is assumed that there are random bounded delays in both SC and CA links, where $0 \leq \tau_{sc}(k), \tau_{ca}(k) \leq d$.

One of the main difficulties in NSCs control is the design of stabilizing state feedback control according to the available delay information in the controller node at sample k. The most known control approach is the two mode-dependent control which designs a switching control according to the assumption of availability of both of the most recent delays $\tau_{sc}(k)$and $\tau_{ca}(k-1)$. This assumption may require additional hardware to feed the controller with $\tau_{ca}(k-1)$ or a delayed version of it i.e $\tau_{ca}(k-\tau_{sc}(k)-1)$. In this approach the switching control values are designed offline using independent Homogenous Markov model probability transition matrices or their Kronecker matrix product for the two delay types [2],[14],[16]. The calculated values are stored in controller memory to be selected according to the delays at sample k. In contrast; this paper uses the single mode-dependent control, in which switching control is designed depending only on the value of $\tau_{sc}(k)$. The approach uses the proposed mutual Markov model probability matrix $P_{ca/sc} = \{p_{ij}\}$ instead the probability matrix of $\tau_{ca}(k-1)$ to compensate for the lack of information about the CA link delay at the control calculation time of sample k, where:

$p_{ij} = \mathrm{pr}\ (\tau_{ca}(k)/\tau_{sc}(k))$, $\sum_{j=0}^{\tau} p_{ij}$ for all $i, j \leq d$.

The control law can be written as:

$u(k) = K(\tau_{sc}(k))x(k-\tau_{sc}(k))$

The augmented state variable is given by:

$\tilde{x}(k) = [x(k)^T\ x(k-1)^T \ldots x(k-d)^T]^T$, then the closed loop system is:

$$\tilde{x}(k+1) = \left(\tilde{A} + \tilde{B}K(\tau_{sc}(k))C(\tau_{sc}(k))\right)\tilde{x}(k),$$

$$\tilde{A} = \begin{bmatrix} A & 0 & . & . & . & 0 & 0 \\ I & 0 & . & . & . & 0 & 0 \\ 0 & I & . & . & . & 0 & 0 \\ . & . & . & & & . & . \\ . & . & & . & & . & . \\ . & . & & & . & . & . \\ 0 & 0 & . & . & . & I & 0 \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} B \\ 0 \\ 0 \\ . \\ . \\ . \\ . \end{bmatrix}$$

$C(\tau_{sc,ca}(k))$ has all elements being zero except for $\tau_{sc}(k)$th block being an identify matrix. This closed loop system equation corresponds to a discrete-time jump linear system (JLS) in the form:

$$x(k+1) = A(\tau_{sc}(k))x(t)$$

The mean square stability of the above system (Theorem 3.1 in [2]) is equivalent to existence of symmetric positive definite matrices $Q_0, Q_1, \ldots, Q_d$ satisfying the following condition:

$$\sum_{i=0}^{d} p_{ji} \hat{A}_i^T Q_i \hat{A}_i < \alpha Q_j \quad \text{for all } 0 \leq j \leq d$$

where $\hat{A}_i = A_i + B_i K_i C_i$ and $\beta = 1/\alpha$ is the decay rate.

A similar V-K iteration algorithm as in [2] is used here to solve the previous bilinear matrix inequalities (BMI) using Lmilab of robust control toolbox of Matlab, Penbmi[6] and Yalmip[8].The algorithm is as follows:

- Design a LQR controller K for the plant without considering delays in the loop. Let $Ki=K$, $i=0, 1,.., d$, use initial probability matrix P_0.
- Repeat
  a. V-step. Given the controllers K_i, solve the LMI feasibility problem for all $j=0,1,\ldots,d$ with $\alpha=1$ to find Q_i, $i=0,1,\ldots,d$.
  b. K-step, given Qi found in V-step, solve the eigenvalue problem to find K_i, which maximize the decay rate (min α) of the closed loop.
  c. Use the new value of α.

  until the desired mutual probability $P_{ca/sc}$ ($\lambda'_{ca/sc}$) with min α is reached or V-step is not feasible.

In this algorithm, using the mutual Markov probability requires a less number of iterations to reach to a feasible solution in comparison to previous agorithms. In addition, it doesn't depend on $\tau_{ca}(k-1)$ or a delayed version of it. The following subsection gives a demonstration numerical example.

Numerical example

Consider the cart and inverted pendulum problem which is a fourth order unstable system. The state variables are $[x\ \dot{x}\ \theta\ \dot{\theta}]^T$.with $\theta(0)$=0.1 rad and all other initial state variables are zero. The parameters are the same as used in [2]; m1=1kg, m2=0.5kg, L=1m, the sampling time is T_s=20ms. Assume the system is controlled using a PLC with parameters Tp=Tc=20ms. The induced random delays resulted from CPN model simulation measured in number of samples are $M(\tau_{sc}) = M(\tau_{ca}) = \{1,2\}$ with mutual transition probability matrix:

$$P_m = \begin{bmatrix} 0 & 0 & 0.95 & 0.051 \\ 0 & 0 & 0.99 & 0.014 \\ 0.71 & 0.29 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \end{bmatrix}$$

First we design an LQR using weighting matrix $Q_x=I_4$ For the state and R_u=1 for the control signal. We get K=[0.9147 2.3520 39.5869 10.9682]

After solving the BMI stability condition using the proposed algorithm for ten steps results min α=0.9823 and K_1=[2.2854 6.5253 60.6805 22.0100], K_2=[0.2486 0.0001 0.0027 7.5940].

The algorithm used for solving the previous condition shows good results as it can be considered a robust algorithm i.e. it converges shortly to a feasible solutions compared to previous algorithms, which need a long time and may not result to a feasible solution [2],[16]. The algorithm depends only on $\tau_{sc}(k)$ and need not to use assumptions for $\tau_{sc}(k)$.

Fig. 9 shows the response of the initial pendulum position for a random delays sample according to the required transition probabilities. As seen the blue line (continuous line without marking) is the ideal system response with LQR and without delays. The green one (dashed with circles) is the normal control system with LQR and with delays. The red one (dotted line with +) is the response with single mode-dependent control with delays. It is noticed in this probability sample that the proposed control is close to ideal response, also the regular LQR can stabilize the system but there is no guarantee to stabilize all the random sequences because it is not a feasible solution for the BMI stability condition.

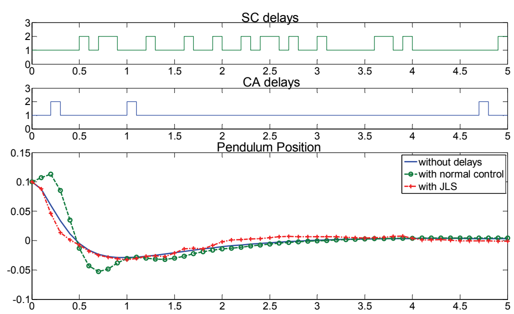

Fig. 9. Pseudo random induced network delays and system response

6. Conclusion

The paper presented a comprehensive procedure for modeling and control of PLC-based networked control systems. First a structure-conserving CPN model is built concentrating on the effect of client/server input/output scanning and program execution cycles for typical NCS cyclic controllers. By simulating the CPN model, sequences of delays are calculated.

Second, Markov analytical models are obtained for both sensor-to-controller τ_{sc} and controller-to-actuator τ_{ca} delays independently and in mutual/composite way to benefit from the correlation between the two delays.

The proposed mutual Markov delay models are used in the third part to design a switching single mode-dependent JLS control law for the system. A numerical example is presented to demonstrate the procedure. The results show that the proposed control design algorithm is efficient in solving the stability conditions in comparison to previous ones.

References

[1] J. Nilsson, "Real Time Control with Delays", PhD dissertation, Lund University, Sweden, 1998.

[2] L. Xiao, A. Hassibi, J.P. How, "Control with random communication delays via discrete time jump system approach" Proc. ACC, Vol. 3, pp. 2199-2204, Illinios, USA, June 2000.

[3] W. Zhang, M.S. Branicky, and S.M. Phillips, "Stability of networked control systems", *IEEE Control Systems Magazine*, Vol. 21(1), pp. 84-99, 2001.

[4] J. Arauz, P. Krishnamurthy, "Markov modelling of 802.11 channels", Proc. IEEE Vehicular Technology Conference, Vol. 2, pp.771-775, Oct. 2003.

[5] V. Casanova, J. Salt, "Multirate control implementation for an integrated communication and control system", *Control Engineering Practice*, Vol. 11, pp. 1335-1348, Nov. 2003.

[6] M Kocvara, M Stingl., "PENNON: A code for convex nonlinear and semidefinite programming", *Optimization Methods and Software*, Vol. 8, pp. 317-333, 2003

[7] D.A. Zaitsev, "An evaluation of network response time using a coloured Petri Net model of switched LAN", 5th Workshop and Tutorial on Practical Use of CPN and CPN Tools, pp. 157-167, Aarhus, Denmark, 2004.

[8] J. Löfberg "YALMIP: A toolbox for modeling and optimization in MATLAB". Proc. CACSD, Taipei, Taiwan, 2004.

[9] L. Zhang, Y. Shi, T. Chen, B. Huang, "A new method for stabilization of networked control systems with random delays", Proc. ACC, Vol. 1, pp. 633-637, Portland, USA, June 2005.

[10] G. Marsal, B. Denis, G. Frey, "Evaluation of response time in Ethernet-based automation systems", Proc. 11th IEEE ETFA, Prague, Czech Rep., pp. 380-387, 2006.

[11] K. Jensen, L. M. Kristensen, and L. Wells, "Coloured Petri nets and CPN tools for modelling and validation of concurrent systems" *International Journal on Software Tools for Technology Transfer,* pp. 213-254, 2007.

[12] A. Ghanaim, G. Frey, "Estimating delays in networked control systems using colored Petri nets and Markov chain models", Proc. 13th IEEE ETFA, Hamburg, Germany, pp. 1100-1103, 2008.

[13] J. Greifeneder, L. Liu, G. Frey, "Methods for analyzing response times in networked automation systems", 17th IFAC World Congress, Seoul, pp.5113-5118, 2008.

[14] B. Yu and Y. Shi, "State feedback stabilization of networked control systems with random time delays and packet dropout", DSCC'08, Ann Arbor, USA, Oct. 2008.

[15] A. Ghanaim, G. A. Borges, G. Frey, "Component based colored Petri net model for Ethernet based networked control systems", Proc. 14th IEEE ETFA, Mallorca, Spain, pp. 1-6, 2009.

[16] Y. Shi, B. Yu, "Output feedback stabilization of networked control systems with random delays modeled by Markov chains" IEEE TAC, V. 54, No. 7, July 2009.