Iranian Journal of Oil & Gas Science and Technology, Vol. 5, No. 1, pp. 53-64
http://ijogst.put.ac.ir

Design and Practical Implementation of a New Markov Model Predictive Controller for Variable Communication Packet Loss in Network Control Systems

Karim Salahshoor[1*], Babak Roshanipour[2], and Iman Karimi[3]

[1]Professor, Department of Instrumentation and Industrial Automation, Petroleum University of Technology, Ahwaz, Iran
[2]M.S. Student, Department of Instrumentation and Industrial Automation, Petroleum University of Technology, Ahwaz, Iran
[3]M.S. Student, Tehran Petroleum Research Center, Petroleum University of Technology, Tehran, Iran

Received: January 09, 2014; *revised:* July 23, 2014; *accepted:* August 23, 2014

Abstract

The current paper investigates the influence of packet losses in network control systems (NCS's) using the model predictive control (MPC) strategy. The study focuses on two main network packet losses due to sensor to controller and controller to actuator along the communication paths. A new Markov-based method is employed to recursively estimate the probability of time delay in controller to actuator path and a generalized predictive control (GPC) method is proposed to compensate the effect of big network time-delay, which leads to packet loss. The proposed methods and algorithms have been evaluated using a practical Smar fieldbus pilot plant to judge the efficiency of the foregoing algorithms. The obtained results clearly demonstrate the superiorities of the proposed control scheme with respect to standard MPC algorithm.

Keywords: Network Control System (NCS), Model Predictive Control (MPC), Markov Model, Packet Loss

1. Introduction

In resent decays, the complexity and size of industrial process plants is rising at a high speed due to the high demand of consumers for high quantities of quality products. As a result, the control actions of process plants are applicable just by using networked control systems (NCS's), which are networks with communications between nodes of individual or distributed controllers over a network. NCS's have attracted much interest because of their importance and effect on controlling processes. A lot of researches have been conducted in this subject (Zhang et al., 2013; Sun and Xu, 2012; Li et al., 2011; Liu et al., 2006). On the other hand, NCS structure has many benefits such as low cost, flexible structure, and simple troubleshooting. In addition, the successful application of NCS is not just in process industry and it can be found in a wide range of applications such as smart power transmission network (smart grid) and intelligent vehicle transportation system. Nevertheless, NCS has some disadvantages such as packet loss, which is the most important one and inevitable in transmission

---

* Corresponding Author:
Email: salahshoor@put.ac.ir

lines and in the process of transmitting a signal. The packet loss can cause instability in the controller and deteriorate the performance of the controller. Because of this prominent issue (instability), significant research has been carried out on it (Zhang et al., 2001). The majority of NCS works have addressed the controller design to provide acceptable stability for NCS's with packet loss. Typically, NCS modeling with packet loss has been performed as asynchronous dynamic systems or Markovian jumping systems (Zhang et al., 2001; Seiler and Sengupta, 2005). Multiple studies on packet loss have leaded to effective compensation strategies to compensate the controller deficiency during packet loss occurrence. Generally, control strategy in this structure is model predictive control (MPC), in which the controller computes a set of future control signals which remove the effects of data dropout (Liu et al., 2007; Zhangand 2007; Wang et al., 2010; Zhao et al., 2009; Zhang et al., 2013). Jiang and Fang (2013) proposed a segmented time-stamped dynamic matrix control (DMC) algorithm, which is a structure in MPC to reduce the online calculation burden. Wang and Yang (2007) proposed an MPC control strategy through buffering future control sequence to overcome the transmission delay problems between the controller and the actuator. In addition, observer-based estimation is also applicable as another compensation methodology (Lin et al., 2009). Furthermore, if the ideal control signal is missing, the latest control is used for compensation (Jiang and Fang, 2013). In network communication issues, in addition to packet loss, packet disordering also exists and should carefully be considered. However, in the mentioned literature, packet disordering is not addressed. Packet disordering means that a packet sent earlier may arrive at the destination node later or vice versa. Recently, packet disordering has been paid special attentions (Zhao et al., 2009; Wang and Yang, 2007). As it was mentioned before, packet loss (time delay) may lead to instability in controller and the controller structure proposed herein takes this issue into consideration. Some papers have investigated the effect of time delay with different approaches toward instability; for example, Zhang et al. (2013) investigated the influence of random time delays with a feedback method and Lin et al. (2009) proposed a model for time delays in NCS with Markov chains.

In this paper, a Markov model was used to estimate time delays and compensate packet loss in an MPC control structure. In addition, a practical pilot plant is utilized to evaluate the control algorithm on it and verify the proposed structure. The control signal for the pilot plant is generated by an MPC controller based on the minimization of cost function and according to the Markov model prediction for the system delay. In fact, we use the Markov model to estimate the controller to the actuator time delay.

2. System structure

The structure of the pilot plant and its control components are illustrated as block diagrams in Figure 1. The control loop is closed over fieldbus, which is linked through an industrial Ethernet network to a PC station. The experimental pilot plant is shown in Figure 2. It should be noted that sensor to controller and controller to actuator packet losses are implemented in the transmission line (Fadaei and Salahshoor, 2008).

3. Implementation of the proposed network control system

3.1. Network packet loss

In the proposed NCS, network delays and packet losses are considered. τ_{SC} and τ_{CA} are the main delays, which denote delays from sensor to controller and controller to actuator respectively. δ_{SC} and δ_{CA} are the packet losses in the same notation with delays.

It should be noted that there is no information about the controller to actuator delay and, in the best case, the previous sampled measurement is the only information on the delay. Thus the control signal cannot be calculated properly. However, in this paper a Markov model is suggested to address this issue, i.e. controller to actuator delay. In fact, the Markov model estimates the probability of delay in the next sample time based on the previous values.

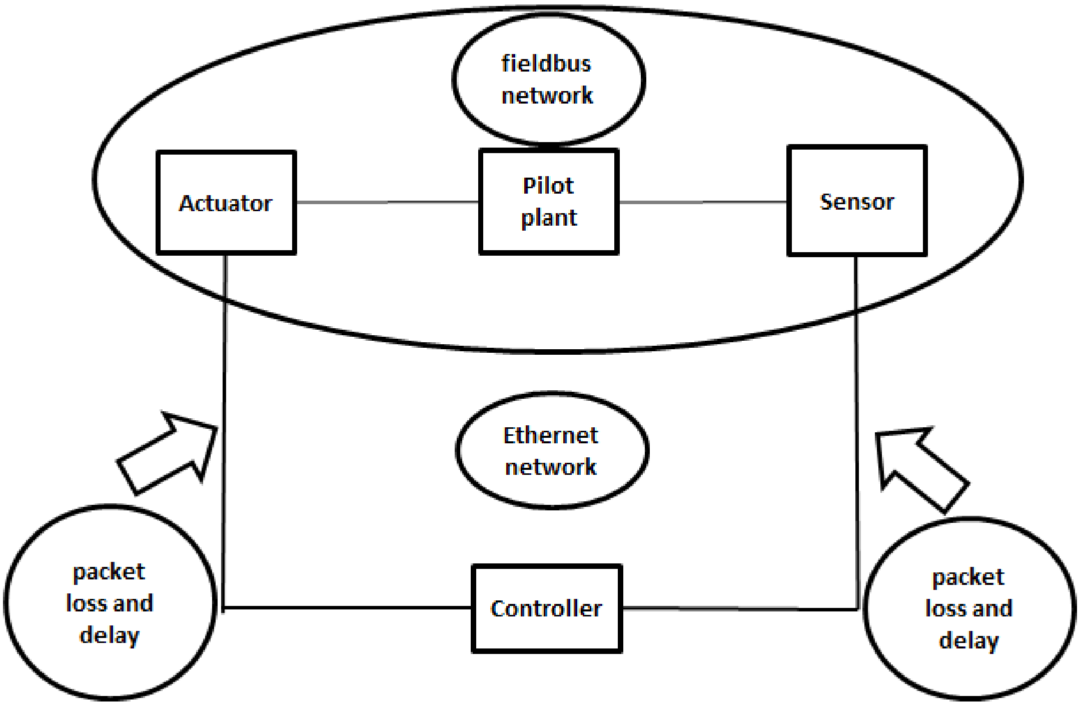

Figure 1
A schematic diagram of the experimental NCS.

Figure 2
Experimental pilot plant.

3.2. Markov model

For a better estimation of delay with the Markov model, the delay (τ_{CA}) is divided into three parts, namely *low*, *medium*, and *high*, which represents the states of the Markov model (Nilson and Bernhardsson, 1997). The three states are indicated by the following indices:

$$\mathrm{i} = \begin{cases} 0 & low \\ 1 & medium \\ 2 & high \end{cases} \tag{1}$$

It should be noted that the probabilities of the three parts are assumed Gaussian and they are shown in Figure 3. The probability of the present time step with respect to the previous step is tabulated in Table 1.

Table 1
Probability of τ_{ca} (now) with respect to τ_{ca} (previous).

		τ_{ca} (now)		
	i \ j	Low	Medium	High
τ_{ca} (previous)	Low	q_{00}	q_{01}	q_{02}
	Medium	q_{10}	q_{11}	q_{12}
	High	q_{20}	q_{21}	q_{22}

The notations used to indicate the probabilities obey the following conventions:

$$q_{ij} = p\,(j\,|i) \tag{2}$$

$$p\,(j\,|i) = \frac{number\ of\ events\ in\ this\ state}{number\ of\ events\ in\ state\ i} \tag{3}$$

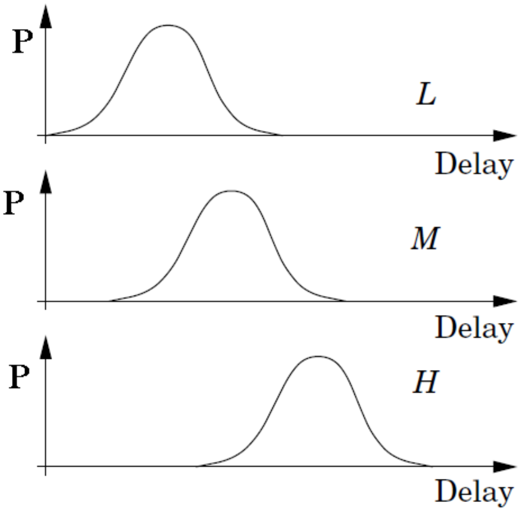

Figure 3
Gaussian distribution of τ_{ca}.

One of the main novelties of this method is the updating process of probabilities in each time step. In reality, delay probability can be varied due to any practical circumstances; thus, the novelty of this paper, namely updating probabilities in each time step, is practically useful. The procedure of the estimation is as follows:

1. Check the previous step to find out the real state;
2. Find the previous step state (*i*=Low, Medium, High), choose the state (*j*) which leads to bigger q_{ij} (with respect to Table 1);
3. For the chosen q_{ij}, calculate τ_{sc} with the Gaussian distribution shown in Figure 3;
4. Update q_{ij}.

3.3. Identification

In this structure, some of the output values will be lost. As a result, an identification procedure is needed to predict the values of the plant output. For the identification, a model-based estimator is employed based on an autoregressive moving average with an exogenous input (ARMAX) model of the following type (Ljung, 1999):

$$A(z^{-1})Y(t) = B(z^{-1})u(k-1) + C(z^{-1})\xi_k \tag{4}$$

$$A(z^{-1}) = 1 + a_1 z^{-1} + \cdots + a_{n_a} z^{-n_a} \tag{5}$$

$$B(z^{-1}) = 1 + b_1 z^{-1} + \cdots + b_{n_b} z^{-n_b} \tag{6}$$

$$C(z^{-1}) = 1 + c_1 z^{-1} + \cdots + c_{n_c} z^{-n_c} \tag{7}$$

where, $u(t)$ is the control variable and $y(t)$ denotes the actual measured plant output; A, B, and C represent the model polynomials in the backward shift operator z^{-1} and ξ_k shows the noise corrupting the controlled variable or the plant output.

The plant output signal can then be estimated by (Ljung, 1999):

$$\hat{y}_k = -a_1 y_{k-1} - a_2 y_{k-2} - \cdots - a_{n_a} y_{k-n_a} + b_1 u_{k-1} + b_2 u_{k-2} + \cdots + b_{n_b} u_{k-n_b} \tag{8}$$

3.4. GPC controller design

Generalized predictive control (GPC) is a method to design an MPC control strategy. It is based on the parametric plant model and, for the first time, it was introduced by Clarke et al. (1987). The cost function of this MPC algorithm is presented by Equation 9. The controller computes the future control signal via the minimization of the mentioned cost function:

$$J = \sum_{j=N_1}^{j=N_2} \rho\left(\hat{y}\left(t+j\middle|t\right) - w\left(t+j\right)\right)^2 + \sum_{j=N_1}^{j=N_u} \lambda \Delta u\left(t+j-1\right)^2 \tag{9}$$

where, N_1, N_2, and N_u are positive scalars defining the starting horizon, prediction horizon, and control horizon respectively. ρ denotes a non-negative control weighting scalar and λ is a non-negative signal control weighting scalar. $\hat{y}(t+j|t)$ denotes the *j* step forward prediction of *y*(*t*) based on the data available up to the time *t* and *w*(*t*+*j*) indicates the future set point value and, in general, is a reference trajectory.

If constraints are not taken into account, the solution to the minimization of the foregoing cost function will be given by (Clarke et al., 1987):

$$\Delta u = \left(G^T \rho G + \lambda I\right)^{-1} G^T \left(w - f\right) \tag{10}$$

where, *f* indicates the free response which depends on the past inputs and outputs of the system and *G* is the dynamic matrix defined as:

$$G = \begin{bmatrix} g_1 & 0 & . & . & . & 0 \\ g_2 & g_1 & 0 & . & . & 0 \\ . & . & . & 0 & . & 0 \\ . & . & . & . & 0 & 0 \\ . & . & . & . & . & 0 \\ g_p & g_{p-1} & . & . & . & g_{p-m+1} \end{bmatrix}$$

3.5. Control signal design for packet loss compensation

In the proposed NCS, packet loss will occur if at least one of the following two equations is satisfied $\delta_{SC} = 0$ or $\delta_{CA} = 0$. The values of δ_{sc} and δ_{ca} are calculated by:

$$\delta_{SC}=\begin{cases}0 & \tau_{SC}>hT_S\\ 1 & \tau_{SC}<hT_S\end{cases}$$
$$\delta_{CA}=\begin{cases}0 & \tau_{CA}>hT_S\\ 1 & \tau_{CA}<hT_S\end{cases} \tag{11}$$

In this paper, *h* is set equal to 4, which means that if the controller does not have the sensor data or the actuator does not have the action signal after 4 sample times ($4T_s$), packet loss will happen. It should be noted that no information is available on τ_{CA} in the current work and it is estimated by the Markov model in each sample time.

Four conditions, as given below, can be defined through the values of δ_{SC} and δ_{CA}.

1. If $\delta_{SC}=0$ and $\delta_{CA}=1$, the packet loss has occurred in the sensor to controller path and the controller should use the signal generated by GPC, which is calculated based on the previous sample time information.
2. If $\delta_{SC}=1$ and $\delta_{CA}=0$, the packet loss has occurred in the controller to actuator path and the controller should use the signal generated by GPC, which is calculated based on the previous sample time information. Here, $\delta_{CA}=0$ means that the estimated τ_{CA} is bigger than hT_S.
3. If $\tau_{SC}=t_i$ and $(\tau_{CA}+t_i)>hT_S$, i.e. $\tau=(\tau_{SC}+\tau_{CA})>hT_S$, packet loss has occurred and the controller should use again the signal generated by GPC, which is calculated based on the previous sample time information.
4. If $\delta_{SC}=1$ and $\delta_{CA}=1$, the packet loss will not happen and the transmission of the present signal to the destination point is granted.

3.6. Procedure of online implementation

For the online implementation of the proposed algorithm shown in Figure 3 and the proposed flowchart of the online control structure depicted in Figure 4, the following procedure is employed to recursively calculate the control signal of the pilot plant:

Step 1: The actual plant output ($y(k)$) is measured via a sensor and sent to the controller.

Step 2: Sensor to controller delay (τ_{sc}) is calculated by sample time and the time stamp of the data; if $\tau_{sc}>hT_s$ ($\delta_{sc}=0$), use the previous GPC control signals and start the algorithm again, else go to next step.

Step 3: System time delay (τ) is determined once the controller to actuator time delay (τ_{ca}) has been estimated by the Markov model in each sample time instant through adding it to the present sensor to controller delay (τ_{sc}); if $\tau>hT_s$, use the previous GPC control signals and start the algorithm again, else go to next step.

Step 4: The plant ARMAX model is identified or updated without the necessity of having access to the controller to actuator time delay, which might introduce inter-sampling effects.

Step 5: The identified model coefficients are sent to the GPC controller.

Step 6: GPC calculates the sequence of future control signals (i.e., $u(k)$, $u(k+1)$, $u(k+2)$, …, $u(k+m)$, …, $u(k+N_u)$).

Step 7: The calculated control signal is sent to the actuator.

Step 8: If a new measurement is received by the controller, the procedure follows the next step, otherwise it goes back to Step 1.

Step 9: Once a new measurement is arrived through the presumed event driven scheme, the output is updated.

Step 10: The procedure is repeated through Steps 2 to 7.

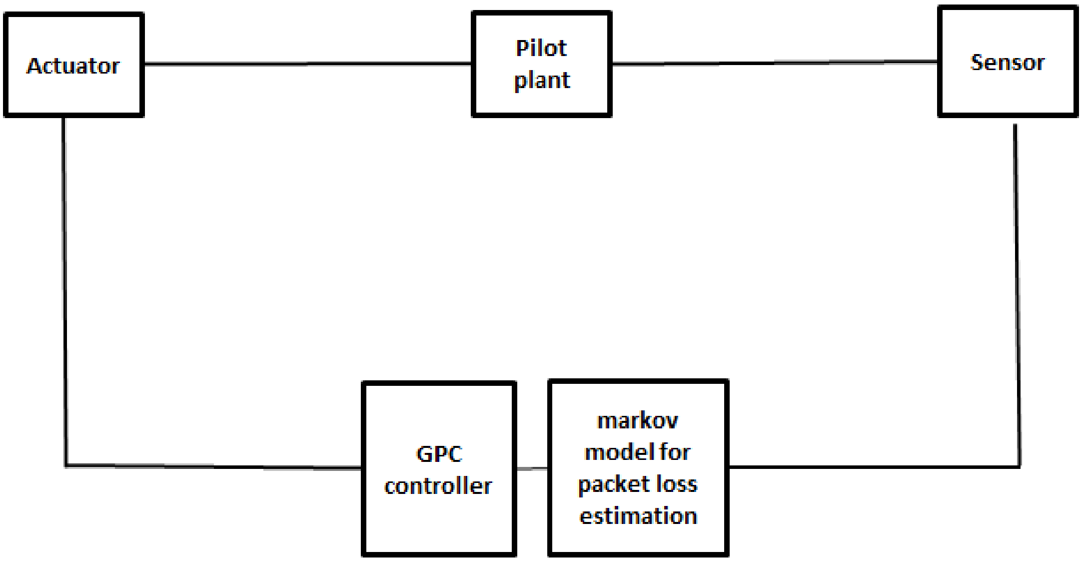

Figure 4
Elements of the proposed online control structure.

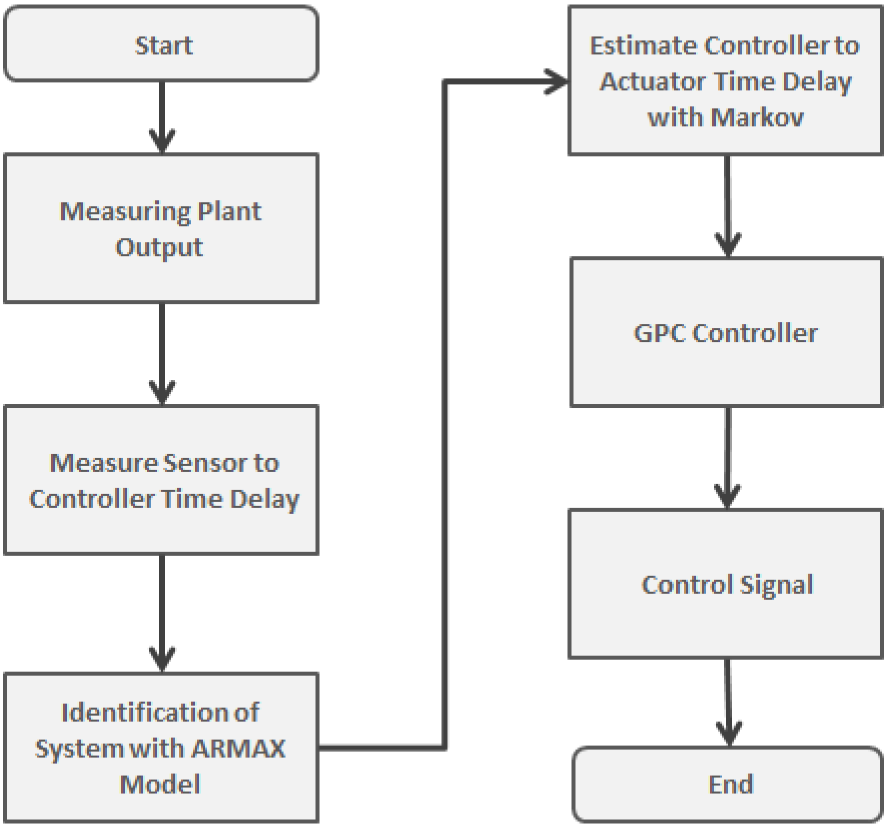

Figure 5
Flowchart of online control structure.

4. Experimental implementation

In this paper, a real fieldbus pilot plant was employed to conduct different test scenarios for exploring the effectiveness of the proposed control methodology in packet loss. Further details on the hardware and software considerations can be found elsewhere (Fadaei and Salahshoor, 2008). The sensor to controller and controller to actuator time delays are generated virtually on a random basis. The controller resides in the computer which is linked to a Smar DFI302 fieldbus through an industrial Ethernet network. The sensor is time-driven, whereas both the controller and actuator are event-driven. Here, event means the arrival of new data from sensors and a new control signal for controller and actuator respectively. At each sample-time, the value of τ_{sc} is known, the plant output is measured, and τ_{ca} will be estimated.

All the experimental tests are carried out under the following arranged assumptions:

1. The sensor is always time-driven.
2. The actuator is always driven by an event, corresponding to the arrival of a new control signal.
3. The controller is also driven by an event, corresponding to the arrival of new data from sensor. In case of packet loss, it uses its previous information.
4. The acquisition delay, the sensor delay, the computational delay, and the actuator delay are considered negligible and hence are accordingly neglected.
5. The system has data packet losses and their occurrence depends on system delays.

6. The communication delays corresponding to both τ_{sc} and τ_{ca} are considered to be random and their time-varying test patterns are manually maintained on an individual basis.
7. The sampling time is set to be constant and equals 0.2 s.
8. The initial conditions of all the tests are always set at the null output. As a result, all the nonlinearity in the transient time is considered.
9. The set point is constant and equal to 40% of the drum.

5. Results and discussion

A test scenario was organized to investigate the effect of packet loss through exploring the actual plant response. About 10% packet losses are considered during the experiments. It means that 10% of the whole data packets transmitted through the lines are lost. Figure 5 shows the foregoing condition of the packets of the data.

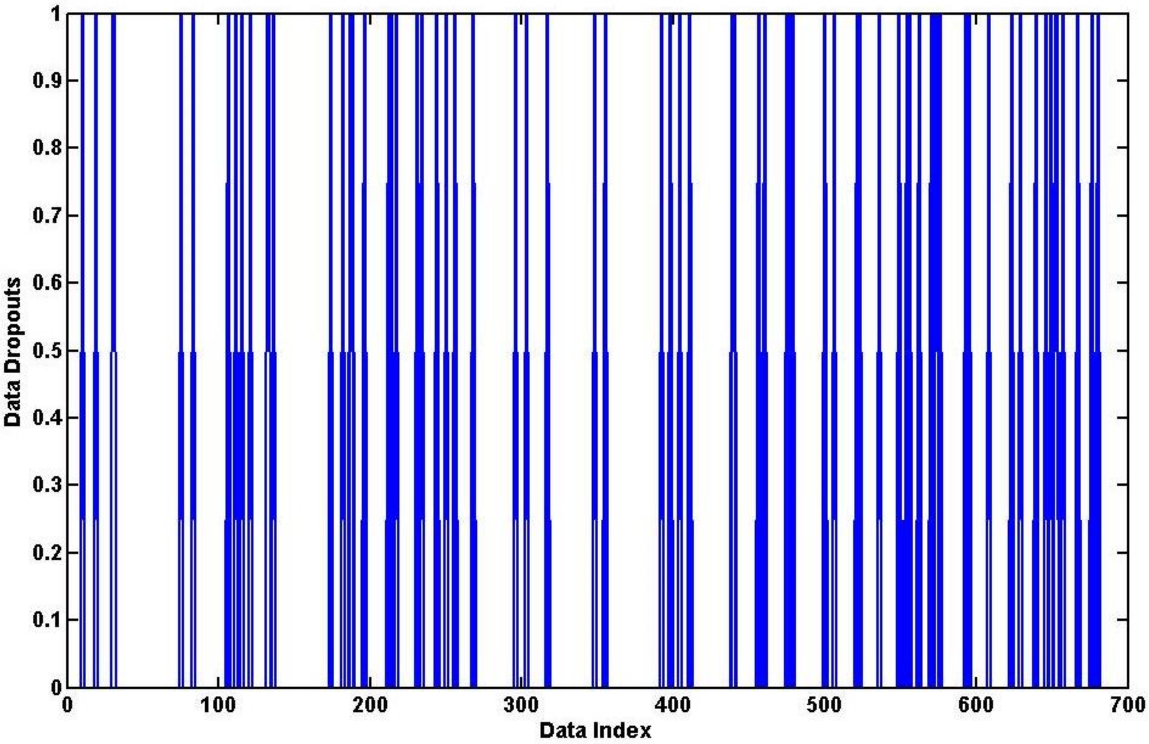

Figure 6
Number of data dropouts.

The plant is run two times with the same percent of packet loss. For the first time, the control strategy just includes MPC without any consideration of packet losses and delay estimation in its structure. The result of the obtained level response is shown in Figure 6 in red dotted line. For the second time, the Markov estimator is applied to the controller and the result is shown in blue line. As it can be seen, the results demonstrate the clear superiority of the proposed Markov-MPC controller over the conventional MPC controller in effectively compensating for the induced packet loss with a destructive effect. The system has two types of packet loss; one is in the sensor to controller path and the other is in the controller to actuator path. The first one is detectable, but the other one, controller to actuator, is not. This event degrades the MPC performance because the controller signal is calculated without detecting the controller to actuator packet loss. As a result, the controller action cannot follow the desirable set-point. But in the Markov MPC, packet loss in the controller to actuator path is estimated and considered in the control signal calculation; this consideration helps MPC to follow the set-point.

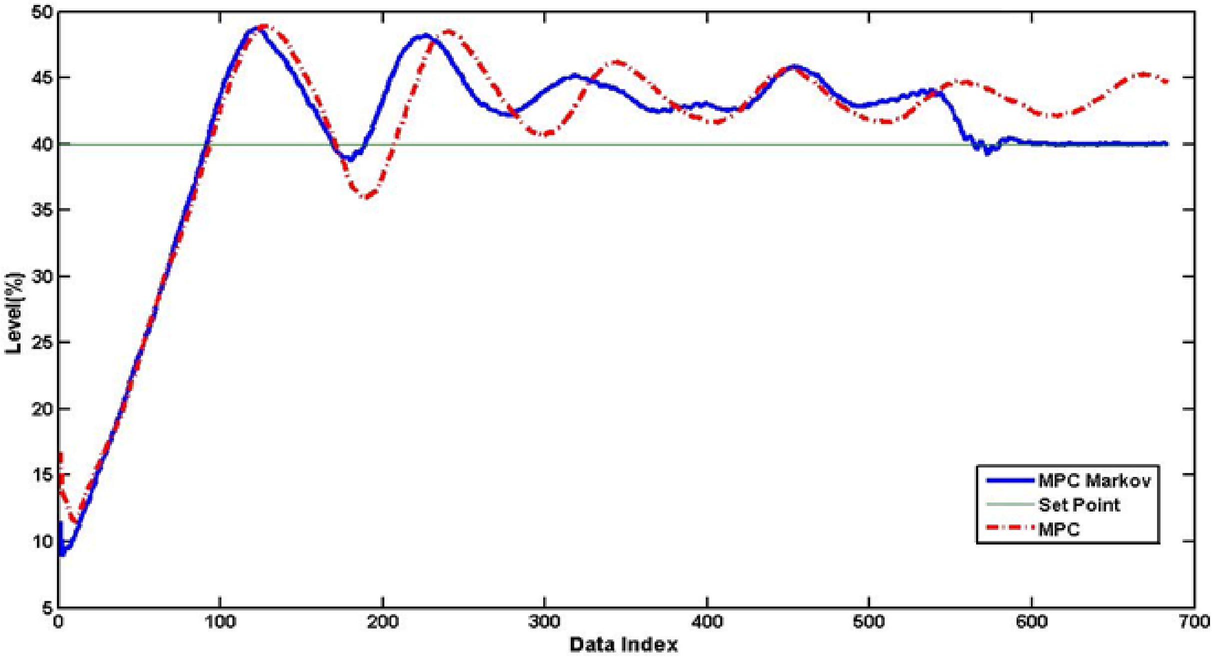

Figure 7
MPC and Markov MPC controller performances in the presence of 10% packet loss.

The selected pilot plant is extremely nonlinear. In addition, the ARMAX algorithm is employed in MPC to obtain the model of the plant. This algorithm is recursive and, in any steps of the estimation, might encounter different errors. Regarding these issues, the obtained result might be different. Consequently, there is no guaranty that similar results are exactly obtained in each run. Therefore, if another test is conducted, different results are obtained. On the other hand, the main goal of this work is related to the performance of the Markov method in time delay estimation and its integration with MPC algorithm; thus one test has only been conducted.

6. Conclusions

The current work investigates the destructive effect of packet loss in network control systems. The issue of packet loss has been studied under two separate parts, namely sensor-to-controller delay, which is intentionally varied, and controller-to-actuator delay, which is recursively estimated at each sample time instant through a new Markov model-based approach. The paper proposes an MPC methodology in combination with a Markov model to effectively compensate the disruptive effect of packet loss in NCS's. An experimental test scenario was designed and used in a practical fieldbus pilot plant to explore its closed-loop performance. The scenario examines the system performance under the influence of 10% sensor-to-actuator packet loss. The experiments comparatively examined the pilot plant performance under the control of both standard and Markov-based MPC controllers in terms of packet loss. The obtained experimental results clearly demonstrate the superiority of the proposed Markov-based MPC control scheme in maintaining both the stability and improving the performance.

Acknowledgments

This project is supported by Iranian Gas Engineering and Development Company (I.G.E.D.C.). The authors would like to thank Mr. Reza Rafie (electronic senior expert) for his kind help and support during this work.

Nomenclature

ARMAX	: Auto Regressive Moving Average with eXogenous input
GPC	: Generalized predictive control
MPC	: Model predictive control
NCS	: Network control system

References

Clarke, D. W., Mohtadi, C., and Tuffs, P. S., Generalized Predictive Control: Part I., the Basic Algorithm, Part II: Extensions and Interpretations, Automatica, Vol. 23, p. 137-160, 1987.

Fadaei, A. and Salahshoor, K., Design and Implementation of a New Fuzzy PID Controller for Networked Control Systems, ISA Transactions, Vol. 50, p. 351-61, 4, 2008.

Jiang, S. and Fang, H. J., Static Output Feedback Control for Nonlinear Networked Control Systems with Time Delays and Packet Dropouts, ISA Transactions, Vol. 52, No. 2, p. 215-222, 2013.

Li, H., Sun, Z., Chow, M., and F. Sun, Gain-scheduling-based State Feedback Integral Control for Networked Control Systems, IEEE Transactions on Industrial Electronics, Vol. 58, No. 6, p. 2465-2472, 2011.

Lin, C., Wang, Z., and Yang, F., Observer-based Networked Control for Continuous-time Systems with Random Sensor Delays, Automatica, Vol. 45, No. 2, p. 578-584, 2009.

Liu, G., Xia, Y., Chen, J., Rees, D., and Hu, W., Networked Predictive Control of Systems with Random Network Delays in both Forward and Feedback Channels, IEEE Transactions on Industrial Electronics, Vol. 54, No. 3, p. 1282-1297, 2007.

Liu, G. P., Mu, J. X., Rees, D., and Chai, S. C., Design and Stability Analysis of Networked Control Systems with Random Communication Time Delay Using the Modified MPC, International Journal of Control, Vol. 79, No. 4, p. 288-297, 2006.

Ljung, L., System Identification Theory for the User, Prentice Hall, Englewood Cliffs, NJ, 1999.

Nilsson, J. and Bernhardsson, B., LQG Control over a Markov Communication Network, Proceedings of the 36th IEEE Conference on Decision and Control, p. 4586-4591, 5 December, 1997.

Seiler, P. and Sengupta, R., An $H8$ Approach to Networked Control, IEEE Transactions on Automatic Control, Vol. 50, No. 3, p. 356-364, 2005.

Sun, Y. and Xu, J., Finite-time Boundedness and Stabilization of Networked Control Systems with Time Delay, Mathematical Problems in Engineering, Vol. 2012, p. 1-12, 2012.

Wang, R., Liu, G., Wang, W., Rees, D., and Zhao, Y. B, Guaranteed Cost Control for Networked Control Systems Based on an Improved Predictive Control Method, IEEE Transactions on Control Systems Technology, Vol. 18, No. 5, p. 1226-1232, 2010.

Wang, Y. L. and Yang, G. H., Control of Networked Control Systems with Time Delay and Packet Disordering, IET Control Theory &Applications, Vol. 1, No. 5, p. 1344-1354, 2007.

Zhang Y. and Li, S., Networked Model Predictive Control Based on Neighborhood Optimization for Serially Connected Large Scale Processes, Journal of Process Control, Vol. 17, No. 1, p. 37-50, 2007.

Zhang, J., Shi, P., and Xia, Y., Fuzzy Delay Compensation Control for T-S Fuzzy Systems over Network, IEEE Transaction on Cybernetics, Vol. 43, No. 1, p. 259-268, 2013.

Zhang, J., Xia, Y., and Shi, P., Design and Stability Analysis of Networked Predictive Control Systems, IEEE Transaction on Control Systems Technology, Vol. 21, No. 4, p. 1495-1501, 2013.

Zhang, W., Branicky, M. S., and Phillips, S. M., Stability of Networked Control Systems, IEEE Control Systems Magazine, Vol. 21, No. 1, p. 84-97, 2001.

Zhao, Y., Liu, G., and Rees, D., Design of a Packet-based Control Framework for Networked Control Systems, IEEE Transactions on Control Systems Technology, Vol. 17, No. 4, p. 859-865, 2009.