Predictive Compensation for Stochastic Time Delay in Network Control Systems[1)]

YU Shi-Ming YANG Ma-Ying YU Li

(*College of Information Engineering, Zhejiang University of Technology, Hangzhou* 310014)

(E-mail: ysm@zjut.edu.cn)

Abstract This paper proposes a predictive compensation strategy to reduce the detrimental effect of stochastic time delays induced by communication networks on control performance. Values of a manipulated variable at the present sampling instant and future time instants can be determined by performing a receding horizon optimal procedure only once. When the present value of the manipulated variable does not arrive at a smart actuator, its predictive one is imposed to the corresponding process. Switching of a manipulated variable between its true present value and the predictive one usually results in unsmooth operation of a control system. This paper shows: 1) for a steady process, as long as its input is sufficiently smooth, the smoothness of its output can be guaranteed; 2) a manipulated variable can be switched smoothly by filtering the manipulated variable just using a simple low-pass filter. Thus the control performance can be improved. Finally, the effectiveness of the proposed method is demonstrated by simulation study.

Key words Network control system, stochastic time delay, model predictive control, time delay compensation, low pass filtering

1 Introduction

The development of control techniques is closely associated with that of computer techniques all the time. Whether from CCS(centralized control system) to DCS (distributed control system) or from DCS to FCS (fieldbus control system), they were all governed and restricted by the developing status of computer techniques at that time. Nowadays, the rapid development of computer networks presents new challenges and problems for control theories and their applications, and computer-network-based control theories, algorithms and systems have attracted a great deal of interest from researchers.

In order to employ traditional control algorithms in network control systems (NCSs), Lee *et al.* proposed a parameter tuning method based on the genetic algorithm for a PID controller[1]. [2] investigated the optimal control strategy limited by communication resources, which was inspired by the fact that one remarkable property of biological systems is the ability to use their resources to perform more than one task simultaneously by appropriately focusing their attention. Considering the difficulties in modeling an NCS, Lee *et al.* presented a fuzzy logical control approach for an NCS[3]. [4] investigated fault detection for a network control system based on a memoryless reduced-order observer. Recently, researchers, including Zhang, Branicky, Yu *et al.*, have investigated the stability of NCSs for bounded network-induced time delays[5~7]. The time delay compensation based on the linear matrix inequality (LMI) approach, proposed by Yoo *et al.*, also requires a bounded time delay such that the NCS is asymptotically stable[8]. Although TOD (try-once-discard) is a dynamic scheduling method of NCSs, if the data packet of a controlled variable can not arrive at an actuator on time due to network time delay, the packet is thrown away and is replaced by the old one without any compensation[9]. Only for the bounded stochastic time delay, can this approach guarantee the stability of an NCS. However it is difficult to make the time delays keep within a prespecified upper bound for NCSs.

Since the 1990′s, model predictive control (MPC) has made rapid advances not only in applications but also in theory[10]. Multiple model predictive control in particular has drawn attention from many researchers[11]. Though MPC concerning NCSs can hardly be seen in the literature, this job will show that the MPC algorithm is a very effective tool to compensate for time delay in an NCS.

This paper presents a predictive compensation strategy for stochastic time delay induced by communication networks, using the property that values of a manipulated variable at the present and future

1) Supported by National High Technology Research and Development Program of P. R. China (2002AA412510) and National Natural Science Foundation of P. R. China (60274034)
Received February 10, 2003; in revised form December 15, 2003

time instants can be determined by performing a receding horizon optimal procedure only once. Unlike TOD, when the present value of a manipulated variable does not arrive at a smart actuator, its predictive one is imposed to the corresponding process and a simple digital low-pass filter is employed to switch the manipulated variable smoothly. Then, the effectiveness is demonstrated by simulation study.

2 MPC-based time delay compensation for NCS

As shown in Fig. 1, an NCS involves two classes of time delays: (a) the time delay d_{sc} from a sensor to the controller; (b) the time delay d_{ca} from the controller to an actuator. For details about the network-induced time delay, refer to [12]. The stochastic time delay in an NCS is induced by contention of multiple network nodes to gain a privilege access to a sharing communication bus. For the sake of simplicity, Fig. 1 only shows the single input and single output process. It is worth noting that the method can be easily extended to multiple input and multiple output processes.

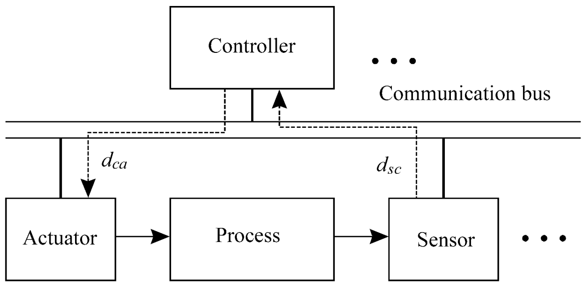

Fig. 1 Network control system diagram

The predictive output of a process consists of two components: 1) the unknown component expressed by the present value and the future values of the manipulated variable; 2) the known component determined by the present and past values of the controlled variable and the past values of the manipulated variable. It is denoted by

$$\boldsymbol{y}_p(k) = A(k)\triangle\boldsymbol{u}(k) + \boldsymbol{b} + \boldsymbol{w}(k) \tag{1}$$

where $\boldsymbol{y}_p(k) = (y(k+1), y(k+2), \cdots, y(k+P))^{\mathrm{T}}$; $\triangle\boldsymbol{u}(k) = (\triangle u(k), \triangle u(k+1), \cdots, \triangle u(k+M-1))^{\mathrm{T}}$; $A(k)$ is a $P \times M$ coefficient matrix determined by parameters of the process model; $\boldsymbol{b}$ is a P-dimension constant vector decided by the model parameters, the present and past output values and the past input values of the process; $\boldsymbol{w}(k)$ is a P-dimension noise vector caused by the network stochastic time delay, testing and quantization errors; P, M are the predictive and control horizons, respectively.

In this paper, considering the model of the process to be determinate and applying the predictive compensation and the filtering switching, we attenuate the adverse influence of the network time delay and the noises on the control performance.

The referential trajectory of MPC is a softened trace starting from the present output of the process and its discrete sequence is determined by

$$y_r(k+i) = \alpha_r^i y_k + (1-\alpha_r^i)sv \quad (i = 1, 2, \cdots, P) \tag{2}$$

where $y_r(k+i)$ is the future referential output at the $(k+i)-th$ time interval; y_k denotes the process output at the present time interval; α_r is a soft factor $(0 < \alpha_r < 1)$; sv represents a setpoint value of the process. If $y(k)$ arrives at the controller from the sensor within the given time, then

$$y_k = y(k) \tag{3}$$

Otherwise when $y(k)$ cannot arrive at the corresponding controller on time due to time delay, then

$$y_k = y(k/k-1) \tag{4}$$

where $y(k/k-1)$ is the predictive value of $y(k)$ solved at the past $(k-1)-th$ time interval. Since $y(k/k-1)$ is computed and stored beforehand by the local controller node, there exists no propagation delay. Once $y(k)$ cannot arrive at the controller on time, it is replaced by $y(k/k-1)$.

Let

$$\boldsymbol{y}_r(k) = (y_r(k+1), y_r(k+2), \cdots, y_r(k+P))^{\mathrm{T}}$$

Then, we gain the following optimal problem

$$\min \ \| \boldsymbol{y}_p(k) - \boldsymbol{y}_r(k) \| + \lambda \| \triangle \boldsymbol{u}(k) \|$$
$$\begin{cases} \triangle u_{\min} \leqslant \triangle u(k) \leqslant \triangle u_{\max} \\ u_{\min} \leqslant u(k) \leqslant u_{\max} \\ (k = 1, 2, \cdots, M) \end{cases} \tag{5}$$

where λ is a weighted factor.

Employing different norms in problem (5), we may obtain the corresponding optimal problem with constraints. Solving the above problem yields

$$\triangle \boldsymbol{u}(k) = (\triangle u(k), \triangle u(k+1), \cdots, \triangle u(k+M-1))^{\mathrm{T}}$$

So far, we have briefly summarized the predictive control algorithm with the constraints and the predictive compensation of $y(k)$. The predictive compensation strategy of the control increment vector $\triangle \boldsymbol{u}$ will be discussed in the following statements when it cannot reach the controller on time. In this situation, we not only consider the time delay of the present control increment $\triangle u(k)$, but also the time delay of the future control increments. So the compensation method coping with the time delay in this case is more complex than in the situation where $y(k)$ cannot reach the controller from a sensor on time.

The predictive compensating principle of $\triangle \boldsymbol{u}$ is shown in Fig. 2. In the same row, $\triangle u(1)$ denotes the present value of the control increment; $\triangle u(2)$ the predictive value at the next time interval; ......; and so forth. If $\triangle u(1)$ determined by the controller at the $(k+1)$th time interval, for instance, can not arrive at the actuator, it is replaced by the predictive value $\triangle u(2)$ computed at the kth time interval; similarly, if $\triangle u(1)$ computed at the $(k+2)$th time interval can not arrive at its actuator, the predictive value $\triangle u(3)$ gained at the kth time interval is available in substitution for $\triangle u(1)$; ......; this procedure can continuously proceed. It is worth noting that we should employ the newest predictive value to replace the present one if it does not arrive at the actuator on time. For instance, $\triangle u(1)$ determined at the $(k+2)$th time interval cannot reach the actuator, whereas the control increment vector $(\triangle u(1), \triangle u(2), \cdots, \triangle u(M))^{\mathrm{T}}$ determined at the $(k+1)$th time interval is available for the actuator, then the control increment $\triangle u(1)$ is replaced by the predictive value $\triangle u(2)$ determined at the $(k+1)$th time interval but not $\triangle u(3)$ gained at the kth time interval.

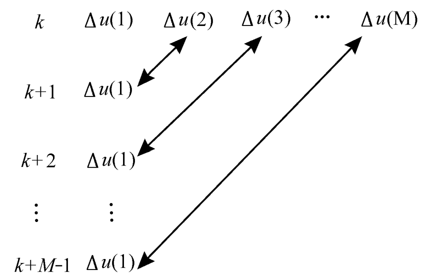

Fig. 2 Relation of present value and predictive values of control

The unsmooth operation of a system may be induced by switching of a manipulated variable between its present value and predictive one. Therefore, a simple low-pass filter with a unit gain is used to overcome the adverse effect of switching on the smooth performance of an NCS.

As illustrated in Fig. 3, the differences of $y(k)$ at the kth and $(k+1)$th time intervals are expressed by

$$\triangle y(k) = y(k) - y(k-1)$$

and

$$\triangle y(k+1) = y(k+1) - y(k)$$

respectively. T is the sampling period; α_k denotes the direction angle at $k-1$, α_{k+1} the direction angle at k. The smooth performance of the system can be described by the time sequence of its output. The smooth index of a time sequence is defined as follows.

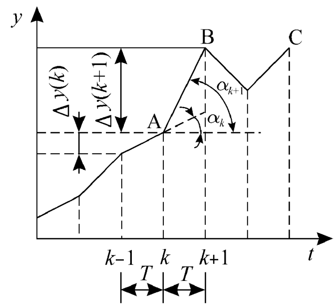

Fig. 3 Smooth analysis of time sequence

Definition 1. The smooth index $r(k)$ of the time sequence $\{y(k)\}$ at k is defined as

$$r(k) = |\tan\alpha_{k+1} - \tan\alpha_k| \tag{6}$$

Obviously, $-\frac{\pi}{2} < \alpha_k < \frac{\pi}{2}$, $0 \leqslant r(k) < \infty$. The smaller the value of $r(k)$, the smoother the time sequence $\{y(k)\}$. According to the definition, we can see from Fig. 3 that point A is smoother than point B. The following theorem presents an important result.

Theorem 1. The smooth index of the time sequence $\{y(k)\}$ at k equals the absolute value of the second difference of $y(k)$ at k divided by the sampling period T, *i.e.*,

$$r(k) = \frac{|\triangle^2 y(k+1)|}{T} \tag{7}$$

where $\triangle^2 y(k) = \triangle y(k) - \triangle y(k-1)$.

Proof. From Fig. 3, we can see that the theorem intuitively holds. The details about the proof are omitted. □

The following theorem shows that for a stable discrete time system, if the input sequence is sufficiently smooth, the smooth index of the output can be guaranteed.

Theorem 2. Assume that the input and output of a steady discrete time system satisfy the relation

$$y(k) = \sum_{m=-\infty}^{\infty} h(k-m)u(m) \tag{8}$$

where $\{h(m)\}$ denotes the unit sampling response sequence. For an arbitrarily given upper bound ε of the smooth index of the output sequence $\{y(k)\}$, there exists an upper bound δ of the smooth index of the input sequence $\{u(k)\}$, and if the smooth index of the input sequence $r_u(k) = \frac{|\triangle^2 u(k+1)|}{T} \leqslant \delta$ then the smooth index of the output sequence $r_y(k) = \frac{|\triangle^2 y(k+1)}{T} \leqslant \varepsilon$.

Proof. Since $\triangle y(k) = y(k) - y(k-1) = \sum_{m=-\infty}^{\infty} h(k-m)u(m) - \sum_{m=-\infty}^{\infty} h(k-1-m)u(m) = \sum_{m=-\infty}^{\infty} h(k-m)u(m) - \sum_{n=-\infty}^{\infty} h(k-n)u(n-1) = \sum_{m=-\infty}^{\infty} h(k-m)\triangle u(m)$, and

$$\triangle y(k+1) = y(k+1) - y(k) = \sum_{m=-\infty}^{\infty} h(k+1-m)u(m) - \sum_{m=-\infty}^{\infty} h(k-m)u(m) = \sum_{n=-\infty}^{\infty} h(k-n)u(n+1) - \sum_{m=-\infty}^{\infty} h(k-m)u(m) = \sum_{m=-\infty}^{\infty} h(k-m)\triangle u(m+1)$$

we have

$$\triangle^2 y(k+1) = \triangle y(k+1) - \triangle y(k) = \sum_{m=-\infty}^{\infty} h(k-m)\triangle^2 u(m+1)$$

$$r_y(k) = \frac{1}{T}|\triangle^2 y(k+1)| = \frac{1}{T}\left|\sum_{m=-\infty}^{\infty} h(k-m)\triangle^2 u(m+1)\right| \leqslant$$

$$\sum_{m=-\infty}^{\infty} |h(k-m)|\frac{|\triangle^2 u(m+1)|}{T} \leqslant \delta \sum_{m=-\infty}^{\infty} |h(k-m)| = \delta \sum_{m=-\infty}^{\infty} |h(m)|$$

Because the discrete time system is stable, $\sum\limits_{m=-\infty}^{\infty} |h(m)| = S < \infty$[13]. As a consequence, if we choose $\delta = \frac{\varepsilon}{S}$, then

$$r_y(k) \leqslant \delta S = \varepsilon \qquad \square$$

The following theorem shows that the adverse effect of switching on the smooth index can be reduced by using a simple low pass filter. This implies that we may design a controller which is suitable for NCSs without considering network-induced time delay.

Theorem 3. Suppose $0 < \alpha_f < 1$ a first-order low pass filer with a unit gain is given by

$$H(z^{-1}) = \frac{\alpha_f}{1-(1-\alpha_f)z^{-1}} \tag{9}$$

Let $\bar{u}(k)$ and $\triangle\bar{u}(k)$ be the filtered control and its increment, respectively, and $r_{\max}$ be the maximum smooth index without network delay. Filtering the control increment $\triangle u(k)$ (it is replaced by the predictive value $\hat{u}(k)$ when it can not arrive on time due to network-induced time delay) by employing the above filter yields

$$\lim_{k\to\infty} r_{\bar{u}}(k) \leqslant r_{\max}$$

Proof. Assuming that the steady gain of filter (9) is g_s, we have

$$g_s = \lim_{z\to 1} H(z^{-1}) = \lim_{z\to 1} \frac{\alpha_f}{1-(1-\alpha_f)z^{-1}} = 1$$

i.e., the steady gain equals 1.

For a constrained predictive control problem, $\triangle u(k)$ is bounded in terms of (5). As a result, $\triangle^2 u(k)$ is also bounded. Assuming $|\triangle^2 u(k)| \leqslant \triangle^2 u_{\max} < \infty$ will yield $r_{\max} = \dfrac{\triangle^2 u_{\max}}{T} < \infty$.

Using (9) yields $[1-(1-\alpha_f)z^{-1}]\triangle\bar{u}(k) = \alpha_f \triangle u(k)$.

Equivalently,

$$\triangle\bar{u}(k) = (1-\alpha_f)\triangle\bar{u}(k-1) + \alpha_f\triangle u(k) \tag{10}$$

Suppose that for a fixed time instant k_0, the control is switched from its real value $\triangle u(k_0)$ to the predictive one $\triangle\hat{u}(k_0)$, and $\triangle\hat{u}(k_0) = \triangle u(k_0) + e(k_0)$, where $e(k_0)$ is the switching error, and that $\triangle u(k_0)$ of (10) will be replaced by $\triangle\hat{u}(k_0)$ if $\triangle u(k_0)$ cannot arrive at the actuator. Then we gain

$$\triangle\bar{u}(k_0) = (1-\alpha_f)\triangle\bar{u}(k_0-1) + \alpha_f\triangle\hat{u}(k_0) = (1-\alpha_f)\triangle\bar{u}(k_0-1) + \alpha_f\triangle u(k_0) + \alpha_f e(k_0)$$

$$\triangle\bar{u}(k_0+r) = (1-\alpha_f)^{r+1}\triangle\bar{u}(k_0-1) + \sum_{i=0}^{r}\alpha_f(1-\alpha_f)^i\triangle u(k_0+r-i) + \alpha_f(1-\alpha_f)^r e(k_0) \tag{11}$$

$$\begin{aligned}\triangle\bar{u}(k_0+r+1) =&(1-\alpha_f)^{r+2}\triangle\bar{u}(k_0-1) + \sum_{i=0}^{r}\alpha_f(1-\alpha_f)^i\triangle u(k_0+r-i+1)\\ &+ \alpha_f(1-\alpha_f)^{r+1}\triangle u(k_0) + \alpha_f(1-\alpha_f)^{r+1}e(k_0)\end{aligned} \tag{12}$$

$$\begin{aligned}r_{\bar{u}}(k_0+r) =&\frac{|\triangle^2\bar{u}(k_0+r+1)|}{T} = \frac{|\triangle\bar{u}(k_0+r+1) - \triangle\bar{u}(k_0+r)|}{T} =\\ &\frac{1}{T}\Big| -\alpha_f(1-\alpha_f)^{r+1}\triangle\bar{u}(k_0-1) + \sum_{i=0}^{r}\alpha_f(1-\alpha_f)^i\triangle^2 u(k+r-i+1)+\\ &\alpha_f(1-\alpha_f)^{r+1}\triangle u(k_0) - \alpha_f^2(1-\alpha_f)^r e(k_0)\Big| \leqslant\end{aligned}$$

$$\frac{1}{T}\alpha_f(1-\alpha_f)^{r+1}|\triangle\bar{u}(k_0-1)| + \frac{\triangle^2 u_{\max}}{T}\sum_{i=0}^{r}\alpha_f(1-\alpha_f)^i +$$
$$\alpha_f(1-\alpha_f)^{r+1}\frac{|\triangle u(k_0)|}{T} + \alpha_f^2(1-\alpha_f)^r\frac{|e(k_0)|}{T}$$

$$\lim_{r\to\infty} r_{\bar{u}}(k_0+r) \leqslant \frac{\triangle^2 u_{\max}}{T} = r_{\max}$$

Let $k_0 + r = k$; then

$$\lim_{k\to\infty} r_{\bar{u}}(k) = \lim_{r\to\infty} r_{\bar{u}}(k_0+r) \leqslant r_{\max} \qquad \square$$

Remark 1. The above theorem shows that the smooth index $r_{\bar{u}}(k)$ of $\bar{u}(k)$ will be no more than $r_{\max}$ (the maximum smooth index without network time delay) with the proceeding of the recursive filtering algorithm. In other words, the filtering asymptotically attenuates the effect of switching on the smooth index. As long as we properly choose the lower and upper bounds $\triangle u_{\min}$, $\triangle u_{\max}$ in the problem (5), without considering network time delay, the $r_{\max}$ can be made small enough. As a consequence, the smooth index $r_{\bar{u}}$ is limited within the allowable range and the network control system performs smoothly.

3 Simulation study

Considering the NCS as illustrated in Fig. 1, we simulate the controller node with one microcomputer, and the process, sensor and actuator node altogether with another microcomputer within a LAN (local area network). The time delay from the process to the sensor and that from the actuator to the process are ignored. The data packets are transported between the nodes through Winsock programming interface , by using VC (Visual C)++ 6.0 as a programming tool and the connection-oriented protocol TCP (transmission control protocol) in the transport layer. The control algorithm can be implemented with the MATLAB and it can be run by being embedded in a VC++ program. Since the two nodes are located within a LAN, the collision and dropout of packets do not occur. To simulate the adverse effect of data collision of multiple nodes in the NCS on the control performance, we deliberately introduce the time delay with the Gaussian distribution both from the sensor to the controller and from the controller to the actuator. That is to say, the data packet to be transmitted waits for a random time length in terms of the value of a Gaussian-distribution random number before it is transmitted.

Suppose that the process model is described by

$$H(z^{-1}) = z^{-2}\frac{0.4638z^{-1} + 0.2346z^{-2} + 0.3921z^{-3}}{1 - 0.9403z^{-1} + 0.4305z^{-2} - 0.2346z^{-3}} \tag{13}$$

In all simulations, we assume that: (a) the sampling period $T = 2s$; (b) if the time delay d_{sc} from the sensor to the controller exceeds 0.95T, the present output packet is thrown away and is replaced by the predictive one using (4); (c) similarly, if the time delay d_{ca} exceeds 0.95T, the control increment packet is abandoned and substituted with the predictive one determined at the last time interval; (d) there exists the white noise with a zero mean and a variance of 0.05; (e) when t=202s, a disturbance with an amplitude of 40% of the setpoint is imposed on the output.

In the presence of network time delay, the controlled variable gained through TOD-based PID algorithm is illustrated in Fig. 4 for the controller gain $K_c = 0.12$ and $K_c = 0.24$. According to the statistics, the lost rates of the output packets $y(k)$ and the control increment $\triangle u(k)$ are 14.5% and 13.5%, respectively. From Fig. 4, we see that for the large controller gain, the PID algorithm yields the large oscillation and override, and hence, deteriorates the control performance; for the small controller gain, although the oscillation and the override are attenuated, the PID algorithm leads to the slow response of the system and the long regulation time due to both the network delay and the process delay. Therefore, whether for a large controller gain or a small controller gain, it is difficult to improve control performance for a complex industrial process in the presence of both network delay and process delay.

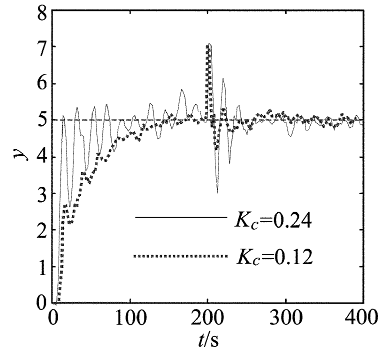

Fig. 4 Process output gained by TOD-based PID algorithm

For the time-delay process above mentioned, in the presence of network time delay, even if applying the predictive compensation, we cannot gain the satisfactory control performance due to the influence of switching on the smooth index. If we filter the control increments using the first-order low pass filter, choosing $\alpha_f = 0.05$, the control performance is significantly improved. Fig. 5 gives the comparison for the three algorithms which are the TOD-based PID algorithm with the controller gain of 0.12, the predictive compensation algorithms with and without a filter, respectively. As the same as in Fig. 4, for the PID algorithm, the lost rates of $y(k)$ and $\triangle u(k)$ are 14.5%, and 13.5%. The packet lost rates of the predictive compensation algorithm without a filter almost approach that of the PID algorithm and they are 15% for the output packets and 14% for the control increment packets. The predictive compensation algorithm with the low pass filter has the output and input packet lost rates of 14.5% and 14%, respectively. We can see form Fig. 5 that although the packet lost rates are very close to each other for the three cases, the control performance is quite different. The TOD-based PID algorithm leads to the slow response of the system and the some oscillation. Though the response speed is obviously improved by the predictive compensation algorithm without a filter, there still exits the oscillation phenomena. The predictive compensation algorithm with the low pass filer gives the fast response and attenuates the oscillation simultaneously.

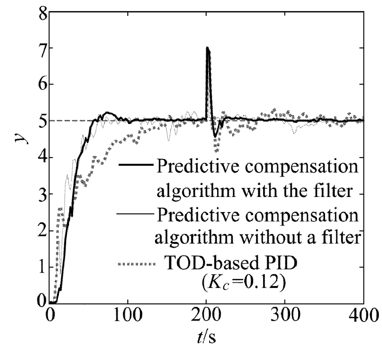

Fig. 5 Comparison of three algorithms

4 Conclusions

It is usually difficult to obtain satisfactory control performance for a complex process with time delay employing an NCS plus a typical PID algorithm due to the interaction of process delay and network delay and the complexity of the process itsclf, whereas the TOD strategy may further worsen the control performance. The predictive compensation algorithm can improve control performance, especially response speed, but it affects the smooth index due to control switching. The theoretical and simulation studies show that using a first-order low pass filter can further improve the control performance. It is quite a significant job to investigate the time-delay predictive compensation and the filtering switching method for NCSs.

References

1 Lee K C, Lee S. Remote controller design of networked control system using genetic algorithm. In: Proceedings of IEEE International Symposium on Industrial Electronics, New Jersey: IEEE Inc., 2001, **3**: 1845∼1850

2 Hristu D. Optimal Control with Limited Communication [Ph.D. dissertation]. Massachusetts: Department of Engineering Science, Harvard University, 1999

3 Lee K C, Lee S, Lee M H. Remote fuzzy logic control of networked control system via profibus-DP. *IEEE Transactions on Industrial Electronics*, 2003, **50**(4): 784∼792

4 Zheng Ying, Fang Hua-Jing, Wang Hua, Xie Lin-Bo. Fault detection approach for network control system based on a memoryless reduced-order observer. *Acta Automatica Sinica*, 2003, **29**(4): 559∼566

5 Zhang Wei. Stability Analysis of Networked Control Systems [Ph. D. dissertation], Ohio: Department of Electrical Engineering and Computer Science, Case Western Reserve University, 2001

6 Branicky M S, Phillips S R, Zhang Wei. Stability of networked control systems: explicit analysis of delay. In: Proceedings of the 2000 American Control Conference, New Jersey: IEEE Inc., 2000, **4**: 2352∼2357

7 Yu Zhi-Xun, Chen Hui-Tang, Wang Yue-Juan. Research on mean square exponential stable of time delayed network control system. *Control and Decision*, 2000, **15**(3): 278∼281, 289

8 Yoo H J, Ryu H S, Yoo K S, Kwon O K. Compensation of networked control systems using LMI-based delay-dependent optimization method. In: Proceedings of the 41st SICE Annual Conference, New Jersey: IEEE Inc., 2002, **1**: 364∼369

9 Walsh G C, Ye H. Scheduling of networked control systems. *IEEE Control Systems Magazine*, 2001, **21**(1): 57∼56

10 Xi Yu-Geng, Geng Xiao-Juan, Chen Hong. Recent advances in research on predictive control performance. *Control Theory and Applications*, 2000, **17**(4): 469∼475

11 Li Ning, Li Shao-Yuan, Xi Yu-Geng. Multiple model predictive control for MIMO systems. *Acta Automatica Sinica*, 2003, **29**(4): 516∼523

12 Lian Feng-Li. Analysis, Design, Modeling, and Control of Networked Control Systems [Ph. D. dissertation]. Michigan: Department of Mechanical Engineering, University of Michigan, 2001

13 Orfannidis S J. Introduction to signal processing. New Jersey: Prentice Hall, 1996, 115∼120

YU Shi-Ming Professor at College of Information Engineering, Zhejiang University of Technology and senior programmer granted by Ministry of Information Industry of China. Received his Ph. D. degree from Zhejiang University. His research interests include complex industrial process control and network control system.

YANG Ma-Ying Professor at College of Information Engineering, Zhejiang University of Technology. Received her Ph. D. degree from Zhejiang University. Her research interests include predictive control and controller performance monitoring.

YU Li Professor at College of Information Engineering, Zhejiang University of Technology. Received his bachelor degree from Nankai University in 1982, and his master degree and Ph. D. degree from Zhejiang University. His research interests include robust control, time-delay systems, and networked control systems.