Journal of Southeast University (English Edition) Vol. 21, No. 2, pp. 149 - 152 June 2005 ISSN 1003—7985

Time-stamped predictive functional control for networked control systems with random delays

Zhang Qizhi Zhang Weidong

(Department of Automation, Shanghai Jiaotong University, Shanghai 200030, China)

Abstract: The random delays in a networked control system (NCS) degrade control performance and can even destabilize the control system. To deal with this problem, the time-stamped predictive functional control (PFC) algorithm is proposed, which generalizes the standard PFC algorithm to networked control systems with random delays. The algorithm uses the time-stamp method to estimate the control delay, predicts the future outputs based on a discrete time delay state space model, and drives the control law that applies to an NCS from the idea of a PFC algorithm. A networked control system was constructed based on TrueTime simulator, with which the time-stamped PFC algorithm was compared with the standard PFC algorithm. The response curves show that the proposed algorithm has better control performance.

Key words: networked control systems; random delays; predictive functional control; industrial Ethernet

Networked control systems (NCS) are those control systems whose control loops are closed via a serial communication network[1]. There is increasing interest in using Ethernet for communication tasks in NCS because of its high speed, widespread usage and openness to web technology. Because the CSMA/CD protocol used by Ethernet is a stochastic bus arbitration scheme, when used as an industrial network, Ethernet will introduce random delays into the control loop, which degrade the control performance and can even destabilize the control system.

To deal with random delays in NCS, some researchers have proposed different control methods. Luck and Ray[2] used queues to reshape random delays in NCS to deterministic delays. Göktas[3] designed networked controllers in frequency domains based on robust control theory. Nilsson et al. [4] proposed a method to design the stochastic optimal controller for NCS with random delays shorter than one sample period. Srinivasagupta et al. [5] proposed a time-stamped dynamic matrix control algorithm, which inherits many advantages of predictive control, such as reliability, easy parameter tuning and constraint handling.

Predictive functional control (PFC) is another of the most popular model-based predictive control algorithms, and it has successfully been applied to many different control areas[6-8]. One outstanding merit of PFC is that a large sample period can be used, which is a desired feature for NCS with limited communication bandwidth. This paper generalizes the standard PFC algorithm into NCS with random delays shorter than one sample period.

1 Algorithm Derivation

1.1 Predicted outputs

The structure of time-stamped PFC is depicted in Fig. 1. Consider the state space model

$$\left.\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{b}u(t) \\ y(t) &= \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}(t) \end{aligned}\right\} \tag{1}$$

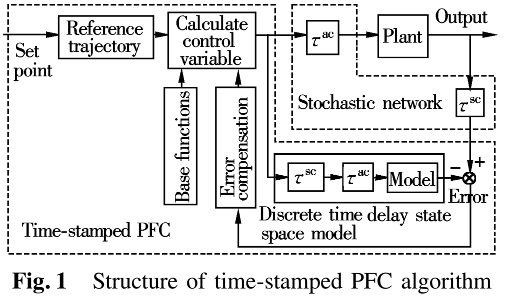

Fig. 1 Structure of time-stamped PFC algorithm

Denote τ_k^{sc}, τ_k^{c} and τ_k^{ac} as sensor-to-controller delay, calculation delay and controller-to-actuator delay, respectively. The effect of τ_k^{c} can be neglected when compared with τ_k^{c} and τ_k^{ac}. The sensor data is time-stamped to measure the sensor-to-controller delay. By doubling, the control delay can be estimated. When the control delay is shorter than one sample period ($\tau_k = \tau_k^{sc} + \tau_k^{ac} < h$), sampling (1) gives the discrete time delay state space model[9]:

Received 2004-10-08.

Foundation items: The National Natural Science Foundation of China (No. 60274032), the Science and Technology Rising-Star Program of Shanghai (No. 04QMH1405).

Biographies: Zhang Qizhi (1978—), male, graduate; Zhang Weidong (corresponding author), male, doctor, professor, wdzhang@sjtu.edu.cn.

$$\left.\begin{aligned}\boldsymbol{x}_{\mathrm{m}}(k) &= \boldsymbol{A}_{\mathrm{m}}\boldsymbol{x}_{\mathrm{m}}(k-1) + \boldsymbol{b}_{\mathrm{m}}^{0}u(k-1) + \boldsymbol{b}_{\mathrm{m}}^{1}u(k-2)\\ y_{\mathrm{m}}(k) &= \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{x}_{\mathrm{m}}(k)\end{aligned}\right\} \tag{2}$$

where $\boldsymbol{A}_{\mathrm{m}} = \mathrm{e}^{Ah}$, $\boldsymbol{b}_{\mathrm{m}}^{0} = \int_{0}^{h-\tau_k}\mathrm{e}^{As}\boldsymbol{b}\mathrm{d}s$, $\boldsymbol{b}_{\mathrm{m}}^{1} = \int_{h-\tau_k}^{h}\mathrm{e}^{As}\boldsymbol{b}\mathrm{d}s$, and $\boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}} = \boldsymbol{c}^{\mathrm{T}}$.

The system state at the $(k+i)$-th control step is

$$\boldsymbol{x}_{\mathrm{m}}(k+i) = \boldsymbol{A}_{\mathrm{m}}\boldsymbol{x}_{\mathrm{m}}(k+i-1) + \boldsymbol{b}_{\mathrm{m}}^{0}u(k+i-1) + \boldsymbol{b}_{\mathrm{m}}^{1}u(k+i-2) \tag{3}$$

Using the recursion method, the system output at the $(k+i)$-th control step can be predicted.

$$\begin{aligned}y_{\mathrm{m}}(k+i) = {}& \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{m}}^{i}\boldsymbol{x}_{\mathrm{m}}(k) + \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{m}}^{i-1}\boldsymbol{b}_{\mathrm{m}}^{0}u(k) + \\ & \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{m}}^{i-1}\boldsymbol{b}_{\mathrm{m}}^{1}u(k-1) + \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{m}}^{i-2}\boldsymbol{b}_{\mathrm{m}}^{0}u(k+1) + \\ & \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{m}}^{i-2}\boldsymbol{b}_{\mathrm{m}}^{1}u(k) + \ldots + \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{m}}\boldsymbol{b}_{\mathrm{m}}^{0}u(k+i-2) + \\ & \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{m}}\boldsymbol{b}_{\mathrm{m}}^{1}u(k+i-3) + \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{b}_{\mathrm{m}}^{0}u(k+i-1) + \\ & \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{b}_{\mathrm{m}}^{1}u(k+i-2)\end{aligned} \tag{4}$$

The future control variables are assumed to be composed of *a priori* known functions:

$$u(k+i) = \sum_{n=1}^{N}\mu_n(k)u_{\mathrm{B}n}(i) \tag{5}$$

where $1 \leqslant i \leqslant P$, P is the predictive horizon, μ_n are the coefficients to be calculated during the optimization process, $u_{\mathrm{B}n}$ are base functions, and N is the number of base functions. Substituting Eq. (5) into Eq. (4) gives

$$\begin{aligned}y_{\mathrm{m}}(k+i) = {}& \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{m}}^{i}\boldsymbol{x}_{\mathrm{m}}(k) + \\ & \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{m}}^{i-1}\boldsymbol{b}_{\mathrm{m}}^{1}u(k-1) + \boldsymbol{\mu}(k)^{\mathrm{T}}\boldsymbol{y}_{\mathrm{B}}(i)\end{aligned} \tag{6}$$

where $\boldsymbol{\mu}(t) = \{\mu_1(k), \mu_2(k), \ldots, \mu_N(k)\}^{\mathrm{T}}$, $\boldsymbol{y}_B(i) = \{y_{\mathrm{B}1}(i), y_{\mathrm{B}2}(i), \ldots, y_{\mathrm{B}N}(i)\}^{\mathrm{T}}$, and $\boldsymbol{y}_{\mathrm{B}n}(i)$ is defined as

$$\begin{aligned}y_{\mathrm{B}n}(i) = {}& \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{m}}^{i-1}\boldsymbol{b}_{\mathrm{m}}^{0}u_{\mathrm{B}n}(0) + \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{m}}^{i-2}\boldsymbol{b}_{\mathrm{m}}^{1}u_{\mathrm{B}n}(0) + \\ & \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{m}}^{i-2}\boldsymbol{b}_{\mathrm{m}}^{0}u_{\mathrm{B}n}(1) + \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{m}}^{i-3}\boldsymbol{b}_{\mathrm{m}}^{1}u_{\mathrm{B}n}(1) + \ldots + \\ & \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{m}}\boldsymbol{b}_{\mathrm{m}}^{0}u_{\mathrm{B}n}(i-2) + \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{b}_{\mathrm{m}}^{1}u_{\mathrm{B}n}(i-2) + \\ & \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{b}_{\mathrm{m}}^{0}u_{\mathrm{B}n}(i-1)\end{aligned} \tag{7}$$

Assume that the future predicted errors are approximated by a polynomial

$$e(k+i) = y_{\mathrm{p}}(k) - y_{\mathrm{m}}(k) + \sum_{j=1}^{N_{\mathrm{e}}}e_j(k)i^j \tag{8}$$

where y_{p} is the plant output, N_{e} is degree of the polynomial approximation, e_j are the coefficients calculated based on the past and present predicted errors. After error compensation, the predicted outputs become

$$y_{\mathrm{p}}(k+i) = y_{\mathrm{m}}(k+i) + e(k+i) \tag{9}$$

Substituting Eq. (6) and Eq. (8) into Eq. (9) gives

$$\begin{aligned}y_{\mathrm{p}}(k+i) = {}& \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{m}}^{i}\boldsymbol{x}_{\mathrm{m}}(k) + \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{m}}^{i-1}\boldsymbol{b}_{\mathrm{m}}^{1}u(k-1) + \\ & \boldsymbol{\mu}(k)^{\mathrm{T}}\boldsymbol{y}_{\mathrm{B}}(i) + y_{\mathrm{p}}(k) - y_{\mathrm{m}}(k) + \sum_{j=1}^{N_{\mathrm{e}}}e_j(k)i^j\end{aligned} \tag{10}$$

1.2 Control law

The reference trajectory is the path towards the future setpoint, which is given by

$$y_{\mathrm{r}}(k+i) = c(k+i) - \beta^i[c(k) - y_{\mathrm{p}}(k)] \tag{11}$$

where $\beta(0 < \beta < 1)$ is selected based on the desired closed-loop response time, and $c(k+i)$ is the setpoint calculated by the polynomial:

$$c(k+i) = c(k) + \sum_{j=1}^{N_{\mathrm{c}}}c_j(k)i^j \tag{12}$$

The performance index is a quadratic sum of the errors between the predicted outputs y_{p} and the reference trajectory y_{r}.

$$J(k) = \sum_{i=1}^{P}[y_{\mathrm{p}}(k+i) - y_{\mathrm{r}}(k+i)]^2 \tag{13}$$

Eq. (10) minus Eq. (11) gives

$$\begin{aligned}y_{\mathrm{p}}(k+i) - y_{\mathrm{r}}(k+i) = {}& \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{m}}^{i}\boldsymbol{x}_{\mathrm{m}}(k) + \\ & \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{m}}^{i-1}\boldsymbol{b}_{\mathrm{m}}^{1}u(k-1) + \boldsymbol{\mu}(k)^{\mathrm{T}}\boldsymbol{y}_{\mathrm{B}}(i) + \\ & y_{\mathrm{p}}(k) - y_{\mathrm{m}}(k) + \sum_{j=1}^{N_{\mathrm{e}}}e_j(k)i^j - c(k) - \\ & \sum_{j=1}^{N_{\mathrm{c}}}c_j(k)i^j + \beta^i[c(k) - y_{\mathrm{p}}(k)] = \boldsymbol{\mu}(k)^{\mathrm{T}}\boldsymbol{y}_{\mathrm{B}}(i) - \\ & \Bigl\{(1-\beta^i)[c(k) - y_{\mathrm{p}}(k)] + \sum_{j=1}^{\max(N_{\mathrm{c}},N_{\mathrm{e}})}\Bigl[c_j(k) - e_j(k)\Bigr]i^j - \\ & \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}(\boldsymbol{A}_{\mathrm{m}}^{i} - \boldsymbol{I})\boldsymbol{x}_{\mathrm{m}}(k) - \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{m}}^{i-1}\boldsymbol{b}_{\mathrm{m}}^{1}u(k-1)\Bigr\}\end{aligned} \tag{14}$$

If all the items in the brace of Eq. (14) are defined as

$$\begin{aligned}d(k+i) = {}& (1-\beta^i)[c(k) - y_{\mathrm{p}}(k)] + \\ & \sum_{j=1}^{\max(N_{\mathrm{c}},N_{\mathrm{e}})}[c_j(k) - e_j(k)]i^j - \\ & \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}(\boldsymbol{A}_{\mathrm{m}}^{i} - \boldsymbol{I})\boldsymbol{x}_{\mathrm{m}}(k) - \boldsymbol{c}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{m}}^{i-1}\boldsymbol{b}_{\mathrm{m}}^{1}u(k-1)\end{aligned} \tag{15}$$

then the performance index can be rewritten as

$$J(k) = \sum_{i=1}^{P}[\boldsymbol{\mu}(k)^{\mathrm{T}}\boldsymbol{y}_B(i) - d(k+i)]^2 \tag{16}$$

The optimization problem is to find a group of coefficients $\mu_1, \mu_2, \ldots, \mu_N$ to minimize $J(k)$.

By $\frac{\partial J(k)}{\partial \mu(k)} = 0$, we have

$$\boldsymbol{\mu}(k) = (\boldsymbol{Y}_{\mathrm{B}}\boldsymbol{Y}_{\mathrm{B}}^{\mathrm{T}})^{-1}\boldsymbol{Y}_{\mathrm{B}}\boldsymbol{d}(k) \tag{17}$$

where $\boldsymbol{Y}_{\mathrm{B}} = \{\boldsymbol{y}_{\mathrm{B}}(1), \boldsymbol{y}_{\mathrm{B}}(2), \ldots, \boldsymbol{y}_{\mathrm{B}}(P)\}$ and $\boldsymbol{d}(k) = \{d(k+1), d(k+2), \ldots, d(k+P)\}^{\mathrm{T}}$.

In fact, only the first control variable is acted on the plant, and the control law at present control step can be obtained from

$$u(k) = \boldsymbol{\mu}(k)\boldsymbol{u}_{\mathrm{B}}(0) = \sum_{n=1}^{N}\mu_n(k)u_{\mathrm{B}n}(0) \tag{18}$$

2 Simulation Platform

TrueTime, a Matlab/Simulink based simulator for real-time control systems[10], was used to validate the proposed control algorithm. The TrueTime simulator offers two Simulink blocks: the computer block and the network block, both of which can be customized to simulate a practical real-time control system.

The NCS simulation platform adopted in a later simulation is depicted in Fig. 2. The sensor, controller and actuator were achieved using three computer blocks. The Ethernet was brought about using a network block with the MAC protocol specified to CSMA/CD. The sensor and actuator were connected to the plant through their A/D converter and D/A converter, respectively. In addition, an inference node brought about by another computer block generated random interfering traffic over the network, which simulated network load generated by other communication nodes.

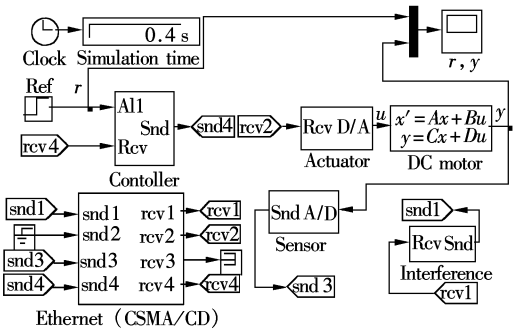

Fig. 2 NCS simulation platform

The time-driven sensor used a periodic task to sample the plant output at each control step. The sample data were time-stamped and then sent to the controller. The controller had an event-driven task that was triggered each time a sample arrived over the network. Upon receiving a sample, the controller computed a control output, which was sent to the event-driven actuator. After receiving the control signal, the actuator A/D coverted it immediately and acted on the plant.

3 Numerical Simulation

The simulation used a DC motor as the control plant, whose numerical state-space model was

$$\left.\begin{aligned} \dot{x} &= \begin{bmatrix} -4 & -0.03 \\ 0.75 & -10 \end{bmatrix} x + \begin{Bmatrix} 2 \\ 0 \end{Bmatrix} u \\ y &= \{0, 1\} x \end{aligned}\right\} \tag{19}$$

The control system was required to meet the following design specifications:

① Rising time shorter than 0.9 s;

② Overshoot of less than 20%;

③ Steady-state error of less than 5%.

The sample period was firstly set to 0.2 s, and the prediction horizon to 10. To trace the first order curve, the base function adopted steps. The response curve by the standard PFC controller is depicted in Fig. 3(a) when there was no delay. This meets the design specifications.

The network-induced delays were changed by setting the network bandwidth occupied by the interference node. In contrast, the response curves with maximum control delays of 0.50, 0.70 and 0.99 h are also depicted in Fig. 3(a). It can be seen that the system response deteriorates when the delay becomes larger.

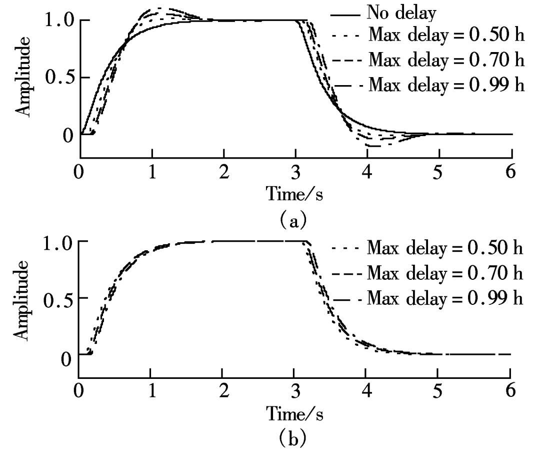

Fig. 3 When the sample period is 0.2 s. (a) Response curves with standard PFC; (b) Response curves with time-stamped PFC

The controller was redesigned according to the time-stamped PFC algorithm proposed in section 1. The design parameters were the same as those in the standard PFC algorithm. Fig. 3(b) depicts the response curves when the maximum network-induced delays were 0.50, 0.70 and 0.99 h. With reduced overshoot and shorter setting time, the control performance is apparently improved.

The above simulation was repeated using a sample period of 0.4 s. The corresponding response curves are depicted in Fig. 4. When a larger sample

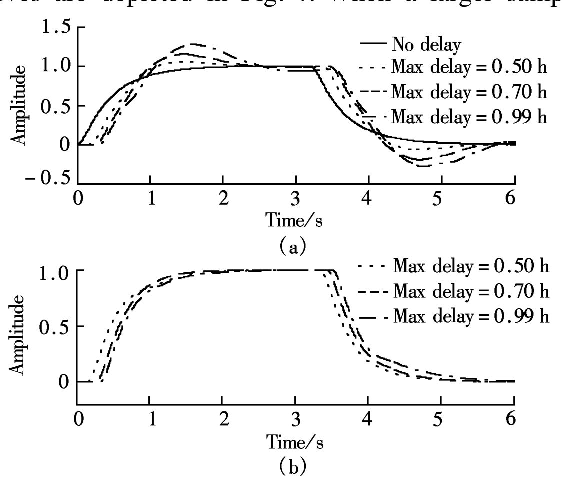

Fig. 4 When the sample period is 0.4 s. (a) Response curves with standard PFC; (b) Response curves with time-stamped PFC

period was adopted, the negative effect of network-induced delay was more significant for the control system using the standard PFC algorithm (see Fig. 4(a)). However, the time-stamped PFC algorithm can still give satisfactory control performance in such a situation (see Fig. 4(b)).

4 Conclusion

In this paper the time-stamped PFC algorithm was proposed for dealing with random delays in NCS. The standard PFC algorithm was modified to consider the network-induced random delays. An NCS simulation platform based on TrueTime simulator was constructed to validate the proposed algorithm. The simulation results show that the proposed time-stamped PFC algorithm can give better control performance compared with the standard PFC algorithm.

References

[1] Walsh G C, Ye H, Bushnell L. Stability analysis of networked control systems [A]. In: *Proceedings of the American Control Conference* [C]. San Diego, CA, 1999. 2876 −2880.

[2] Luck R, Ray A. Experimental verification of a delay compensation algorithm for integrated communication and control systems [J]. *International Journal of Control*, 1994, **59**(6): 1357 − 1372.

[3] Göktas F. Distributed control of systems over communication networks [D]. USA: Department of Computer and Information Science of University of Pennsylvania, 2000.

[4] Nilsson J, Bernhardsson B, Wittenmark B. Stochastic analysis and control of real-time systems with random time delays [J]. *Automatica*, 1998, **34**(1): 57 − 64.

[5] Srinivasagupta D, Schattler H, Joseph B. Time-stamped model predictive control: an algorithm for control of processes with random delays [J]. *Computers and Chemical Engineering*, 2004, **28**(8): 1337 − 1346.

[6] Vivas A, Poignet P, Pierrot F. Predictive functional control for a parallel robot [A]. In: *Proceedings of IEEE International Conference on Intelligent Robots and Systems* [C]. Las Vegas, Nevada, 2003. 2785 − 2790.

[7] Richalet J, Estival J L, Fiani P. Industrial applications of predictive functional control to metallurgical industries [A]. In: *Proceedings of 4th IEEE Conference on Control Applications* [C]. Albany, New York, 1995. 934 − 942.

[8] Pan H, Su H, Chu J. The study of predictive functional control for multivariable systems with time-delay and disturbance [J]. *Control Theory and Applications*, 2000, **17**(6): 813 − 819.

[9] Astrom K J, Wittenmark B. *Computer-controlled systems: theory and design* [M]. NJ, USA: Prentice-Hall, Inc, 1997. 267 − 268.

[10] Henriksson D, Cervin A. TrueTime 1.2 reference manual [EB/OL]. http://www.control.lth.se/~dan/truetime/. 2003-10-23/2004-09-15.

随机延迟网络中的时戳预测函数控制

张奇智　　张卫东

(上海交通大学自动化系, 上海 200030)

摘要: 针对网络控制系统中的随机延迟会恶化控制品质, 甚至使系统变得不稳定这一问题, 把标准预测函数控制算法推广到带随机延迟的网络控制系统中, 提出了时戳预测函数控制算法. 该算法通过时戳方法来估计网络在控制系统中引入的总延迟, 根据系统离散的延迟状态空间模型来预测未来输出, 并由预测函数控制的思想得到了适用于网络控制系统的控制规律. 在基于 TrueTime 工具箱搭建的网络控制系统仿真平台上, 对比了时戳预测函数算法和传统预测函数算法, 系统响应曲线显示前者具有更好的控制品质.

关键词: 网络控制系统; 随机延迟; 预测函数控制; 工业以太网

中图分类号: TP273