Journal of Zhejiang University SCIENCE A
ISSN 1009-3095
http://www.zju.edu.cn/jzus
E-mail: jzus@zju.edu.cn

Application of generalized predictive control in networked control system

YANG Can (杨 灿)†, ZHU Shan-an (朱善安), KONG Wan-zeng (孔万增), LU Li-ming (卢立明)

(*School of Electrical Engineering, Zhejiang University, Hangzhou 310027, China*)

†E-mail: yangcan_1220@163.com

Received Oct. 25, 2004; revision accepted Mar. 19, 2005

Abstract: A new framework for networked control system based on Generalized Predictive Control (GPC) is proposed in this paper. Clock-driven sensors, event-driven controller, and clock-driven actuators are required in this framework. A queuing strategy is proposed to overcome the network induced delay. Without redesigning, the proposed framework enables the existing GPC controller to be used in a network environment. It also does not require clock synchronization and is only slightly affected by bad network condition such as package loss. Various experiments are designed over the real network to test the proposed approach, which verify that the proposed approach can stabilize the Networked Control System (NCS) and is robust.

Key words: Networked Control System (NCS), Generalized Predictive Control (GPC), Queuing strategy
doi:10.1631/jzus.2006.A0225 **Document code:** A **CLC number:** TP273

INTRODUCTION

Networked Control System (NCS) is composed of a central controller and a remote system containing a physical plant, sensors and actuators (Halevi and Ray, 1988; Nilsson, 1998; Walsh *et al*., 1999a; 1999b; Branicky *et al*., 2000). The controller and the plant are located at different spatial locations and directly connected through network to form a closed loop control. Fig.1 shows the NCS configuration and the time delay induced by network, where u is the control signal and y the output signal. The network delay can be categorized into two parts according to the direction of data transfers. One is the Controller-to-Actuator delay τ^{ca}, and the other is the Sensor-to-Controller delay τ^{sc}.

Network delays are well known to degrade the performance of control systems or even to destabilize it (Zhang *et al*., 2001), so the overall NCS performance is greatly affected by the network delay regardless of the types of networks used.

Recently, various methods have been developed to maintain the stability and the performance of NCS with delay problems:

(1) A deterministic predictor-based delay compensation was developed (Luck and Ray, 1990; 1994). This method uses an observer to estimate the plant states and a predictor to compute the predictive control based on past output measurements. However, this method prolongs the delay artificially so that it is so conservative that the performance depends greatly on the model accuracy.

(2) An optimal stochastic control method is proposed to control an NCS on random delay networks (Nilsson, 1998). The effects of random network delays in an NCS were treated as a Linear-Quadratic-Gaussian (LQG) problem.

(3) A non-linear and perturbation theory (Walsh *et al*., 1999a; 1999b) was used to formulate periodic and random network delay effects in an NCS as the vanishing perturbation of a continuous-time system under the assumption that there is no observation noise. However, this methodology requires a very small sampling time so that an NCS can be approximated as a continuous-time system.

(4) By using robust control theory, a networked

controller is designed in frequency domain (Göktas, 2000). A major advantage of this methodology is that it does not require a priori information about the probability distributions of network delays.

(5) An intelligent control was proposed using fuzzy logic to adaptively compensate for IP network induced time delay in NCS applications (Almutairi *et al*., 2001). The advantage of the fuzzy logic compensator is that the existing PI controller need not to be redesigned, modified, or interrupted for use on a network environment.

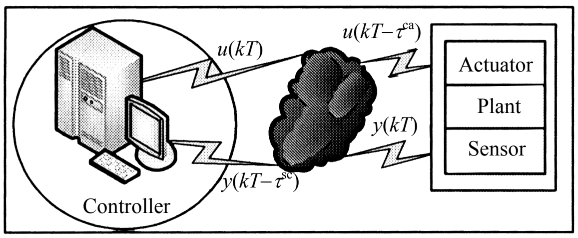

Fig.1 Configuration of NCS

In this paper, a new methodology based on Generalized Predictive Control (GPC) (Clarke *et al*., 1987) was proposed to cope with the network induced time delay. This methodology uses GPC to compute predictive control signals based on the past measurement. A queuing strategy for using predictive control signal is proposed. We use the campus-wide network of Zhejiang University for the experiments, which is a local area network (ZJULAN) containing Ethernet at the physical layer.

PLANT DESCRIPTION AND GPC ALGORITHM FOR REAL TIME CONTROL

Consider the plant described by CARIMA (Controlled Auto-Regressive Integrated Moving Average) model

$$\boldsymbol{A}(q^{-1})y(t)=\boldsymbol{B}(q^{-1})u(t-1)+\boldsymbol{C}(q^{-1})\xi(t)/(1-q^{-1})$$

and the cost function

$$J(N_1,N_2,N_u,\lambda)=E\left\{\sum_{j=N_1}^{N_2}[(y(t+j)-y_r(t+j)]^2+\sum_{j=1}^{N_u}\lambda(j)[\Delta u(t+j-1)]^2\right\}, \quad (1)$$

where $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are polynomials in the backward shift operator q^{-1}; N_1 is the minimum costing horizon; N_2 is the maximum costing horizon and $\lambda(j)$ is a control-weighting sequence.

Minimizing the cost function yields the control law

$$\Delta\boldsymbol{U}=(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}+\lambda\boldsymbol{I})^{-1}\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{Y}_r-\boldsymbol{Y}_1), \quad (2)$$

where $\boldsymbol{G}$ is a $(N_2-N_1+1)\times N_u$ matrix:

$$\boldsymbol{G}=\begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_2-1} & g_{N_2-1} & \cdots & g_{N_2-N_u} \end{bmatrix}.$$

Its elements are the coefficients of step response. $\boldsymbol{Y}_r=[y(t+N_1), y(t+N_1+1), \ldots, y(t+N_2)]^{\mathrm{T}}$ is the reference signal within the prediction horizon; $\boldsymbol{Y}_1=[y_1(t+N_1), y_1(t+N_1+1), \ldots, y_1(t+N_2)]^{\mathrm{T}}$ is the prediction based on the past measurement.

Notice that $\Delta\boldsymbol{U}$ is not a scalar but a vector which can be written as Eq.(3):

$$\Delta\boldsymbol{U}_t=[\Delta u(t)\ \Delta u(t+1)\ \ldots\ \Delta u(t+N_u)]. \quad (3)$$

In real time control,

$$u(t)=u(t-1)+\Delta U_t[0], \quad (4)$$

where $\Delta U_t[0]$ is its first element $\Delta\boldsymbol{U}_t$. Let us define: $\Delta U_t[j]$ $(j<N_u)$ to be the $(j-1)$th element of $\Delta\boldsymbol{U}$ at time t and the control signal series to be $U_t[j]$, $j=0,\ldots,N_u-1$

$$\begin{cases} U_t[0]=u(t-1)+\Delta U_t[0] \\ U_t[j]=U_t[j-1]+\Delta U_t[j] \quad (j\geq 1) \end{cases}, \quad (5)$$

where $u(t-1)$ is the plant input at time $t-1$.

GPC FOR NCS

In this section, we investigate how to apply GPC in NCS. Note that only the first value of $\Delta\boldsymbol{U}$ is actually used in real time control. There are N_u-1 elements in $\Delta\boldsymbol{U}$ which are left unused. Using them to cope with the affects of network delay is our main consideration.

Response modes of sensor, controller and actuator

To apply GPC in NCS, requires that:

(1) clock-driven sensors that sample the plant outputs periodically at sampling instants;

(2) an event-driven controller which can be implemented by an external event interrupt mechanism and calculates the control signal as soon as the sensor data arrives;

(3) clock-driven actuators, which means the plant inputs are not changed until the sampling instants (Fig.2).

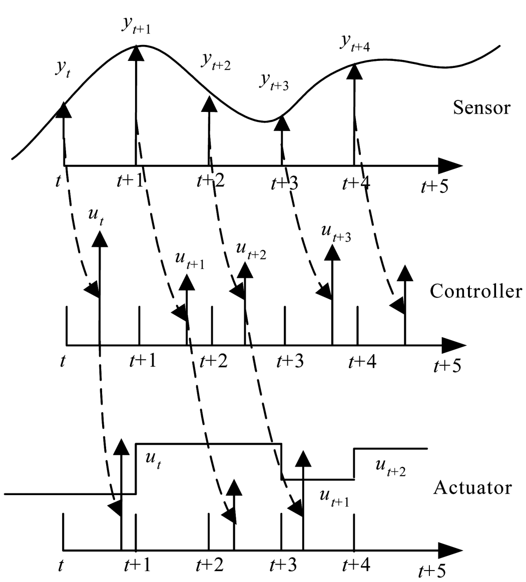

Fig.2 Response modes: Senor & Actuator: time driven; Controller: event driven

Queuing strategy (QS)

How to choose the input of the plant for the actuator in NCS? To demonstrate our basic idea, we assume no exceptional errors (such as package loss, the control signal arriving in disorder, etc.) occur. The situation can be categorized into two cases: (1) no new signal arrives within the sample period; (2) new signal arrives.

Case 1: No new signal arrives

When the delay occurs, it is natural for the actuator to deal with the delay by using the predictive control signals calculated from previous state instead of the current control signal which has not arrived at the actuator yet. Fig.3 shows the method.

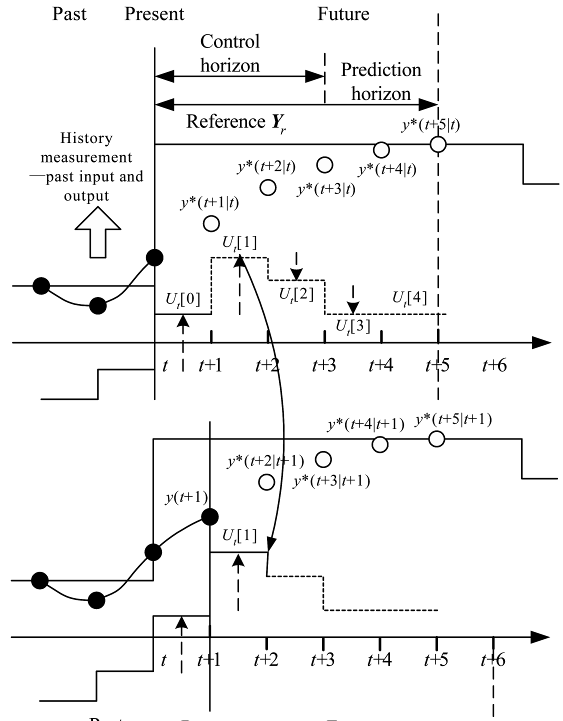

Fig.3 Queuing strategy for Case 1

Case 2: New signal arrives

Obviously, the newest control signal has the highest priority to be the input of the plant. The issue is which element of the newest control signal should be chosen. Take time t for example. Suppose that the newest control signal $\boldsymbol{U}_{t-2}$ is based on the state at time t–2. To match the state, $U_{t-2}[2]$ should be taken as the input of the plant (Fig.4).

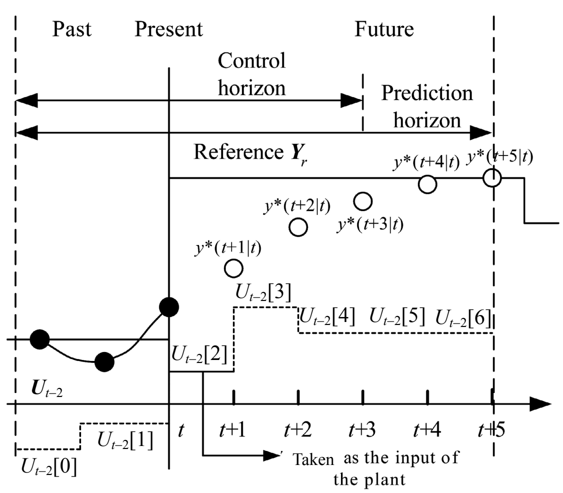

Fig.4 Queuing strategy for Case 2

Combination of Case 1 and Case 2 and the resulting general principle is shown in Fig.5. The general expression can be easily concluded as:

$$\boldsymbol{U}_t = U_n[t-n] \text{ (for } t\text{th sample time)}, \quad (6)$$

where n is the time-tag of the control signal and satisfies $n \le t$.

Sample time	Control signal Should be applied			
1	$U_1[0]$			
2	$U_2[0]$	$U_1[1]$		
3	$U_3[0]$	$U_2[1]$	$U_1[2]$	
4	$U_4[0]$	$U_3[1]$	$U_2[2]$	$U_1[3]$
…	…			
	High	Priority		Low

Fig.5 Queuing strategy for general case

The time-tag is especially useful when the earlier control signal arrives at the actuator later. The actuator can pick up the newest control signal among all the signals which arrive during a sample period. Fig.6 is the flowchart of the queuing strategy for the actuator.

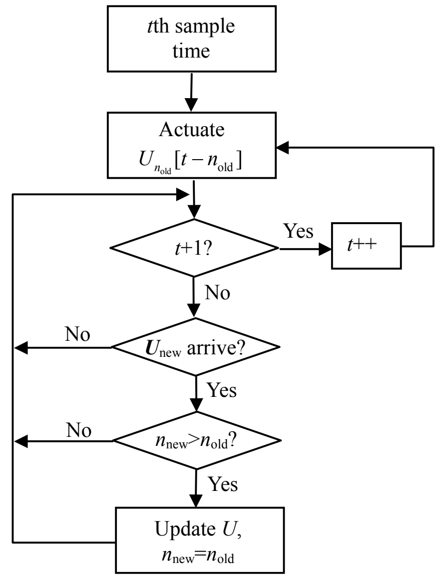

Fig.6 Flow chart of queuing strategy for actuator

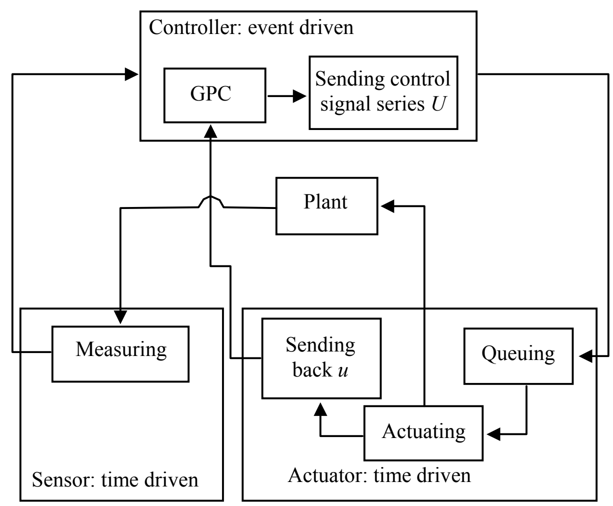

Fig.7 Framework of applying GPC in NCS

Summarization of applying GPC in NCS

The strategy for applying GPC in NCS is summarized as follows:

(1) The controller also has to cope with the situation that earlier measurement arrives later. The treatment can be the same as that for the actuator.

(2) Using model to predict future output is indispensable for both real time control and NCS. However, the controller knows the input of the plant in real time control, while the central controller in NCS may not know that according to the queuing strategies. So the actuator is required to send back the input of the plant.

(3) The proposed approach does not involve Model-based compensation (Zhang *et al*., 2001), so that clock synchronization (Mills, 1991) is not necessary.

(4) Package loss does not greatly affect the NCS performance.

The framework of applying GPC in NCS is shown in Fig.7.

MODELING UNCERTAINTY OF *P*-STEP PREDICTION

The performance of predictive control depends on the model used for the prediction. So it is worthwhile to investigate the effects of modeling uncertainty on the performance of the proposed algorithm.

The first step is to convert the time-varying delay into a constant delay. Supposing that the maximum delay is τ_{max}, p-step prediction must be used to deal with the maximum delay according to the algorithm. The worst case is that the control signal $u(t)$ is always calculated from the state $x(t-p)$. The most conservative consideration for modeling the uncertainty's effect is to take the worst case instead of the ordinary case, which may involve time-varying delay.

According to Clarke *et al*.(1994), the true dynamic system can be described as:

$$\boldsymbol{\alpha}_{t+1} = \boldsymbol{A}'\boldsymbol{\alpha}_t + \boldsymbol{B}'\Delta u_t , \tag{7}$$

where

$$\boldsymbol{\alpha}_t = [y_t \quad \dots \quad y_{t-n+1} \quad \Delta u_{t-1} \quad \dots \quad \Delta u_{t-n+2}] ,$$

$$\boldsymbol{A}' = \begin{bmatrix} -a_1' & -a_2' & \cdots & -a_{n-1}' & -a_n' & b_2' & b_2' & \cdots & b_{n-1}' & b_n' \\ 1 & 0 & \cdots & 0 & 0 & & & & & \\ 0 & 1 & \cdots & 0 & 0 & & & 0 & & \\ \vdots & \vdots & \cdots & \vdots & \vdots & & & & & \\ 0 & 0 & \cdots & 1 & 0 & & & & & \\ & & & & & 0 & 0 & \cdots & 0 & 0 \\ & & 0 & & & 1 & 0 & \cdots & 0 & 0 \\ & & & & & \vdots & \vdots & \cdots & \vdots & \vdots \\ & & & & & 0 & 0 & \cdots & 1 & 0 \end{bmatrix},$$

$$\boldsymbol{B}' = \begin{bmatrix} b_1' & 0 & 0 & \cdots & 0 & | 1 & 0 & \cdots & 0 \end{bmatrix}.$$

Due to our conservative consideration and the queuing strategy, the control law can be written as: (the reference signal which does not affect the analysis is set to be zero.)

$$\Delta u_t = \boldsymbol{K}_p \boldsymbol{\beta}_{t|p} , \tag{8}$$

where $\boldsymbol{K}_p$ is the gain of GPC controller corresponding to p-step prediction, and $\boldsymbol{\beta}_{t/p}$ is the estimation of $\boldsymbol{\alpha}_t$ based on the recorded measurement of $\boldsymbol{\alpha}_{t-p}$.

And the virtual control law can be written as:

$$\Delta u_{t|0} = \boldsymbol{K}_0 \boldsymbol{\alpha}_t , \tag{9}$$

$$\Delta u_{t|j} = \boldsymbol{K}_j \boldsymbol{\beta}_{t|j} \quad (2 \le j < p) . \tag{10}$$

The virtual control law requires that the control signal is not used for control but is used for prediction ($U_{t-2}[0]$ and $U_{t-2}[1]$ in Fig.4). So $\boldsymbol{\beta}_{t/p}$ can be given by Eqs.(11) and (12):

$$\boldsymbol{\beta}_{t+1|1} = \boldsymbol{A}\boldsymbol{\alpha}_t + \boldsymbol{B}\Delta u_{t|0} , \tag{11}$$

$$\boldsymbol{\beta}_{t+1|j} = \boldsymbol{A}\boldsymbol{\beta}_{t|j-1} + \boldsymbol{B}\Delta u_{t|j-1} \quad (2 \le j \le p) . \tag{12}$$

where Eqs.(11) and (12) are the non-adaptive model used in GPC algorithm. Then the uncertainty can be given as:

$$\begin{aligned} \delta\boldsymbol{A} &= \boldsymbol{A}' - \boldsymbol{A} , \\ \delta\boldsymbol{B} &= \boldsymbol{B}' - \boldsymbol{B} . \end{aligned} \tag{13}$$

As we want to model the performance affected by the predictive error, we define the error as:

$$\boldsymbol{e}_t = \boldsymbol{\alpha}_t - \boldsymbol{\beta}_{t|1} . \tag{14}$$

Let us discuss the situations when p=1 and p=2 are used to demonstrate our basic idea and then give the model of the general case.

Case 1: p=1

Substituting Eqs.(7), (11) and (9) into Eq.(14) yields

$$\begin{aligned} \boldsymbol{e}_t &= \boldsymbol{A}'\boldsymbol{\alpha}_{t-1} + \boldsymbol{B}'\Delta u_{t-1} - (\boldsymbol{A}\boldsymbol{\alpha}_{t-1} + \boldsymbol{B}\Delta u_{t-1|0}) \\ &= \delta\boldsymbol{A}\boldsymbol{\alpha}_{t-1} + \boldsymbol{B}'\Delta u_{t-1} - \boldsymbol{B}\Delta u_{t-1|0} \\ &= \delta\boldsymbol{A}\boldsymbol{\alpha}_{t-1} + \boldsymbol{B}'\Delta u_{t-1} - \boldsymbol{B}(K_0\boldsymbol{\alpha}_{t-1}) \\ &= (\delta\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_0)\boldsymbol{\alpha}_{t-1} + \boldsymbol{B}'\Delta u_{t-1}. \end{aligned} \tag{15}$$

Substituting Eq.(14) into Eq.(8) yields

$$\Delta u_{t-1} = \boldsymbol{K}_1(\boldsymbol{\alpha}_{t-1} - \boldsymbol{e}_{t-1}) . \tag{16}$$

Substituting Eqs.(16) and (8) into Eq.(7) yields

$$\begin{aligned} \boldsymbol{\alpha}_{t+1} &= \boldsymbol{A}'\boldsymbol{\alpha}_t + \boldsymbol{B}'\boldsymbol{K}_1(\boldsymbol{\alpha}_t - \boldsymbol{e}_t) \\ &= (\boldsymbol{A}' + \boldsymbol{B}'\boldsymbol{K}_1)\boldsymbol{\alpha}_t - \boldsymbol{B}'K_1\boldsymbol{e}_t . \end{aligned} \tag{17}$$

Substituting Eq.(15) into Eq.(13) yields

$$\begin{aligned} \boldsymbol{e}_t &= (\delta\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_0)\boldsymbol{\alpha}_{t-1} + \boldsymbol{B}'\boldsymbol{K}_1\boldsymbol{\beta}_{t-1|1} \\ &= (\delta\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_0)\boldsymbol{\alpha}_{t-1} + \boldsymbol{B}'\boldsymbol{K}_1(\boldsymbol{\alpha}_{t-1} - \boldsymbol{e}_{t-1}). \end{aligned} \tag{18}$$

Eqs.(17) and (18) enable us to obtain an augmented state-variable representation Eq.(19) for p=1

$$\begin{bmatrix} \boldsymbol{\alpha}_{t+1} \\ \boldsymbol{e}_{t+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}' + \boldsymbol{B}'\boldsymbol{K}_1 & -\boldsymbol{B}'\boldsymbol{K}_1 \\ (\delta\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_0 + \boldsymbol{B}'\boldsymbol{K}_1) & -\boldsymbol{B}\boldsymbol{K}_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_t \\ \boldsymbol{e}_t \end{bmatrix}. \tag{19}$$

Case 2: $p = 2$

Substituting Eqs.(12), (10) into Eq.(8) yields

$$\begin{aligned} \Delta u_{t-1} &= \boldsymbol{K}_2 \boldsymbol{\beta}_{t-1|2} = \boldsymbol{K}_2(\boldsymbol{A}\boldsymbol{\beta}_{t-2|1} + \boldsymbol{B}\Delta u_{t-2|1}) \\ &= \boldsymbol{K}_2(\boldsymbol{A}\boldsymbol{\beta}_{t-2|1} + \boldsymbol{B}\boldsymbol{K}_1\boldsymbol{\beta}_{t-2|1}) = \boldsymbol{K}_2(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}_1)\boldsymbol{\beta}_{t-2|1} \\ &= \boldsymbol{K}_2(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}_1)(\boldsymbol{\alpha}_{t-2} - \boldsymbol{e}_{t-2}). \end{aligned} \tag{20}$$

Substituting Eq.(20) into Eq.(15) yields:

$$\begin{aligned} \boldsymbol{e}_t &= (\delta\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_0)\boldsymbol{\alpha}_{t-1} + \boldsymbol{B}\boldsymbol{K}_2(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}_1)\boldsymbol{\alpha}_{t-2} \\ &\quad - \boldsymbol{B}'\boldsymbol{K}_2(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}_1)\boldsymbol{e}_{t-2}. \end{aligned} \tag{21}$$

Substituting Eq.(20) into Eq.(7) yields:

$$\boldsymbol{\alpha}_{t+1} = \boldsymbol{A}'\boldsymbol{\alpha}_t + \boldsymbol{B}'\boldsymbol{K}_2(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}_1)\boldsymbol{\alpha}_{t-1} - \boldsymbol{B}'\boldsymbol{K}_2(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}_1)\boldsymbol{e}_{t-1}. \tag{22}$$

Eqs.(21) and (22) enable us to obtain an augmented state-variable representation Eq.(23) for p=2

$$[\boldsymbol{\alpha}_{t+1} \;\; \boldsymbol{\alpha}_t \;\; \boldsymbol{e}_{t+1}]^{\mathrm{T}} = \begin{bmatrix} \boldsymbol{A}' & \boldsymbol{B}'\boldsymbol{K}_2(\boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}_1) & -\boldsymbol{B}'\boldsymbol{K}_2(\boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}_1) \\ \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ (\delta\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}_0) & \boldsymbol{B}\boldsymbol{K}_2(\boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}_1) & -\boldsymbol{B}'\boldsymbol{K}_2(\boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}_1) \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_t \\ \boldsymbol{\alpha}_{t-1} \\ \boldsymbol{e}_{t-1} \end{bmatrix}. \tag{23}$$

Case 3: The general case

Following the ideas of Case 1 and Case 2, we can easily obtain the augmented state-variable representation Eq.(24) for general case.

The above Eqs.(19), (23) and (24) can be divided into nominal and uncertain parts:

$$\boldsymbol{X}_{t+1} = (\boldsymbol{\Psi} + \delta\boldsymbol{\Psi})\boldsymbol{X}_t,$$

where $\boldsymbol{X}_t = [\boldsymbol{\alpha}_{t+1} \;\; \boldsymbol{\alpha}_t \;\; \ldots \;\; \boldsymbol{\alpha}_{t-p+2} \;\; \boldsymbol{\alpha}_{t-p+1} \;\; \boldsymbol{e}_{t+1}]^{\mathrm{T}}$; $\boldsymbol{\Psi}$ is the nominal part which is a constant matrix containing matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\delta\boldsymbol{\Psi}$ is the uncertainty part which may be time-varying.

Note that the state $\boldsymbol{\alpha}_t$ is only related to the error $\boldsymbol{e}_{t-p}$ Eq.(24). Compared to the methodology proposed by (Luck and Ray, 1990), our method is much less affected by the accumulated error. Moreover, Eq.(24) is obtained with the most conservative consideration, so a better dynamics can be achieved during its practical application (Section 5).

CONTROL EXPERIMENTS OVER ZJULAN

Experimental environment

NetLab at Zhejiang University provides remote access to many experimental facilities.

The system architecture of NetLab is shown in Fig.8. Generally speaking, NetLab is comprised of three important parts: client site, server and control site. The network clients can be Intranet clients or Internet clients from all over the world. The control site and server are in the Intranet, communicating with each other in SOCKET. The server is composed of two parts: the main server and the database server. The main server handles user requests, communicates with the database server and the control computer on the control site. The main server can also be accessed via WWW (http://www.netlab.zju.edu.cn). These two servers run on an XEON 2G*2 computer with 2 G RAM.

$$\begin{bmatrix} \boldsymbol{\alpha}_{t+1} \\ \boldsymbol{\alpha}_t \\ \vdots \\ \boldsymbol{\alpha}_{t-p+2} \\ \boldsymbol{\alpha}_{t-p+1} \\ \boldsymbol{e}_{t+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}' & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{B}'\boldsymbol{K}_p \prod_{j=1}^{p-1}(\boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}_j) & -\boldsymbol{B}'\boldsymbol{K}_p \prod_{j=1}^{p-1}(\boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}_j) \\ \boldsymbol{I} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ (\delta\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}_0) & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{B}\boldsymbol{K}_p \prod_{j=1}^{p-1}(\boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}_j) & -\boldsymbol{B}'\boldsymbol{K}_p \prod_{j=1}^{p-1}(\boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}_j) \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_t \\ \boldsymbol{\alpha}_{t-1} \\ \vdots \\ \boldsymbol{\alpha}_{t-p+1} \\ \boldsymbol{\alpha}_{t-p} \\ \boldsymbol{e}_{t-p+1} \end{bmatrix}. \tag{24}$$

There are three significant reasons for us to set up our experiment based on NetLab: (1) The established architecture enables us to set up our experiment more easily. The client site is replaced by the controller of NCS and the controller site is replaced by actuator and sensor of NCS; (2) The opening properties of NetLab enable us to test our proposed algorithm under complex network conditions; (3) We can easily use the Zhejiang University campus-wide network (ZJUnet), which is a local area network (LAN) containing Ethernet at the physical layer. Communication between nodes is done using TCP/IP sockets (at the transport and network layer, respectively).

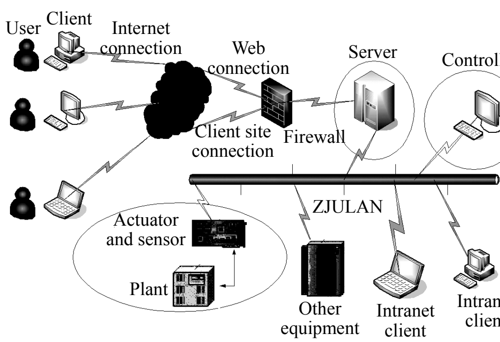

Fig.8 System architecture of NETLAB

Experiments

Three experiments are carried out: (1) a virtual plant controlled by GPC with the exact model; (2) a virtual plant controlled by GPC with a biased model; (3) a real plant controlled by GPC with online estimation.

All experiments show that the proposed method can be applied in practice because of its stability and robustness.

1. Experiment 1

Generally, the delay of ZJULAN occurs abruptly and is less than 0.5 s, which make us choose the prediction region to be 0.5 s.

Remark: (1) the delay is recorded in the controller as: delay=$T(\boldsymbol{U}_t)-T(\boldsymbol{U}_{t-1})$, where $T(\boldsymbol{U}_t)$ is the time instant of $\boldsymbol{U}_t$ arriving at the controller. (2) "gpc+qs" means that GPC algorithm and queuing strategy are combined in NCS and "gpc" means only GPC is used.

In this experiment, a second order system with low damping ratio was chosen to be the plant:

$$G(s)=\frac{K\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}, \tag{25}$$

where K=2, ζ=0.1768 and ω_n=14.1421.

The model can be obtained exactly with sample time T=0.05 s:

$$\boldsymbol{A}(q^{-1})y(t)=\boldsymbol{B}(q^{-1})u(t-1),$$

where $\boldsymbol{A}$=[1 −1.355 0.7788] and $\boldsymbol{B}$=[0.4422 0.4063].

The parameters for the experiment are: prediction horizon N_1=1, N_2=10, control horizon N_u=10, weighting factor λ=5. The interface of the experiment is shown in Fig.9. To show the result of the experiment clearly, the data is re-drawn by MATLAB (Fig.10). Clearly, the delay does not affect the performance so that the response is very similar to that of the real-time control system.

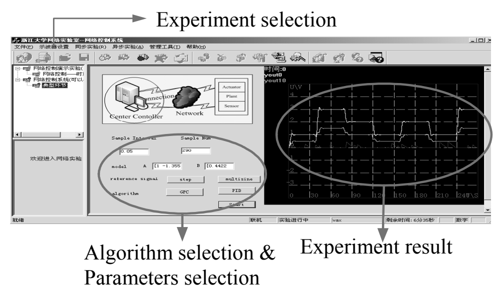

Fig.9 Experiment interface

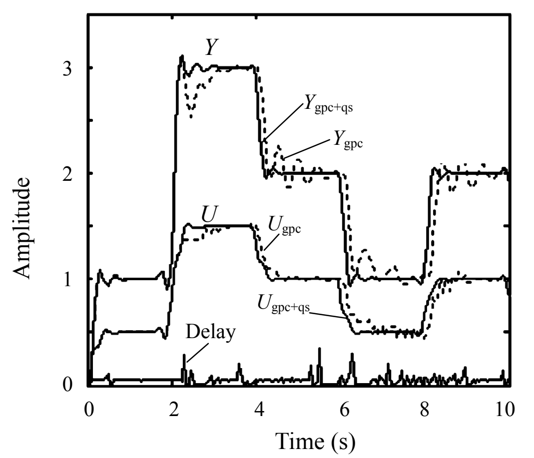

Fig.10 Application of GPC in NCS with an exact model

2. Experiment 2

It is well known that the performance of predictive control depends on the model. The robustness of

GPC was analyzed in Clarke *et al.*(1994), where its robustness both in non-adaptive case and in adaptive case was shown. This experiment aimed at testing the performance with a biased model. The model is artificially selected as $\boldsymbol{A}$=[1 −1.0 0.7] and $\boldsymbol{B}$=[0.4 0.4]. Other parameters selected are the same as those in Experiment 1. The performance is shown in Fig.11. Though the performance is affected by the net delay compared with the real time application, the system remains stable, which indicates that the proposed method inherits the robustness of GPC. Better performance can be achieved by using online identification and the result is given in Experiment 3.

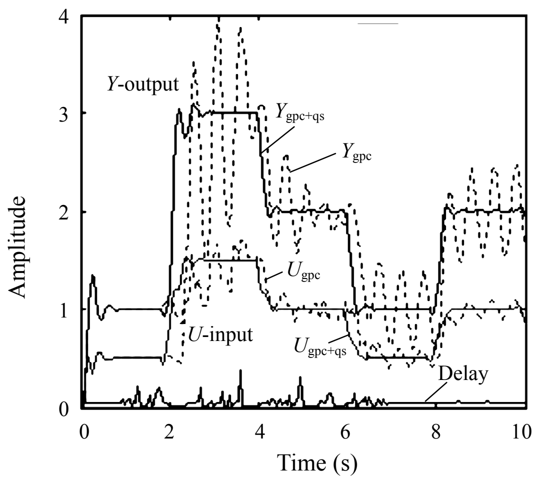

Fig.11 Applied GPC in NCS with a biased model

3. Experiment 3

This experiment was quite different from the above two. A second order system Eq.(25) which consists of amplifiers, resistances and capacitance was selected to be the plant, so noises and model uncertainty were involved. Online identification was used to estimate the parameters of the plant by recursive least squares method. The initial parameters of $\boldsymbol{A}$, $\boldsymbol{B}$ were all set to be zeros.

TCP/IP sockets provided reliable transmission of data packets, regardless of possible collisions that might happen on the physical transmission medium. Therefore, TCP/IP will result in packet delay but not packet loss. To test the effect of package loss, 1% of data were rejected deliberately at some time selected randomly. The result is shown in Fig.12, which indicates that the package loss does not greatly affect the system performance as long as the model can be estimated correctly.

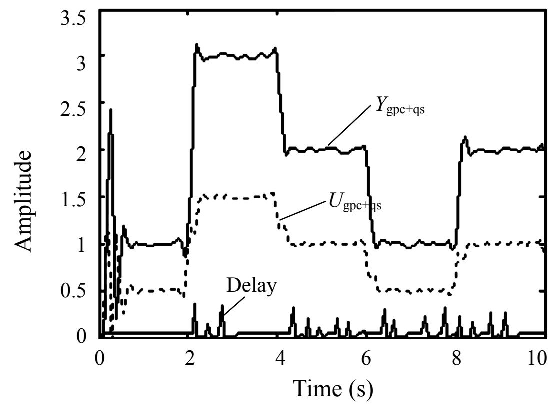

Fig.12 Application of GPC in NCS of adaptive case

CONCLUSION

This article proposes a queuing strategy for GPC application in NCS. The test was carried on ZJULAN and indicated that good performance can be achieved through the proposed method. The advantages of the proposed approach are summarized as follow:

(1) Compared to (Luck and Ray, 1990; 1994), the proposed approach enables NCS to inherit the merits of GPC, such as stability, robustness and good dynamics. The advantage of the proposed queuing strategy is that the existing GPC controller need not be redesigned for use on a network environment.

(2) The proposed approach has an effective framework which does not need clock synchronization, which makes the proposed algorithm simpler and easier than some other approaches.

(3) In the proposed approach, the close-loop system performance is not greatly affected when package loses occur, which enables an easy application in practice.

References

Almutairi, N.B., Chow, M.Y., Tipsuwan, Y., 2001. Network-based Controlled DC Motor with Fuzzy Compensation. The 27th Annual Conference of the IEEE Industrial Electronics Society (IECON 01), **3**:1844-1849.

Branicky, M.S., Phillips, S.M., Zhang, W., 2000. Stability of Networked Control Systems: Explicit Analysis of Delay. Proc. Amer. Control Conf., Chicago, IL, p.2352-2357.

Clarke, D.W., Mohtadi, C., Tuffs, P.S., 1987. Generalized predictive control, Part 1 and Part 2. *Automatica*, **23**(2): 137-160. [doi:10.1016/0005-1098(87)90087-2]

Clarke, D.W., Mosca, E., Scattolini, R., 1994. Robustness of an adaptive predictive controller. *IEEE Trans. AC*, **39**(5):

1052-1056.

Göktas, F., 2000. Distributed Control of Systems over Communication Networks. Ph.D Thesis, University of Pennsylvania.

Halevi, Y., Ray, A., 1988. Integrated communication and control systems: Part I—Analysis. *Dynamic Syst., Measure Contr.*, **110**:67-373.

Luck, R., Ray, A., 1990. An observer-based compensator for distributed delays. *Automatica*, **26**(5):903-908. [doi:10.1016/0005-1098(90)90007-5]

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

Mills, D., 1991. Internet time synchronization: The network time protocol. *IEEE Trans. Commun.*, **39**(10):1482-1493. [doi:10.1109/26.103043]

Nilsson, J., 1998. Real-time Control Systems with Delays. Ph.D Thesis, Dept. Automatic Control, Lund Institute of Technology, Lund, Sweden.

Walsh, G.C., Beldiman, O., Bushnell, L., 1999a. Asymptotic Behavior of Networked Control Systems. Proceedings of the 1999 IEEE International Conference on Control Applications, Kohala Coast, HI, **2**:1448-1453.

Walsh, G.C., Ye, H., Bushnell, L., 1999b. Stability Analysis of Networked Control Systems. Proceedings of the 1999 American Control Conference, San Diego, CA, **4**: 2876-2880.

Zhang W., Branicky, M.S., Phillips, S.M., 2001. Stability of networked control systems. *IEEE Control Systems Magazine*, **21**(1):84-99. [doi:10.1109/37.898794]