Intelligent Scheduling Control of Networked Control Systems with Networked-induced Delay and Packet Dropout

Hongbo Li, Zengqi Sun*, Badong Chen, Huaping Liu, and Fuchun Sun

Abstract: Networked control systems (NCSs) have gained increasing attention in recent years due to their advantages and potential applications. The network Quality-of-Service (QoS) in NCSs always fluctuates due to changes of the traffic load and available network resources. To handle the network QoS variations problem, this paper presents an intelligent scheduling control method for NCSs, where the sampling period and the control parameters are simultaneously scheduled to compensate the effect of QoS variation on NCSs performance. For NCSs with network-induced delays and packet dropouts, a discrete-time switch model is proposed. By defining a sampling-period-dependent Lyapunov function and a common quadratic Lyapunov function, the stability conditions are derived for NCSs in terms of linear matrix inequalities (LMIs). Based on the obtained stability conditions, the corresponding controller design problem is solved and the performance optimization problem is also investigated. Simulation results are given to demonstrate the effectiveness of the proposed approaches.

Keywords: Linear matrix inequalities (LMIs), networked control systems (NCSs), network-induced delay, packet dropout, quality-of-service (QoS), stability.

1. INTRODUCTION

Networked control systems (NCSs) are spatially distributed control systems with control loops closed via communication networks. Using NCSs has many advantages over the traditional control systems, such as reduced system wiring, simple installation, increased system flexibility, and the great benefits from sharing of the resources. Therefore, NCSs have been finding many application areas such as DC motors [1,2], robots [3], vehicles [4], car suspension system [5], ball maglev system [6,7], etc. However, the use of a shared network, in contrast to using dedicated wiring, raises new challenging problems for the analysis and design of NCSs such as network-induced delay and packet dropout. As a result, conventional control theories with assumptions such as non-delayed signals and no lost of information, must be re-evaluated before applying to NCSs. Recently, NCSs have attracted much attention from research communities. Issues such as network-induced delay [1-3,6,7], packet dropout [2,3,6,7], network constraints [8,9], scheduling [10,11], and the signal quantization [12,13], have been investigated with results reported in the literature. In general, these results can be classified into two main approaches. One approach is to design the NCSs without considering the presence of the network, and then study the network protocol or scheduling strategy to guarantee certain qualitative properties that the network should possesses [14]. The other approach is to take the network effects into account explicitly and study control methodologies to accommodate the unpredictable nature of network information transmission [1-3].

Network-induced delays and packet dropouts are two major causes for the NCSs performance deterioration and potential NCSs instability. In the literature, some important control methodologies, such as stochastic optimal control [6], robust control [7], predictive control [1], and state feedback control [15], have been proposed to address the problems of network-induced delays or packet dropouts. For more details on this topic, please refer to [1,6,7,15] and the reference therein. It is noticed that most of the existing control methodologies for NCSs adopt the constant control parameters and sampling period regardless of network Quality-of-Services (QoS) variations. In practical circumstances, the network QoS always fluctuates due to changes of the traffic load and

Manuscript received November 8, 2006; revised September 27, 2007; accepted April 16, 2008. Recommended by Editorial Board member Bin Jiang under the direction of Editor Young-Hoon Joo. This work was jointly supported by the National Science Foundation of China (Grant No: 60574035, 60674053) and the National Key Project for Basic Research of China (Grant No: 2002cb312205).

Hongbo Li, Zengqi Sun, Badong Chen, Huaping Liu, and Fuchun Sun are with Department of Computer Science and Technology, State Key Laboratory of Intelligent Technology and Systems, Tsinghua University, Beijing, P. R. China (e-mails: Hb-li04@mails.tsinghua.edu.cn, szq-dcs@mail.tsinghua.edu.cn, chenbd04@mails.tsinghua.edu.cn, hpliu@mail.tsinghua.edu.cn, fcsun@mail.tsinghua.edu.cn).

* Corresponding author.

available network resources. NCSs are functionally related system and their performance depends not only on the control algorithm but also on the network QoS. Therefore, despite the progress made in NCSs controller design, it has become evident that advanced control methodologies with QoS variation compensation are required.

Unfortunately, there has only been a limited amount of research on control methodologies with the QoS variation compensation taking into account. [11] proposed gain scheduling controller for NCSs, where the control parameters are adjusted on-line based on network QoS variations. [10] optimized the control parameters for gain scheduling controller to improve the NCSs performance. It is noticed that those works focused primarily on controller design while the NCSs stability analysis problem have not been addresses. Therefore, the stability analysis for NCSs with scheduled control parameters still remains as an open research area. Moreover, it is well known that both control parameters and sampling period are closely related to the NCSs performance. Therefore, it seems more suitable for us to construct an controller with the sampling period and control parameters simultaneously scheduled to further improve the NCSs performance.

Motivated by the above observations, we focus on investigating the intelligent scheduling control for NCSs in this paper, where the control parameters and sampling period are simultaneously adjusted on-line according to network QoS variations to improve the control performance of NCSs. For a class of NCSs with network-induced delays and packet losses, this paper proposes a discrete-time switch model to describe the closed-loop system. The proposed NCSs model describes the NCS as a switch system and enables us to apply the theory from switch systems to study NCSs in the discrete-time domain. In the framework of the given model, the stability conditions are derived for NCSs in terms of linear matrix inequalities (LMIs). Based on the obtained stability conditions, the corresponding controller design problem is solved and the performance optimization problem is also investigated.

This paper is organized as follows. The problem statement and some definitions are formulated in Section 2. The structure of intelligent scheduling controller is presented in Section 3. The stabilization problem of NCSs is solved in Section 4 and the performance optimization problem is investigated in Section 5. Numerical examples are provided in Section 6 and conclusions are given in Section 7.

Notation: Throughout this paper, $\mathbb{R}^n$ and $\mathbb{R}^{n\times m}$ denote the n dimensional Euclidean space and the set of all $n\times m$ real matrices respectively. $\|\cdot\|$ refers to the Euclidean norm for vectors and induced 2-norm for matrices. The superscript "T" denotes matrix transposition; and for symmetric matrices X and Y, the notation $X > Y$ means that $X - Y$ is positive definite. I is the identity matrices with appropriate dimensions, and the notation $\mathbb{Z}^+$ stands for the set of nonnegative integers. Finally, in symmetric block matrices, we use "$*$" as an ellipsis for the terms introduced by symmetry.

2. PROBLEM STATEMENT

The NCS considered in this paper is depicted in Fig. 1, where the sensor and the controller are connected by communication networks. The dynamics of the controlled plant is given by

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t), \end{cases} \tag{1}$$

where $x(t) \in R^n$ denotes the plant state; $y(t) \in R^p$ represents the output of the plant; $u(t) \in R^m$ means the control input of the plant. A, B, C are matrices with appropriate dimensions.

In the considered NCSs, the sensor is time-driven. At each sampling period, the output of the plant and its timestamp (i.e., the time the plant output is sampled) are encapsulated into a packet and sent to the controller via the network. The timestamp will ensure the controller compute the network-induced delay for the sensor packets, and correspondingly select the right one to compute the control signal. It is noticed that, to obtain better NCSs performance, the sampling period may be adjusted on-line according to some mechanism.

In practice, the data packets in NCSs usually suffer network-induced delay and packet dropout during the network transmissions. The network-induced delay here means the sensor-to-controller delay and it may be longer than the sampling period h. It will be shown that, for the k-th packet from the sensor, if it encounters a delay smaller than h, it will be used to compute the control signal at time instant $(k+1)h$. Otherwise, at time instant $(k+1)h$, the proposed controller will use the predicted control signal to control plant in such a way to compensate long time delay and packet dropout. Note that for the packets from the sensor, only the sensor packets with delay

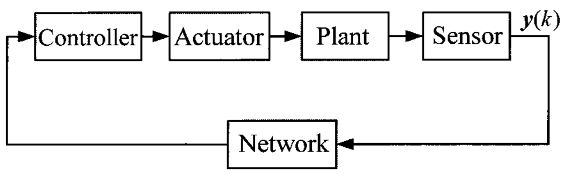


Fig. 1. The structure of NCSs.

smaller than h are used by the proposed controller. Therefore, the packets with delay longer than h can be considered as dropped packet for the NCS under investigation. Based the above observations, we introduce the following definition to capture the nature of packet dropouts in NCSs.

Definition 1: The packet from the sensor is called *effective packet* if its encountered delay is shorter than the sampling period h. Let $S \triangleq \{i_1, i_2, \cdots\} \subseteq \{0, 1, 2, \cdots\}$ denote the sequence of time index of the *effective packets*, then the packet dropout process is defined as

$$\{\eta(i_m) \triangleq i_{m+1} - i_m,\quad i_m \in S\}, \tag{2}$$

which means that, from i_m to i_{m+1}, the number of dropped packets is $\eta(i_m) - 1$. Especially, when two consecutive sensor packets are *effective sensor packets*, we have $\eta(i_m) = 1$, which means no packet is dropped because $\eta(i_m) - 1 = 0$. For the sake of analysis, we define $N_{drop} \triangleq \max_{i_m \in S}\{\eta(i_m)\}$. Then we can conclude that $\eta(i_m)$ takes values in a finite set $\Omega \triangleq \{1, 2, \cdots, N_{drop}\}$.

Let τ_{i_m} express the network-induced delay encountered by the i_mth effective packet. Without loss of generality, in this paper, we assume $\tau_{\min}$, the lower bound of τ_{i_m}, is smaller than h. By considering the definition of effective packet, we have $\tau_{i_m} \in U \triangleq [\tau_{\min}, h)$. Obviously, with the effective packets in NCSs, there exists a sequence of pairs $\{\eta(i_m), \tau_{i_m}\}$. Therefore, the time delay and packet dropout information can be embodied as

$$\{\{\eta(i_m), \tau_{i_m}\} : i_m \in S,\ \tau_{i_m} \in U,\ \eta(i_m) \in \Omega\}. \tag{3}$$

The objective of this paper is to construct an intelligent scheduling controller for the NCSs under investigation, where the control parameters and sampling period are simultaneously adjusted on-line according to network QoS variations.

3. INTELLIGENT SCHEDULING CONTROLLER STRUCTURE

This section will present the structure of the intelligent scheduling controller. The basic idea of constructing the intelligent scheduling controller is as follows. The network QoS of NCSs is divided into different levels first. Then, for different network QoS levels, we design different sampling period and control parameters. Finally, the intelligent scheduling

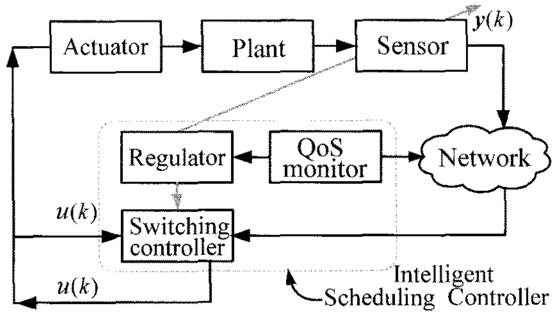


Fig. 2. The controller structure.

controller will schedule the sampling period and control parameters on-line for NCSs based on the network QoS variation. As shown in Fig. 2, the intelligent scheduling controller can be divided into three parts: 1) the QoS monitor; 2) the switching controller; and 3) the regulator. Each component is described in the following sections.

3.1. Network QoS partition and QoS monitor design

There are different ways to define the QoS for sensor-to-controller network. In this paper, one of the most popular QoS measures is used and defined as follows.

- T_{delay} denotes the point-to-point network-induced delay in sensor-to-controller network; it is used to indicate how long a packet is expected to be delivered from the sensor to the controller.

Let the upper and lower bounds of T_{delay} in NCSs be T^{max} and T^{min}, then the universe of discourse of network QoS can be described as $Q \triangleq \{[T^{min}, T^{max}]\}$. Divide Q into N different subsets denoted as $Q_i \triangleq \{[\check{T}_{\text{delay}}(i), \hat{T}_{\text{delay}}(i)]\}$, where i is a piecewise constant function called a switching signal taking values in a finite set $\mathbb{S} \triangleq \{1, 2, \cdots, N\}$, $\check{T}_{\text{delay}}(i)$ and $\hat{T}_{\text{delay}}(i)$ are the upper and lower bounds of Q_i. The partitioning of Q satisfies

$$Q_1 \cup Q_2 \cup \cdots \cup Q_{\mathrm{N}_r} = Q. \tag{4}$$

Thus far, we have divided the universe of discourse of network QoS into N different levels. Corresponding to each QoS level Q_i, T_{delay} is varying in deterministic intervals defined by the switch $\{i\}$, i.e., $T_{\text{delay}} \in [\check{T}_{\text{delay}}(i), \hat{T}_{\text{delay}}(i)]$. Moreover, similar to the discussion in Definition 1, we define the maximum consecutive packet dropout in Q_i as $N^i_{\text{drop}} \triangleq \max_{i_m \in S}\{\eta(i_m)\}$. Then one can infer that, for each network QoS level Q_i, $\eta(i_m)$ takes values in a finite set $\Omega_i \triangleq \{1,2,\cdots,N^i_{\text{drop}}\}$.

In the proposed intelligent scheduling controller, the function of the QoS monitor unit is to determine which level the current network QoS belongs to. In the NCSs, each sensor packet is marked with the time when the plant output is sampled (i.e., the timestamp). Therefore, under the condition that the sensor and the controller are synchronized, the QoS monitor unit can easily calculate the network-induced delay for the arriving sensor packet by simply comparing the timestamp of the sensor packet with the clock of the controller node. With the detected delay information, the QoS monitor can determine which level the current network QoS belongs to. The information on the current network QoS level is then utilized by the switching controller to schedule the sampling period and control parameters.

3.2. The switching controller design

In the switching controller, there exists a special sampling period h_i for each network QoS level Q_i, where $i \in \mathbb{S}$. Denote the m-th sampling period relative to initial time as $h_{i,m}$, where $i \in \mathbb{S}$ and the value of i is determined by the network QoS condition at the m-th sampling period. To simplify writing, we denote the sum of the first k sampling periods as $\overline{h}_k \triangleq \sum_{m=1}^{k} h_{i,m}$. Let $Q(\overline{h}_k)$ denote the network QoS condition at time instant $\overline{h}_k$, $t_{sc}(k)$ denote the network-induced delay encountered by the k-th packet from the sensor. Noting that only effective packets are used by the controller, we can assume that $t_{sc} = \infty$ if a packet is dropped or encounters delay larger than the current sampling period. Then the switching controller is designed as

$$\begin{cases} \text{Switching rule } i: \text{ If } Q(\overline{h}_k) \text{ is } Q_i, \text{ then} \\ \quad \text{Local controller rules:} \\ \quad\quad \text{Rule 1: If } t_{sc} \in [\tau_{\min} \quad h_i), \text{ then} \\ \quad\quad\quad \hat{x}(\overline{h}_k + h_i) = F_i \hat{x}(\overline{h}_k) + G_i u(\overline{h}_k) \\ \quad\quad\quad\quad - K_i[y(\overline{h}_k) - C\hat{x}(\overline{h}_k)] \\ \quad\quad\quad u(\overline{h}_k + h_i) = -L_i \hat{x}(\overline{h}_k + h_i) \\ \quad\quad \text{Rule 2: If } t_{sc} = +\infty, \text{ then} \\ \quad\quad\quad \hat{x}(\overline{h}_k + h_i) = F_i \hat{x}(\overline{h}_k) + G_i u(\overline{h}_k) \\ \quad\quad\quad u(\overline{h}_k + h_i) = -L_i \hat{x}(\overline{h}_k + h_i), \end{cases} \tag{5}$$

where $i \in \mathbb{S}$, F_i and G_i are given as

$$F_i = e^{Ah_i}, \quad G_i = \int_0^{h_i} e^{A\tau} d\tau B. \tag{6}$$

The switching controller is time-driven. At each sampling period, the switching controller compute the control signal for the plant and then send it to the actuator. Immediately after receiving the control signal, the actuator will use the arriving control signal to control the plant.

Discretize the continuous-time plant (1) at the sampling instants as

$$\begin{cases} \text{Switching rule } i: \text{ If } Q(\overline{h}_k) \text{ is } Q_i, \text{ then} \\ \quad \begin{cases} x(\overline{h}_k + h_i) = F_i x(\overline{h}_k) + G_i u(\overline{h}_k) \\ y(\overline{h}_k + h_i) = Cx(\overline{h}_k + h_i), \end{cases} \end{cases} \tag{7}$$

where $i \in \mathbb{S}$, C is of form (1), F_i and G_i are of form (6).

Let $e(\overline{h}_k) \triangleq x(\overline{h}_k) - \hat{x}(\overline{h}_k)$ and introduce $z(\overline{h}_k) = \left[x(\overline{h}_k)^T \ e(\overline{h}_k)^T\right]^T$ into (7) and (5), then we have

$$\begin{cases} \text{Switching rule } i: \text{ If } Q(\overline{h}_k) \text{ is } Q_i, \text{ then} \\ \quad \text{Rule 1: If } t_{sc} \in [\tau_{\min} \quad h_i), \text{ then} \\ \quad\quad z(\overline{h}_k + h_i) = \begin{bmatrix} F_i - G_i L_i & G_i L_i \\ 0 & F_i + K_i C \end{bmatrix} z(\overline{h}_k) \\ \quad \text{Rule 2: If } t_{sc} = +\infty, \text{ then} \\ \quad\quad z(\overline{h}_k + h_i) = \begin{bmatrix} F_i - G_i L_i & G_i L_i \\ 0 & F_i \end{bmatrix} z(\overline{h}_k). \end{cases} \tag{8}$$

It is notice that the sensor packets during the two effective packets are not used to compute the control signal. It means that, for $i_m + 1 \le k \le i_{m+1} - 1$, we have $t_{sc}(k) = +\infty$. Based on this observation, it is obvious that (8) can be represented as

$$\begin{aligned} &\text{Switching rule } i: \text{ If } Q(\overline{h}_{i_m}) \text{ is } Q_i, \text{ then} \\ &\quad z(\overline{h}_{i_{m+1}+1}) = M_i z(\overline{h}_{i_m+1}) \\ &\quad\quad = \begin{bmatrix} \Pi_i^{11} & \Pi_i^{12} \\ 0 & \Pi_i^{22} \end{bmatrix} z(\overline{h}_{i_m+1}), \end{aligned} \tag{9}$$

where

$$\Pi_i^{11} = (F_i - G_i L_i)^{\eta(i_m)},$$

$$\Pi_i^{22} = (F_i + K_i C) F_i^{\eta(i_m)-1},$$

$$\Pi_i^{12} = \sum_{n=0}^{\eta(i_m)-1} \left[(F_i - G_i L_i)^{\eta(i_m)-n-1} G_i L_i F_i^{\,n}\right].$$

For simplicity of notation, we rewrite (9) into the following *discrete-time switch system*:

$$z(\overline{h}_{i_{m+1}+1}) = M_{r(i_m)} z(\overline{h}_{i_m+1}), \tag{10}$$

where $M_{r(i_m)}$ are matrices upon the switch $r(i_m)$, which takes values in $\mathbb{S}$. For each possible value of $r(i_m)=i$, we have $M_{r(i_m)}=M_i$, where M_i is of form (9).

Moreover, for $i_m+1<l<i_{m+1}+1$, the dynamics of NCSs can be expressed as

$$z(\bar{h}_l)=\bar{M}_{r(i_m)}z(\bar{h}_{i_m+1}), \tag{11}$$

where $\bar{M}_{r(i_m)}$ are matrices upon the switch $r(i_m)$ and of following form

$$\bar{M}_i=\begin{bmatrix}(F_i-G_iL_i)^{(l-i_m-1)} & \Pi_i\\ 0 & F_i^{(l-i_m-1)}\end{bmatrix} \tag{12}$$

with $\Pi_i=\sum_{n=0}^{l-i_m-2}\left[(F_i-G_iL_i)^{l-i_m-n-2}G_iL_iF_i^{\,n}\right]$.

3.3. The regulator design

Based on the controller design approach introduced in the following two sections, h_i, L_i and K_i can be designed for each QoS level Q_i, where $i\in\mathbb{S}$. Correspondingly, a table with respect to Q_i, h_i, L_i and K_i can be formed. Then, lookup table technique can be used in the regulator to schedule the sampling period and control parameters on-line according to the network QoS condition information provided by QoS monitor.

4. STABILIZATION OF NCS

4.1. Stability analysis of NCSs

This section will analyze the stability property of NCSs, where the stability analysis problem can be stated as follows.

Problem 1 (Stability analysis)**:** Given a plant (1) and a networked controller (5), check whether the closed-loop NCS (10) is asymptotically stable over networks with given parameters D^i_{drop}, where $i\in\mathbb{S}$.

In the following theorem and corollary, sufficient stability conditions are derived for NCS (10) via both sampling-period-dependent Lyapunov function and common quadratic Lyapunov function.

Theorem 1: Let Q_i be a given set of network QoS levels, and let h_i, L_i, K_i be the given parameters for the intelligent scheduling controller, where $i\in\mathbb{S}$. The NCS (10) over networks with given parameters D^i_{drop} is asymptotically stable, if for $i,j\in\mathbb{S}$, there exist positive definite matrices $P_i\in\mathbb{R}^{2n\times 2n}$, $P_j\in\mathbb{R}^{2n\times 2n}$, satisfying

$$M_i^{\,T}P_jM_i-P_i<0, \tag{13}$$

where M_i is of form (9) with $\eta(i_m)\in\Omega_i$.

Proof: For the closed-loop NCS (10), let a sampling-period-dependent Lyapunov function candidate be:

$$V(\bar{h}_{i_m+1})=z^T(\bar{h}_{i_m+1})P_{r(i_m)}z(\bar{h}_{i_m+1}), \tag{14}$$

where $P_{r(i_m)}$ are matrices upon the switching $r(i_m)$ in NCSs model (10) and $r(i_m)$ takes values in $\mathbb{S}$.

Let $i\triangleq r(i_m)$ and $j\triangleq r(i_{m+1})$, then we have $V(\bar{h}_{i_m+1})=z^T(\bar{h}_{i_m+1})P_iz(\bar{h}_{i_m+1})$ and $V(\bar{h}_{i_{m+1}+1})=z^T(\bar{h}_{i_m+1})M_i^{\,T}P_jM_iz(\bar{h}_{i_m+1})$. From the conditions (13), one can easily show that

$$\begin{aligned}\Delta V&=V(\bar{h}_{i_{m+1}+1})-V(\bar{h}_{i_m+1})\\&=z^T(\bar{h}_{i_m+1})[M_i^{\,T}P_jM_i-P_i]z(\bar{h}_{i_m+1})<0,\end{aligned} \tag{15}$$

where $z(\bar{h}_{i_m+1})\neq 0$.

First, let us prove that the closed-loop NCS (10) is stable if the conditions (13) hold. That is, given any $\varepsilon>0$, we can find a $\delta(\varepsilon)>0$ such that $\|z(0)\|<\delta(\varepsilon)$ implies $\|z(\bar{h}_l)\|<\varepsilon$ for $l\in\mathbb{Z}^+$. For this purpose, introduce $\alpha_1\triangleq\max\{\max_{i\in\mathbb{S}}\|M_i\|,1\}$, $\alpha_2\triangleq\max_{i\in\mathbb{S}}\|\bar{M}_i\|$, $\alpha_3\triangleq\max_{i\in\mathbb{S}}\|P_i\|$ and $\alpha_4\triangleq\min_{i\in\mathbb{S}}\left\{1/\|P_i^{-1}\|\right\}$. From the definition of Lyapunov function, we can get $\alpha_4\|\mathbf{z}(\bar{h}_{i_m+1})\|^2\le V(\bar{h}_{i_m+1})\le\alpha_3\|\mathbf{z}(\bar{h}_{i_m+1})\|^2$. In NCS (10), three cases arise and are discussed as follows.

Case 1: $0\le l<i_1+1$, where i_1 is the time index of the first effective packet. Let the NCS initial states be $x(0)$, $\hat{x}(0)$ and $\mathbf{z}(0)=\left[x(0)^T\ e(0)^T\right]^T$. Then the dynamics of the NCS in this case is given by $z(\bar{h}_l)=\bar{M}_{r(0)}z(0)$. Given any $\varepsilon>0$, if let $\|z(0)\|<\min\{1/\alpha_2,1\}\varepsilon$, then we have $\|z(\bar{h}_l)\|\le\|\bar{M}_{r(0)}\|\|z(0)\|<\|\bar{M}_{r(0)}\|\min\{1/\alpha_2,1\}\varepsilon<\varepsilon$.

Case 2: $l=i_m+1$. In this case, we have $z(\bar{h}_{i_1+1})=M_{r(0)}z(0)$. From previous discussions and (15), we can conclude that $V(\bar{h}_l)<V(\bar{h}_{i_1+1})\le\alpha_3\|z(\bar{h}_{i_1+1})\|^2\le\alpha_3\|M_{r(0)}\|^2\|z(0)\|^2$ and $\|z(\bar{h}_l)\|\le\sqrt{1/\alpha_4V(\bar{h}_l)}$. Therefore, for any given $\varepsilon>0$, letting

$\|z(0)\| < \sqrt{\alpha_4/(\alpha_3\alpha_1^2)}\varepsilon$ will lead to $\|z(\bar{h}_l)\| \le \sqrt{\alpha_3/\alpha_4}\|M_{r(0)}\|\|z(0)\| \le \sqrt{(\alpha_1^2\alpha_3)/\alpha_4}\|z(0)\| < \varepsilon$.

Case 3: $i_m+1 < l < i_{m+1}+1$. Given any $\varepsilon > 0$, if let $\|z(0)\| < \sqrt{\alpha_4/(\alpha_3\alpha_1^2\alpha_2^2)}\varepsilon$, then from (11) we can get $\|z(\bar{h}_l)\| = \|\bar{M}(r(i_m))z(\bar{h}_{i_m})\| \le \alpha_2\|z(\bar{h}_{i_m})\| \le \alpha_2\sqrt{(\alpha_1^2\alpha_3)/\alpha_4}\|z(0)\| < \varepsilon$.

Based on the above analysis, if we let $\alpha \triangleq \min\left\{\min\{1/\alpha_2, 1\}, \sqrt{\alpha_4/(\alpha_3\alpha_1^2)}, \sqrt{\alpha_4/(\alpha_3\alpha_1^2\alpha_2^2)}\right\}$, then we can conclude that $\|z(0)\| < \alpha\varepsilon$ implies $\|z(\bar{h}_l)\| < \varepsilon$ for $l \in \mathbb{Z}^+$.

We now prove that $\lim_{l\to\infty}\|z(\bar{h}_l)\| = 0$ for NCS (10). From (15), we can get $\lim_{(\bar{h}_{i_m+1})\to\infty} V(\bar{h}_{i_m+1}) = 0$, which implies $\lim_{(\bar{h}_{i_m+1})\to\infty} z(\bar{h}_{i_m+1}) = 0$. On the other hand, noting that $\|z(\bar{h}_l)\| = \|\bar{M}_{r(i_m)}z(\bar{h}_{i_m})\| \le \alpha_2\|z(\bar{h}_{i_m})\|$ for $i_m+1 < l < i_{m+1}+1$, we can get $\lim_{l\to\infty}\|z(\bar{h}_l)\| = 0$ for $l \ne i_m+1$. Summarizing the above two cases leads to the conclusion that $\lim_{l\to\infty}\|z(\bar{h}_l)\| = 0$ for NCS (10), where $l \in \mathbb{Z}^+$.

According to the definition of asymptotically stable, we can complete the proof. □

Via common quadratic Lyapunov function, a simpler condition that checks whether the NCS (10) is asymptotically stable can be obtained as follows.

Corollary 1: Let Q_i be a given set of network QoS levels, and let h_i, L_i, K_i be the given parameters for the intelligent scheduling controller, where $i \in \mathbb{S}$. The NCS (10) over networks with given parameters D_{drop}^i is asymptotically stable, if for $i \in \mathbb{S}$, there exists a common positive definite matrix $P \in \mathbb{R}^{2n\times 2n}$ satisfying

$$M_i^T P M_i - P < 0, \tag{16}$$

where M_i is of form (9) with $\eta(i_m) \in \Omega_i$.

Proof: It directly follows from Theorem 1. □

Apparently, Theorem 1 is based on the sampling-period-dependent Lyapunov function, while Corollary 1 is obtained via the common quadratic Lyapunov function. Thus, the stability conditions of Theorem 1 are less conservative than Corollary 1. However, Corollary 1 provides a simpler way to check whether the NCS (10) is asymptotically stable. Note that the conditions of Theorem 1 and Corollary 1 are all LMIs, which can be readily checked by using standard numerical software such as the LMI toolbox in MATLAB.

4.2. Stabilization of NCS

The previous section presents the stability conditions for NCSs. An immediate question is whether the obtained stability conditions can be further extended to cope with the controller design problem, which is stated as follows.

Problem 2 (The controller design): Give a given set of network QoS levels Q_i and a plant in the form of (1), where $i \in \mathbb{S}$, design h_i, L_i, K_i for the intelligent scheduling controller such that the closed-loop NCS (10) over networks with given parameters D_{drop}^i is asymptotically stable.

Note that it is not easily to calculate h_i, L_i and K_i directly from the conditions of Theorem 1 (or Corollary 1) by convex optimization techniques. The main difficulty lies in the fact that, the selection of the best sampling period for NCSs is a compromise due to the interaction of networks [16], and the conditions of Theorem 1 (or Corollary 1) are nonlinear matrix inequalities (NMI) due to $(F_i - G_iL_i)^j$. To circumvent the synthesis problem, we propose a three-step design procedure for the concerned NCSs, which enables us to design h_i, L_i and K_i one by one.

4.2.1 The sampling period selection

As discussed in [16], the NCSs performance chart versus the sampling period can be derived as Fig. 3. Point A comes as no surprise since NCSs are a type of digital control system. As sampling period gets smaller, the traffic load in the network becomes heavier. Correspondingly, more packets are dropped and longer network-induced delays result. This situation causes the existence of Point B in NCSs. As a result, there exists a sampling period range (h_B, h_A) for the NCSs. When the sampling period is

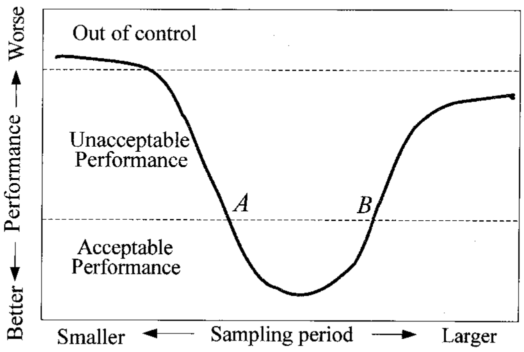


Fig. 3. The NCSs performance vs. the sampling period.

within (h_B, h_A), the control performance of NCSs is acceptable.

Based on the above discussions, one can infer that, to guarantee the control performance of NCSs, h_i should be selected within the sampling period range (h_A^i, h_B^i). Therefore, the h_i design problem is casted into the h_A^i and h_B^i computing problem, which is addressed as follows.

- The computing of h_A^i

Let w_{bw} denote the control system bandwidth, which is defined as the maximum frequency at which the output of a system will track an input sinusoid in a satisfactory manner. Alternatively, the control system bandwidth is the frequency of input at witch the output is attenuated to a factor of 0.797 times the input (or down 3 dB) [17]. In order to guarantee the control performance, the "rule of thumb" for selecting sampling period in digital control is that the desired sampling multiple is [17]

$$20 \le \frac{w_s}{w_{bw}} \le 40, \tag{17}$$

where w_s is the sampling frequency (or sampling rate) in digital control.

Then, by considering (17) and by considering the effect of network-induced delay, the sampling period of Point A for Q_i, i.e., h_A^i, can be computed by

$$h_A^i = \frac{1}{20 w_{bw}} - \hat{T}_{\text{delay}}(i), \tag{18}$$

where $\hat{T}_{\text{delay}}(i)$ is the upper bound of network-induced delay in Q_i, as discussed in Section 3.

- The computing of h_B^i

As discussed in [16], h_B^i can be computed by

$$h_B^i = \frac{T_{ttt}^i}{0.696}, \tag{19}$$

where T_{ttt}^i denotes the total transmission time of all cyclic messages in the network. In a more concrete sense, let n be the number of devices in the network, and define the transmission time of each message as T_{tx}^j, then we have $T_{ttt} = \Sigma_{i=j}^{2n} T_{tx}^j$ for strobe connection network, and have $T_{ttt} = \Sigma_{j=1}^{(n+1)} T_{tx}^j$ for poll connection network. For more details on this topic, please refer to [16].

For network with other connection type or with T_{ttt}^i being unavailable, an estimation of T_{ttt}^i is given by

$$T_{ttt}^i = 0.5(\check{T}_{\text{delay}}(i) + \hat{T}_{\text{delay}}(i)) - \tau_{\min}, \tag{20}$$

where $\tau_{\min}$ is the lower bound of network-induced delay in NCS, as discussed in Section 2; $\check{T}_{\text{delay}}(i)$ and $\hat{T}_{\text{delay}}(i)$ are the lower and upper bounds of network-induced delay in Q_i, as discussed in Section 3.

Thus far, we have provided the method to compute the h_A^i and h_B^i for NCSs. Based on the presented method, we can obtain the sampling period range (h_B^i, h_A^i) for NCSs, where $i \in \mathbb{S}$. Then we can select the sampling period h_i within the sampling period range (h_B^i, h_A^i).

4.2.2 The feedback gain matrix design

With h_i designed from the first stage, this section will address the feedback gain matrix design problem. Note that a matrix is called schur stable if all its eigenvalues are placed in the unit circle. Then the necessary conditions under which the intelligent scheduling controller will stabilize the NCSs (10) are established as follows.

Theorem 2: For a set of given parameter D_{drop}^i, if for $i \in \mathbb{S}$, the conditions of Theorem 1 or Corollary 1 are satisfied (i.e., there exist positive definite matrices $P_i \in \mathbb{R}^{2n \times 2n} > 0$ such that (13) hold, or there exists a common positive definite matrix $P \in \mathbb{R}^{2n \times 2n} > 0$ such that (16) holds), then the matrices $(F_i - G_i L_i)^j$ are schur stable, where $j \in \Omega_i$.

Proof: This can be proved by contradiction. Assume that one eigenvalue of $(F_i - G_i L_i)^j$ is outside the unit circle, where $j \in \Omega_i$. By considering (9), the characteristic polynomial for M_j can be given by

$$|M_j| = \left| zI - \begin{bmatrix} \Pi_j^{11} & \Pi_j^{12} \\ 0 & \Pi_j^{22} \end{bmatrix} \right| = \left| \begin{bmatrix} zI - \Pi_j^{11} & -\Pi_j^{12} \\ 0 & zI - \Pi_j^{22} \end{bmatrix} \right|. \tag{21}$$

Since $|M_j|$ is a block triangular matrix, its determinant can be written as

$$\begin{aligned} |M_j| &= \left|zI - \Pi_j^{11}\right| \left|zI - \Pi_j^{22}\right| \\ &= \left|zI - (F_j - G_j L_j)^{\eta(i_m)}\right| \left|zI - (F_j + K_j C) F^{\eta(i_m) - 1}\right|. \end{aligned} \tag{22}$$

By considering the assumption that one eigenvalue of $(F_i - G_iL_i)^j$ is outside the unit circle, one can conclude that there do not exist positive definite matrices $P_i \in \mathbb{R}^{2n\times 2n} > 0$ such that (13) hold, or there does not exist a common positive definite matrix $P \in \mathbb{R}^{2n\times 2n} > 0$ such that (16) holds. That contradicts with the conditions of this theorem. Therefore, the matrices $(F_i - G_iL_i)^j$ must be schur stable. □

Theorem 2 enlightens us to design L_i by guaranteeing that $(F_i - G_iL_i)^j$ are schur stable for all $j \in \Omega_i$. Note that A being schur stable implies A^j being schur stable. Therefore we can design the L_i just by guaranteeing that $F_i - G_iL_i$ is schur stable. This enables us to employ pole placement or other control design methods to design L_i. Those methods are available in the literature, and so they are omitted.

4.2.3 The observer gain matrix design

With h_i and L_i designed via the above stages, the observer gain matrix K_i design technique is provided as follows.

Theorem 3: For a given NCS (10) and a set of given parameters L_i and h_i, if for $i \in \mathbb{S}$, there exist symmetric matrices $P_{11i} \in \mathbb{R}^{n\times n}$, $P_{22i} \in \mathbb{R}^{n\times n}$ and matrices $P_{21i} \in \mathbb{R}^{n\times n}$, $\mathbb{X}_{1i} \in \mathbb{R}^{n\times n}$, $\mathbb{X}_{2i} \in \mathbb{R}^{n\times n}$, $\mathbb{Y}_i \in \mathbb{R}^{n\times p}$, satisfying

$$\begin{bmatrix} P_{11i} & * \\ P_{21i} & P_{22i} \end{bmatrix} > 0, \tag{23}$$

$$\begin{bmatrix} -P_{11i} & * & * & * \\ -P_{21i} & -P_{22i} & * & * \\ \Upsilon_1 & \mathbb{X}_{1i}\Pi_i^{12} & \Upsilon_2 & * \\ 0 & \Upsilon_3 & P_{21j} & \Upsilon_4 \end{bmatrix} < 0, \tag{24}$$

where

$\Upsilon_1 = \mathbb{X}_{1i}(F_i - G_iL_i)^{\eta(i_m)}$, $\Upsilon_2 = P_{11j} - \mathbb{X}_{1i} - \mathbb{X}_{1i}{}^T$,

$\Upsilon_3 = \mathbb{X}_{2i}F_i{}^{\eta(i_m)} + \mathbb{Y}_iCF_i{}^{\eta(i_m)-1}, \Upsilon_4 = P_{22j} - \mathbb{X}_{2i} - \mathbb{X}_{2i}{}^T$,

then the NCS (10) under the intelligent scheduling controller is asymptotically stable, where K_i is given by $K_i = \mathbb{X}_{2i}^{-1}\mathbb{Y}_i$.

Proof: Let $\mathbb{X}_i \triangleq diag\{\mathbb{X}_{1i}, \mathbb{X}_{2i}\}$ and

$$P_i \triangleq \begin{bmatrix} P_{11i} & * \\ P_{21i} & P_{22i} \end{bmatrix}. \tag{25}$$

It is obvious that (24) can be written as

$$\begin{bmatrix} -P_i & * \\ \Gamma_i & P_j - \mathbb{X}_i - \mathbb{X}_i^T \end{bmatrix} < 0, \tag{26}$$

where

$$\Gamma_i = \begin{bmatrix} \Upsilon_1 & \mathbb{X}_{1i}\Pi_i^{12} \\ 0 & \Upsilon_3 \end{bmatrix}. \tag{27}$$

Letting Υ_1, Υ_{13}, Π_i^{12}, and $\mathbb{Y}_i = \mathbb{X}_{2i}K_i$ into (27) yields

$$\Gamma_i = \mathbb{X}_iM_i, \tag{28}$$

where M_i is of form (9).

From the viewpoint of $\Gamma_i = \mathbb{X}_iM_i$, (26) can be rewritten as

$$\begin{bmatrix} -P_i & * \\ \mathbb{X}_iM_i & P_j - \mathbb{X}_i - \mathbb{X}_i^T \end{bmatrix} < 0. \tag{29}$$

Pre- and Post-multiplying (29) by $[I \;\; M_i{}^T]$ and its transpose respectively leads to

$$M_i{}^TP_jM_i - P_i < 0. \tag{30}$$

According to Theorem 1, (30) implies that the closed-loop NCS (10) is asymptotically stable. This completes the proof. □

Based on the conditions of Corollary 1, which is obtained via the common quadratic Lyapunov function, a simpler observer gain matrix design method can be obtained as follows.

Corollary 2: For a given NCS (10) and a set of given parameters L_i and h_i, if for $i \in \mathbb{S}$, there exist symmetric matrices $P_{11} \in \mathbb{R}^{n\times n}$, $P_{22} \in \mathbb{R}^{n\times n}$ and matrices $P_{21} \in \mathbb{R}^{n\times n}$, $\mathbb{X}_{1i} \in \mathbb{R}^{n\times n}$, $\mathbb{X}_{2i} \in \mathbb{R}^{n\times n}$, $\mathbb{Y}_i \in \mathbb{R}^{n\times p}$, satisfying

$$\begin{bmatrix} P_{11} & * \\ P_{21} & P_{22} \end{bmatrix} > 0, \tag{31}$$

$$\begin{bmatrix} -P_{11} & * & * & * \\ -P_{21} & -P_{22} & * & * \\ \bar{\Upsilon}_1 & \mathbb{X}_{1i}\bar{\Pi}_i & \bar{\Upsilon}_2 & * \\ 0 & \bar{\Upsilon}_3 & P_{21} & \bar{\Upsilon}_4 \end{bmatrix} < 0, \tag{32}$$

where

$\bar{\Upsilon}_1 = \mathbb{X}_{1i}(F_i - G_iL_i)^{\eta(i_m)}$,

$\bar{\Upsilon}_2 = P_{11} - \mathbb{X}_{1i} - \mathbb{X}_{1i}{}^T$,

$\bar{\Upsilon}_3 = \mathbb{X}_{2i}F_i{}^{\eta(i_m)} + \mathbb{Y}_iCF_i{}^{\eta(i_m)-1}$,

$\bar{\Upsilon}_4 = P_{22} - \mathbb{X}_{2i} - \mathbb{X}_{2i}{}^T$, then the NCS (10) under the intelligent scheduling controller is asymptotically stable, where K_i is given by $K_i = \mathbb{X}_{2i}^{-1}\mathbb{Y}_i$.

Proof: The proof is analogous to that of Theorem 3, and so it is omitted. □

In summary, the proposed controller design approach can be formulated as the following three-step procedure.

Step 1: According to (18) and (19), compute h_A^i and h_B^i for each network QoS level Q_i, where for $i \in \mathbb{S}$. Then select h_i within $(h_A^i \ h_B^i)$.

Step 2: Design L_i by guaranteeing that $F_i - G_i L_i$ is schur stable.

Step 3: Design K_i by solving the LMIs in Theorem 3 or Corollary 2.

5. THE NCS PERFORMANCE OPTIMIZATION

Since the control parameters and the sampling period are closely related to the control performance of NCSs, the NCSs performance can be improved by using optimum sampling period and control parameters. Unfortunately, a closed-form relationship among the network QoS, the sampling period, the control parameters and the control performance is not available. To obtain the optimal h_i, L_i and K_i for NCSs, we will transform the optimal parameter design problem into an optimization problem and then solve the optimization problem.

- The optimization variables

In the optimization problem, the optimization variables are the elements of h_i, L_i and K_i:

$$V = [h_1 \, K_1 \, L_1 \cdots h_N \, K_N \, L_N], \tag{33}$$

where $i \in \mathbb{S}$, N is the number of different QoS levels.

Note that L_i and K_i may be matrices. For future use, we transform the optimization variables into a vector:

$$V = (v_1, \ v_2, \ldots, \ v_{\bar{S}}), \tag{34}$$

where $\bar{S}$ is the dimension of V.

- The fitness function

In this paper, we define the fitness function as

$$J = \sum_{k=0}^{N_J} |r(k) - y(k)|, \tag{35}$$

where N_J is the appropriate time index such that the tracking has arrived at the steady state, $r(k)$ and $y(k)$ are the measurements on reference input and plant output at time index k.

- Sampling Period Constraint (SPC)

To ensure the physical feasibility of the sampling period and the system performance, we introduce the sampling period constraint subjected by the optimization problem as follows:

$$\text{SPC}: \ h_B^i < h_i < h_A^i, \tag{36}$$

where $i \in \mathbb{S}$, h_B^i and h_A^i are the lower and upper bounds of h_i, and they are designed according to (18) and (19).

- Stable Domain Constraint (SDC)

A set $\Psi \in \mathbb{R}^{1\times\bar{S}}$ is called stable domain (SD) of NCSs (10), if any $V \in \Psi$ satisfies the stability conditions of Theorem 1 or Corollary 1. Obviously, any V that can provide satisfactory control performance for NCSs must be within the defined *Stable domain*, since satisfactory control performance not only requires a NCS to be asymptotically stable, but also require the NCS to meet some performance specifications. Therefore, it is sufficient to search the optimal parameters within the *Stable domain*, which can effectively reduce the search space and guarantee the stability of NCSs during the optimization procedure. Based on this idea, *Stable Domain Constraint* (SDC) is introduced as follows:

$$\text{SDC}: \ V \in \Psi, \quad \forall V, \tag{37}$$

which means that the optimization variables are constrained to the defined *Stable Domain* throughout the optimization procedure.

In summary, the optimization problem can be expressed as

$$\begin{aligned} &\textbf{OP}: \min \mathbf{J} \\ &\text{s.t.} \ \ \text{SPC}: \ h_B^i < h_i < h_A^i \\ &\qquad \text{SDC}: \ V \in \Psi, \ \forall V \end{aligned} \tag{38}$$

(38) asks for V that minimizes the value of the cost function within the defined *Stable Domain*. Obviously, the SDC can guarantee the stability of NCSs during the optimization procedure. On the other hand, by minimizing the cost function, the control performance of NCSs is optimized. Therefore, both stability and control performance can be guaranteed by solving the optimization problem (38).

In this paper, a population-based, heuristic search algorithm namely Estimation of Distribution Algorithm (EDA) is used to solve the optimization problem (38). EDA has recently been recognized as a new computing paradigm in evolutionary computation. Unlike other Evolutionary Algorithms (EAs), EDA does not use crossover or mutation. Instead, it explicitly extracts global statistical information from the promising solutions, and subsequently builds a probability distribution model of promising solutions based on the extracted information. New population are generated by sampling from the probability distribution model rather than rely on random search operators (such as crossover and mutation). Thus, one of the most appealing advantages of EDAs over classical EAs is the reduction in the number of parameters to be tuned or assessed by the user. For more details on EDA, please refer to [18-20] and the references therein. In this paper, a typical EDA algorithm is used to solve the optimization problem (38), where the optimization variables in the search space are modeled as a multivariate normal distribution $p(V)=\Pi_{i=1}^{\bar{S}} p(v_i)$, which is a product of $\bar{S}$ independent univariate normal distributions $p(\bar{v}_i)=(\mu_i,\sigma_i)$. The EDA algorithm used in this paper can be summarized as follows:

1. Generate n_1 individuals meeting Stable Domain constraint in the initial search space randomly to form an initial population, where the positive integer n_1 denotes the population size of EDA, the generated n_1 individuals are denoted as $V^j=(v_1^j,v_2^j,\ldots,v_{\bar{S}}^j)$, $j=1,\ldots,n_1$.

2. Repeat the following steps until the termination criterion is met.

a. Select the best n_2 ($n_2<n_1$) individuals from the parent generation, where the positive integer n_2 denotes the population size of promising individuals, the selected n_2 gain candidates are denoted as $\hat{V}^j=(\hat{v}_1^j,\hat{v}_2^j,\ldots,\hat{v}_{\bar{S}}^j)$, $j=1,\ldots,n_2$.

b. Update the probability distribution model $p(V)$ using the selected n_2 promising individuals according to

$$\mu_i=\frac{1}{n_2}\sum_{j=1}^{n_2}\hat{v}_i^j, \tag{39}$$

$$\sigma_i=\sqrt{\frac{1}{n_2-1}\sum_{j=1}^{n_2}(\hat{v}_i^j-\mu_i)^2}. \tag{40}$$

c. Generate the next population: The best individual in the current population is copied to the next population; generate n_1-1 individuals (we use n_1-1 rather than n_1 because the best individual in the current population has already been selected as an individual) meeting Stable Domain constraint based on the updated probability distribution model $p(V)=\Pi_{i=1}^{\bar{S}}(\mu_i,\sigma_i)$.

Although we do not know the exact *Stable Domain* of a NCS, we can realize the *Stable Domain Constraint* based on the stability conditions described in (13) or (16). Taking Step 1 for example, we can generate an individual candidate in the search space randomly, and then check whether the individual candidate satisfies the stability conditions described in (13) or (16). If it does, the individual candidate will be selected as an individual of the population. Otherwise, we discard the individual candidate point and repeat the above steps until we obtain one satisfying the stability conditions described in (13) or (16). Similar arrangements are for Step 2-c, except that the gain candidate is generated according to the probability distribution model $p(V)$ at Step 2-c.

6. ILLUSTRATIVE EXAMPLES

In order to illustrate the effectiveness of the proposed approaches, let us consider a networked servo motor control system with setup shown in Fig.1. The parameters of the motor used in this paper are listed in Table 1.

Let $x_p=[\theta,\omega]^T$, where θ and ω are the output angle and the angular speed respectively, then the DC motor dynamics can be expressed as:

$$\dot{x}_p(t)=\begin{bmatrix}0 & 1\\ 1 & -217.4\end{bmatrix}x_p(t)+\begin{bmatrix}0\\ 1669.5\end{bmatrix}u(t). \tag{41}$$

In the presence of network, the network-induced delay corresponding to medium and high network loads ranges in [2ms 22ms] and [25ms 43ms] randomly, with packet dropout rates of 2% and 4%, respectively. In the following text, we provide a comparative study of the proposed method with two published methods.

Case 1: The state feedback controller

Let us consider the networked controller in [15], which is a memoryless state feedback controller of form $u=Kx$. Noting that this networked controller can only be applied to NCSs with network-induced delay less than a sampling period, the sampling period of networked DC motor control system is set to 0.05s. Then we solve the LMIs of Theorem 2 in [15] and

Table 1. The parameters of the networked DC motor.

J	Inertia	$11\times10^{-7}\text{kg}\cdot\text{m}^2$
L	Inductance	$0.24\times10^{-3}\text{H}$
R	Resistance	2.32Ω
K	Torque Constant	$23.4\times10^{-3}\text{N}\cdot\text{m/A}$
n	Gear reduction ratio	1/318
K_e	Back-EMF Constant	$23.4\times10^{-3}\text{Vs/rad}$

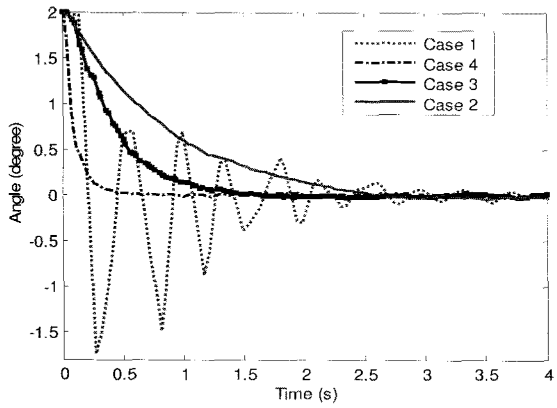


Fig. 4. Typical system performances using different networked controllers.

obtain $K = [-1.8774\ -0.0086]$. With the initial state $x_0 = [2\ -2]^T$, the simulation result under the networked controller $u = [-1.8774\ -0.0086]x$ is depicted by dotted line in Fig. 4.

Case 2: The observer-based controller with fixed sampling period and fixed control parameters

Let us consider the networked controller in [2], which is an observer-based controller with fixed sampling period and fixed control parameters. We apply the controller design method in [2] to the networked DC motor control system, and therefore obtain $L = [0.1577\ 0.0006]$ and $K = [-0.9865\ -0.0000]^T$. With the same initial state, the simulation result of networked motor control system under in this case is depicted by solid line in Fig. 4.

Case 3: The intelligent scheduling controller

Let $Q_1 \triangleq \{T_{\text{delay}} \in [0.02, 0.22]\}$ and $Q_2 \triangleq \{T_{\text{delay}} \in [0.25, 0.43]\}$ denote the network QoS conditions corresponding to the medium and high network loads. Then we apply the proposed three-step controller design procedure to the networked DC motor control system, and therefore obtain $h_1 = 0.021$, $L_1 = [0.486\ 0.0123]$, $K_1 = [-1.1068\ -0.0010]^T$ for Q_1 and $h_2 = 0.039$, $L_2 = [0.1397\ 0.0005]$, $K_2 = [-0.9235\ 0.0004]^T$ for Q_2. With the same initial state, the simulation result of system under the proposed intelligent scheduling controller is depicted by dotted-solid line in Fig. 4.

Case 4: The intelligent scheduling controller with optimized sampling period and control parameters

The simulation used a population size of 60 for the EDA, with the best 20 individuals selected from the parent generation to update the multivariate Gaussian distribution for the next generation. By using EDA to solve the optimization problem (38), we obtain $h_1 = 0.02$, $L_1 = [1.3827\ 0.009]$, $K_1 = [-1.1089\ -0.0180]^T$ for Q_1, and $h_2 = 0.04$, $L_2 = [0.5827\ 0.008]$, $K_2 = [-1.0813\ -0.0100]^T$ for Q_2. With the same initial state, the simulation result of system under the intelligent scheduling controller with optimized sampling period and control parameters is depicted by dotted-dashed line in Fig. 4.

The simulation results demonstrate that for the concerned networked DC motor control system, the system using the memoryless state feedback controller has more overshoot and is more oscillatory. This indicates that the memoryless state feedback controller is not capable of controlling the networked system very well. This phenomena is reasonable since it uses less information about the system than other considered networked controllers. When an observer-based networked controller with fixed sam-pling period and fixed control parameters is used, the system has no overshoot, but still has long settling time. However, using the intelligent scheduling controller, especially using the intelligent scheduling controller with optimized sampling period and control parameters, the system not only has no over-shoot, but also has significantly shorter settling time.

Moreover, analysis of the sampling period of the intelligent scheduling controller indicates an interesting nature. For high network loads, the controller sampling period is relative long. However, for low network loads, the intelligent scheduling controller increases the sampling rate to obtain better performance. This mechanism enables the NCSs to achieve higher resource utilization rate and avoid overloads.

7. CONCLUSIONS

This paper presents an intelligent scheduling control method for NCSs under effects of network-induced delay and packet dropout, where the control parameters and sampling period are simultaneously adjusted to improve the NCSs performance. For the concerned NCSs, a discrete-time switch model is proposed, which enables us to apply the theory from switch systems to study NCSs in discrete-time domain. In the proposed framework, we obtained two types of stability conditions in terms of LMIs, which can be easily checked by using standard numerical software. Based on the obtained stability conditions, the corresponding controller design problem is solved and the performance optimization problem is also investigated. Simulation results are given to demonstrate the effectiveness of the proposed approaches.

REFERENCES

[1] S. Chai, G. P. Liu, D. Rees, and Y. Xia, "Design and practical implementation of internet-based predictive control of a servo system," *IEEE Trans. on Control Systems Technology*, vol. 16,

no. 1, pp. 158-168, 2008.

[2] H. Li, Z. Sun, H. Liu, and M.-Y. Chow, "Predictive observer-based control for networked control systems with network-induced delay and packet dropout," *Asian Journal of Control*, vol. 10, no. 6, pp. 1-3, 2008.

[3] Y. Tipsuwan and M.-Y. Chow, "Gain scheduler middleware: A methodology to enable existing controllers for networked control and teleoperation-part II: teleoperation," *IEEE Trans. on Industrial Electronics*, vol. 51, no. 6, pp. 1228-1237, 2004.

[4] P. Seiler and R. Sengupta, "An H_∞ approach to networked control," *IEEE Trans. on Automatic Control*, vol. 50, no. 3, pp. 356-364, 2005.

[5] M. Gaid, A. Çela, and Y. Hamam, "Optimal integrated control and scheduling of networked control systems with communication constraints: Application to a car suspension system," *IEEE Trans. on Control Systems Technology*, vol. 14, no. 4, pp. 776-787, 2006.

[6] K. Ji and W.-J. Kim, "Stochastic optimal control and network co-design for networked control systems," *International Journal of Control, Automation and Systems*, vol. 5, no. 5, pp. 515-525, 2007.

[7] K. Ji and W.-J. Kim, "Robust control for networked control systems with admissible parameter uncertainties," *International Journal of Control, Automation and Systems*, vol. 5, no. 4, pp. 372-378, 2007.

[8] L. A. Montestruque and P. J. Antsaklis, "On the model-based control of networked systems," *Automatica*, vol. 39, no. 10, pp. 1837-1843, 2003.

[9] P. V. Zhivoglyadov and R. H. Middleton, "Networked control design for linear systems," *Automatica*, vol. 39, no. 4, pp. 743-750, 2003.

[10] M.-Y. Chow and Y. Tipsuwan, "Gain adaptation of networked DC motor controllers based on QoS variations," *IEEE Trans. on Industrial Electronics*, vol. 50, no. 5, pp. 936-943, 2003.

[11] Y. Tipsuwan and M.-Y. Chow, "On the gain scheduling for networked PI controller over IP network," *IEEE/ASME Trans. on Mechatronics*, vol. 9, no. 3, pp. 491-498, 2004.

[12] K. Li and J. Baillieul, "Robust quantization for digital finite communication bandwidth (DFCB) control," *IEEE Trans. on Automatic Control*, vol. 49, no. 9, pp. 1573-1584, 2004.

[13] L. A. Montestruque and P. J. Antsaklis, "Static and dynamic quantization in model-based networked control systems," *International Journal of Control*, vol. 80, no. 1, pp. 87-101, 2007.

[14] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *Control Systems Magazine*, vol. 21, no. 1, pp. 84-99, 2001.

[15] M. Yu, L. Wang, T. G. Chu, and G. M. Xie, "Stabilization of networked control systems with data packet dropout and network delays via switching system approach," *Proc. of the 43th IEEE Conference on Decision and Control*, pp. 3539-3544, 2004.

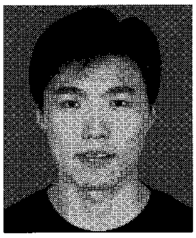
[16] F. L. Lian, *Analysis, Design, Modeling, and Control of Networked Control Systems*, Ph.D. Dissertation, University of Michigan, 2001.

[17] G. F. Franklin, M. L. Workmann, and J. D. Powell, *Digital Control of Dynamic Systems*, 3rd edition, Addison-Wseley, 1998.

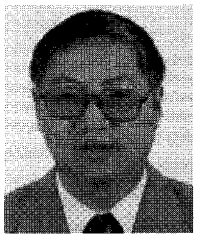
[18] C. González, *Contributions on Theoretical Aspects of Estimation of Distribution Algorithms*, Ph.D. Dissertation, University of the Basque Country, Spain, 2005.

[19] Y. Cai, X. Sun, and P. Jia, "Probabilistic modeling for continuous EDA with Boltzmann selection and Kullback-Leibeler divergence," *Proc. of the 8th Annual Conf. on Genetic and Evolutionary Computation*, pp. 389-396, 2006.

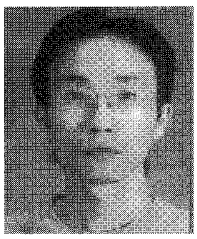
[20] P. A Bosman and J. Grahl, "Matching inductive search bias and problem structure in continuous Estimation-of-Distribution algorithms," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1246-1264, 2008.



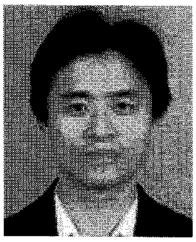
HongBo Li received the B.S. degree in 2004 from Northeastern University, Shenyang, China. He is now pursuing a Ph.D. degree in the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His current research interests include networked control systems and intelligent control.



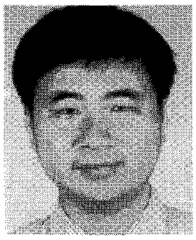
ZengQi Sun received the Ph.D. degree in Control Engineering in 1981 from the Chalmas University of Technology in Sweden. He is currently a Professor in the Department of Computer Science and Technology, Tsinghua University, China. His current research interests include intelligent control, robotics, fuzzy systems, neural networks and evolution computing etc.



Badong Chen received the B.S. and M.S. degrees in Control Theory and Engineering from Chongqing University, Chongqing, China, in 1997 and 2003. He is currently pursuing a Ph.D. degree in Computer Science and Technology from Tsinghua University, Beijing, China. His current research interests are in signal processing and adaptive control.



Huaping Liu received the Ph.D. degree in 2004 from the Department of Computer Science and Technology, Tsinghua University, Beijing, China. Currently he is an Associate Professor in Department of Computer Science and Technology, Tsinghua University. His research interests include intelligent control and robotics.



Fuchun Sun received the Ph.D. degree in 1998 from the Department of Computer Science and Technology, Tsinghua University, Beijing, China. Currently, he is a Professor in the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include intelligent control, neural networks, fuzzy systems, variable structure control, nonlinear systems and robotics.