

Optimal Stabilizing Gain Selection for Networked Control Systems With Time Delays and Packet Losses

Hongbo Li, Mo-Yuen Chow, *Fellow, IEEE*, and Zengqi Sun, *Senior Member, IEEE*

Abstract—This brief addresses the optimal-stabilization problem for networked control systems (NCSs) with time delays and packet losses. The closed-loop NCS is modeled as a discrete-time switched system, and the stability conditions are derived in terms of linear matrix inequality. A controller design method with both system stability and control performance taken into account is proposed, and estimation of distribution algorithm is used to select the optimal stabilizing gain. The proposed method can be easily implemented to various applications, since it has simple structure and has no assumptions on time-delay and packet-loss models. Simulation and experimental results are given to demonstrate the effectiveness of the proposed approach.

Index Terms—Estimation of distribution algorithm (EDA), networked control systems (NCSs), packet loss, time delay.

I. INTRODUCTION

NETWORKED control systems (NCSs) of which data networks are used for the connections between spatially distributed system components have recently attracted much attention from research communities. Issues such as time delays [1]–[5], packet losses [6], [7], network constraints [8], [9], signal quantization [10], [11], and scheduling [13] have been investigated, with results reported in the literature. In addition, due to the advantages of reduced system wiring, simple installation, increased system flexibility, and resources sharing, NCSs have been finding applications in dc motors [3], robots [4], vehicles [14], car suspension system [15], ball maglev system [16], industrial fish-processing machine [17], etc.

Time delay in NCSs is a major cause for system-performance deterioration and potential system instability. In the literature, time delays have been modeled by using various formulations such as constant delay [6], independently random delays [1], [2], and random delays governed by Markov chains [5]. Another important issue in NCSs is packet loss. There are two

typical approaches to describe packet losses in the literature. The first approach assumes that packet losses follow certain probability distributions such as Markov chains. The second one is the deterministic approach, which specifies the packet losses in a time-averaged sense or in terms of bounds on maximum-allowed consecutive packet losses. Note that the packets in NCSs usually suffer time delays and packet losses simultaneously during network transmission. Therefore, the analysis and synthesis of NCSs with time delays and packet losses is a challenge but practical problem. In the literature, some important methodologies, such as queuing methodology, stochastic optimal control, and predictive control, are proposed to address packet-loss and/or time-delay issues. However, these methods require large controller memory to store a large amount of past information, and the dynamic model of the plant has to be very precise [18]. Moreover, as pointed out in [1] and [18], for stochastic optimal control of NCSs, the probability distributions of time delays must be known *a priori*.

Memoryless state-feedback control is a simple and effective control method. It has a simple structure and does not require controller memory to store past information. Moreover, it has no assumptions on time-delay and packet-loss models. Therefore, memoryless state-feedback control for NCSs with time delays and packet losses, either in the continuous-time domain [18]–[20] or in the discrete-time domain [7], [21]–[25], has received considerable attention in the past few years. It was noticed that, for the controller design methods in the aforementioned results, most of them are derived directly from the stability conditions, and none of them considered the control performance during the controller design stage. As a result, the designed controller can only guarantee NCSs to be asymptotically stable while the control performance of NCSs may not be satisfied. However, control practices not only require an NCS to be asymptotically stable but also require the NCS to meet some performance specifications. Therefore, the memoryless state-feedback controller design method with both system stability and control performance taken into account is much needed, which motivates this brief.

In this brief, we investigate the optimal-stabilization problem for NCSs with time delays and packet losses. In this brief, the closed-loop NCS is modeled as a discrete-time switched system, and the stability conditions are derived in terms of linear matrix inequality. Based on the obtained stability conditions, estimation of distribution algorithm (EDA) is adopted to select the optimal gain for the corresponding stabilizing controller. In comparison with the existing stabilizing controllers, the proposed method takes both system stability and control performance into account during the controller design stage. Moreover, the proposed method can be easily implemented to various applications, since it is simple and has no assumptions

Manuscript received March 24, 2008; revised June 13, 2008. Manuscript received in final form August 03, 2008. First published April 14, 2009; current version published August 26, 2009. Recommended by Associate Editor C. N. Hadjicostis. This work was supported in part by the National Science Foundation of China under Grant 60574035 and Grant 60674053, by the National Key Project for Basic Research of China under Grant 2002cb312205, and by the National Science Foundation under Grant IIS-0426852.

H. Li is with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27606 USA, and also with the Department of Computer Science and Technology, State Key Laboratory of Intelligent Technology and Systems, Tsinghua University, Beijing 100084, China (e-mail: hli7@ncsu.edu; hb-li04@mails.tsinghua.edu.cn).

M.-Y. Chow is with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27606 USA (e-mail: chow@ncsu.edu).

Z. Sun is with Department of Computer Science and Technology, State Key Laboratory of Intelligent Technology and Systems, Tsinghua University, Beijing 100084, China (e-mail: szq-dcs@mail.tsinghua.edu.cn).

Color versions of one or more of the figures in this brief are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCST.2008.2004346

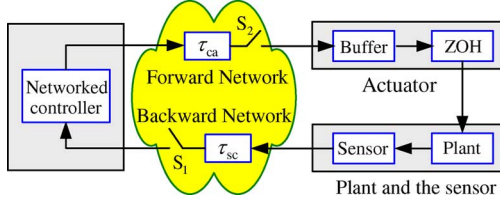


Fig. 1. Structure of the concerned NCSs.

on time-delay and packet-loss models. Simulation and experimental results are given to demonstrate the effectiveness of the proposed approaches.

Notation: Throughout this brief, \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 matrix with appropriate dimensions, and the notation \mathbb{Z}^+ stands for the set of nonnegative integers.

II. PROBLEM FORMULATION

As shown in Fig. 1, the considered NCS can be grouped into four modules: 1) the plant and the sensor; 2) the data network; 3) the networked controller; and 4) the actuator. Each module is described in the following sections.

A. Controlled Plant and Sensor

The dynamics of the remote-controlled plant is given by the following linear discrete-time model:

$$\mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{G}\mathbf{u}(k) \quad (1)$$

where $\mathbf{x}(k) \in \mathbb{R}^n$ is the plant state and $\mathbf{u}(k) \in \mathbb{R}^m$ is the control input. \mathbf{F} and \mathbf{G} are known matrices with appropriate dimensions. Note that (1) can be considered as discretized from a continuous-time system given by

$$\dot{\mathbf{x}}_p(t) = \mathbf{A}\mathbf{x}_p(t) + \mathbf{B}\mathbf{u}(t) \quad (2)$$

with sampling period h and

$$\mathbf{F} = e^{\mathbf{A}h} \quad \mathbf{G} = \int_0^h e^{\mathbf{A}\tau} d\tau \mathbf{B}. \quad (3)$$

The sensor is time-driven. At each sampling period, the sampled plant state and its timestamp (i.e., the time the plant state is sampled) are encapsulated into a packet and sent to the controller via the network.

B. Data Network

In practice, the data packets in NCSs usually suffer time delays and packet losses during network transmissions. As shown in Fig. 1, sensor-to-controller delay and controller-to-actuator delay are denoted by τ_{sc} and τ_{ca} , respectively, and two switches S_1 and S_2 are used to model the packet losses in the backward and the forward networks. For example, when S_1 is open,

the sensor packet is lost during network transmission from the sensor to the controller, whereas when S_1 is closed, the packet is successfully transmitted to the controller with a sensor-to-controller delay τ_{sc} . Similar arrangements are for the forward network.

C. Networked Controller

The networked controller is a memoryless state-feedback controller with the following form:

$$\mathbf{u} = \mathbf{K}\bar{\mathbf{x}} \quad (4)$$

where $\mathbf{K} \in \mathbb{R}^{m \times n}$ is the feedback gain to be designed and $\bar{\mathbf{x}}$ is the plant state from the arriving sensor packet. The controller is event-driven. Whenever there is a sensor packet that reaches the controller, the controller starts calculating the control signal. Immediately after the calculation, the new control signal and the timestamp of the used plant state are encapsulated into a packet and sent to the actuator via the network.

D. Actuator

The actuator is time-driven. It is assumed that the actuator and the sensor have the same sampling period h , and they are synchronized. Note that the actuator and the sensor are at the same location, and theretofore, the synchronization between them can be easily achieved by hardware synchronization, for instance, using special wiring to distribute a global clock signal to the sensor and the actuator. The actuator has a buffer size of one, which means that the latest control packet is used to control the plant. Let $\tau_k \triangleq \tau_{sc}(k) + \tau_{ca}(k)$ express the round-trip-time (RTT) delay encountered by the k th sampling time packet from the sensor, i.e., τ_k is from the time instant when the sensor samples the plant state to the time instant when the control signal based on this packet reaches the actuator. To simplify the analysis, we only use the control packets with $\tau_k \leq 2h$ to control the plant (i.e., when a control packet reaches the actuator, it is used to update the buffer if its RTT delay is no longer than $2h$ and its timestamp is newer than that of the control signal in buffer; otherwise, it will be discarded), but the idea behind this brief can be easily extended to the case $\tau_k > 2h$. The actuator uses a zeroth-order hold (ZOH). At each sampling period, the ZOH reads the control signal from buffer and uses it to control the plant.

III. DISCRETE-TIME SWITCHED MODEL FOR NCSs

This section will present a discrete-time switched model for NCSs. To address this problem, we introduce the definition of an *effective sensor packet*, and it will be used throughout this brief. A packet from the sensor is called an *effective sensor packet* for controller (4), if its RTT delay is no longer than $2h$. Let $S \triangleq \{i_1, i_2, \dots\}$, a subsequence of $\{0, 1, 2, \dots\}$, denotes the sequence of time index of *effective sensor packets*. Since only *effective sensor packets* are used to control the plant, the sensor packets between two *effective sensor packets* can all be considered as dropped packets. To avoid confusion, we used the following illustration to explain and demonstrate the used nota-

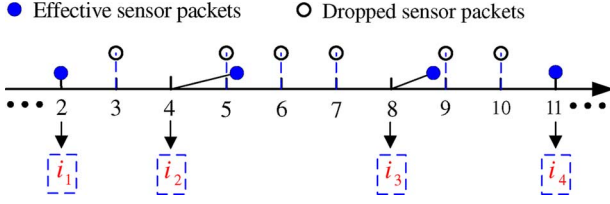


Fig. 2. Illustration of effective packets in NCSs.

tions. Let us assume that $S \triangleq \{i_1, i_2, i_3, i_4, \dots\}$ in an NCS is $\{2, 4, 8, 11, \dots\}$ (see Fig. 2). It means that the 2nd, 4th, 8th, and 11th sensor packets are *effective sensor packets*, while the i th sensor packets ($i = 3, 5, 6, 7, 9, 10$) are not used to control the plant, and they are considered as dropped packets.

Based on the earlier notations, the packet-loss process in the NCSs can be defined as

$$\left\{ \eta(i_m) \triangleq i_{m+1} - i_m, \quad i_m \in S \right\} \quad (5)$$

which means that, from i_m to i_{m+1} , the number of dropped packets is $\eta(i_m) - 1$. For the sake of analysis, we define $\mathbf{N}_{\text{drop}} \triangleq \max_{i_m \in S} \{\eta(i_m)\}$. Then, we can conclude that $\eta(i_m)$ takes values in a finite set $\Omega \triangleq \{1, 2, \dots, \mathbf{N}_{\text{drop}}\}$.

Let τ_{i_m} express the RTT delay encountered by the m th *effective sensor packet*. As shown in Fig. 3, for NCSs during the time interval between two effective packets, i.e., $[i_m h, i_{m+1} h]$, four cases may arise and are discussed as follows.

Case 1) $\tau_{i_m} = 0$, i.e., the RTT delay encountered by the m th *effective sensor packet* is zero. Since the sensor packets between two *effective sensor packets* are lost, the control input for $l \in [i_m, i_{m+1})$ will be $\mathbf{u}(l) = \mathbf{K}\mathbf{x}(i_m)$ because of ZOH. Therefore, the dynamics of the NCS can be described as follows:

$$\begin{aligned} \mathbf{x}(i_m + 1) &= \mathbf{F}\mathbf{x}(i_m) + \mathbf{G}\mathbf{K}\mathbf{x}(i_m) \\ \mathbf{x}(i_m + 2) &= \mathbf{F}^2\mathbf{x}(i_m) + \mathbf{F}\mathbf{G}\mathbf{K}\mathbf{x}(i_m) + \mathbf{G}\mathbf{K}\mathbf{x}(i_m) \\ &\vdots \end{aligned} \quad (6)$$

Correspondingly, for the time instant $i_{m+1}h$, we have

$$\mathbf{x}(i_{m+1}) = \mathbf{F}^{\eta(i_m)}\mathbf{x}(i_m) + \sum_{j=0}^{\eta(i_m)-1} \mathbf{F}^j \mathbf{G}\mathbf{K}\mathbf{x}(i_m). \quad (7)$$

Case 2) $\tau_{i_m} \in (0, h]$, i.e., the RTT delay encountered by the m th *effective sensor packet* ranges within $(0, h]$. In this case, the last control signal $\mathbf{K}\mathbf{x}(i_{m-1})$ and the new control signal $\mathbf{K}\mathbf{x}(i_m)$ are used to control the plant during $[i_m h, (i_m + 1)h)$ and $[(i_m + 1)h, i_{m+1}h)$, respectively. Therefore, the dynamics of the NCS can be described as follows:

$$\begin{aligned} \mathbf{x}(i_m + 1) &= \mathbf{F}\mathbf{x}(i_m) + \mathbf{G}\mathbf{K}\mathbf{x}(i_{m-1}) \\ \mathbf{x}(i_m + 2) &= \mathbf{F}^2\mathbf{x}(i_m) + \mathbf{F}\mathbf{G}\mathbf{K}\mathbf{x}(i_{m-1}) + \mathbf{G}\mathbf{K}\mathbf{x}(i_m) \\ &\vdots \end{aligned} \quad (8)$$

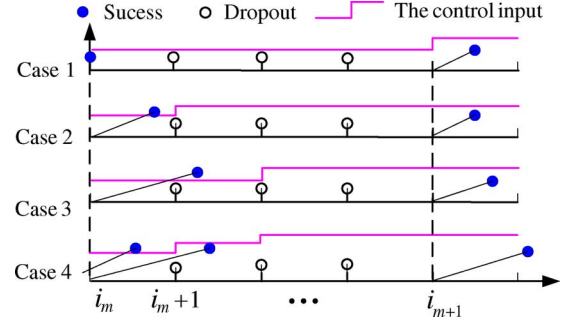


Fig. 3. Illustration of time delays and packet losses in NCS.

Correspondingly, for the time instant $i_{m+1}h$, we have

$$\begin{aligned} \mathbf{x}(i_{m+1}) &= \mathbf{F}^{\eta(i_m)}\mathbf{x}(i_m) + \sum_{j=0}^{\eta(i_m)-2} \mathbf{F}^j \mathbf{G}\mathbf{K}\mathbf{x}(i_m) \\ &\quad + \mathbf{F}^{\eta(i_m)-1} \mathbf{G}\mathbf{K}\mathbf{x}(i_{m-1}) \end{aligned} \quad (9)$$

where the notation $\sum_{j=a}^b \mathbf{F}^j$ satisfies the following property:

$$\begin{cases} \sum_{j=a}^b \mathbf{F}^j = 0, & b < a \\ \mathbf{F}^j = 0, & j < 0. \end{cases} \quad (10)$$

Note that property (10) is used to ensure that (9) is a general expression. For example, when $\eta(i_m)$ is one, (9) can be reduced to the first equation in (8) with property (10).

Case 3) $\tau_{i_m} \in (h, 2h]$ and $\tau_{i_{m-1}} < \eta(i_{m-1})h$, i.e., the RTT delay encountered by the m th *effective sensor packet* ranges within $(h, 2h]$, and the control signal based on the $(m-1)$ th *effective sensor packet* arrives at the actuator before the time instant $i_m h$. In this case, the new and the last control signals, i.e., $\mathbf{K}\mathbf{x}(i_m)$ and $\mathbf{K}\mathbf{x}(i_{m-1})$, are used to control the plant during $[(i_m + 2)h, i_{m+1}h)$ and $[i_m h, (i_m + 2)h)$, respectively. Based on the similar analysis in cases 1) and 2), the dynamics of the NCS in this situation can be described as follows:

$$\begin{aligned} \mathbf{x}(i_{m+1}) &= \mathbf{F}^{\eta(i_m)}\mathbf{x}(i_m) + \sum_{j=0}^{\eta(i_m)-3} \mathbf{F}^j \mathbf{G}\mathbf{K}\mathbf{x}(i_m) \\ &\quad + \sum_{j=\eta(i_m)-2}^{\eta(i_m)-1} \mathbf{F}^j \mathbf{G}\mathbf{K}\mathbf{x}(i_{m-1}). \end{aligned} \quad (11)$$

Case 4) $\tau_{i_{m-1}} \in (\eta(i_{m-1})h, (\eta(i_{m-1}) + 1)h]$ and $\tau_{i_m} \in (h, 2h]$, i.e., the RTT delay encountered by the m th *effective sensor packet* ranges within $(h, 2h]$, and the control signal based on the $(m-1)$ th *effective sensor packet* arrives at the actuator during the time interval $(i_m h, (i_m + 1)h]$. In this case, the new control signal $\mathbf{K}\mathbf{x}(i_m)$ is also used to control the plant during $[(i_m + 2)h, i_{m+1}h)$. However, the last two

control signals, i.e., $\mathbf{K}\mathbf{x}(i_{m-2})$ and $\mathbf{K}\mathbf{x}(i_{m-1})$, are used to control the plant during $[i_m h, (i_m + 1)h)$ and $[(i_m + 1)h, (i_m + 2)h)$, respectively. The dynamics of the NCS in this situation is described as

$$\begin{aligned} \mathbf{x}(i_{m+1}) = & \mathbf{F}^{\eta(i_m)} \mathbf{x}(i_m) + \sum_{j=0}^{\eta(i_m)-3} \mathbf{F}^j \mathbf{G} \mathbf{K} \mathbf{x}(i_m) \\ & + \mathbf{F}^{\eta(i_m)-2} \mathbf{G} \mathbf{K} \mathbf{x}(i_{m-1}) + \mathbf{F}^{\eta(i_m)-1} \mathbf{G} \mathbf{K} \mathbf{x}(i_{m-2}). \end{aligned} \quad (12)$$

From the earlier analysis, we can be seen that, during $[i_m h, i_{m+1} h]$, there are four different cases that may arise, and the corresponding dynamics of NCSs are described as four different subsystems (7), (9), (11), and (12). Now, taking all the subsystems into consideration, we introduce the augmented state $\mathbf{z}(i_m) = [\mathbf{x}_{i_m}^T, \mathbf{x}_{i_{m-1}}^T, \mathbf{x}_{i_{m-2}}^T]^T$ into (7), (9), (11), and (12). Then, the four subsystems can be lumped into a general framework, which is described by the following discrete-time switched system:

$$\mathbf{z}(i_{m+1}) = \mathbf{M}_{r(i_m)} \mathbf{z}(i_m) \quad (13)$$

where $r(i_m)$ is a piecewise constant function called a switch signal, which takes values in a finite set $\mathbb{S} \triangleq \{1, 2, 3, 4\}$. Noting that the switch state $r = 1, r = 2, r = 3$, and $r = 4$ are corresponding to (7), (9), (11), and (12), respectively, $\mathbf{M}_{r(i_m)}$ is in the form of (15), shown at the bottom of the page. Moreover, for $i_m < l < i_{m+1}$, the dynamics of NCSs can be expressed by

$$\mathbf{z}(l) = \bar{\mathbf{M}}_{l,r(i_m)} \mathbf{z}(i_m) \quad (14)$$

where $\mathbf{z}(l) \triangleq [\mathbf{x}_l^T, \mathbf{x}_{l-1}^T, \mathbf{x}_{l-2}^T]^T$, $\bar{\mathbf{M}}_{l,r(i_m)}$ is similar to (15) but with $\eta(i_m)$ replaced by $l - i_m$.

Note that, in the NCS model (13), both time delays and packet losses are taken into account, and there is no assumption on the time-delay model or packet-loss model (i.e., the time delay can be constant, independently random, or random but governed by a Markov chain, and the packet losses can be independently random or random but governed by a Markov chain). Therefore, the NCS model (13) and the following proposed methods are quite general and can be applied to various applications. In the following theorem, the stability conditions are derived for NCSs via a state-dependent Lyapunov approach. This theorem will play an instrumental role in the controller design stage.

Theorem 1: NCS (13) in the presence of time delays and packet losses is asymptotically stable, if for $i, j \in \mathbb{S}$, there exist positive-definite matrices \mathbf{P}_i and \mathbf{P}_j satisfying

$$\mathbf{M}_i^T \mathbf{P}_j \mathbf{M}_i - \mathbf{P}_i < 0 \quad (16)$$

where \mathbf{M}_i is of form (15) with a given gain matrix \mathbf{K} .

Proof: For NCS (13), let a state-dependent Lyapunov function candidate be

$$V(i_m) = \mathbf{z}^T(i_m) \mathbf{P}_r \mathbf{z}(i_m) \quad (17)$$

where \mathbf{P}_r are matrices upon the switch state r in the NCSs model (13). Let the switch signal r at time instants $i_m h$ and $i_{m+1} h$ be i and j , where $i, j \in \mathbb{S}$. The difference of Lyapunov function is given by

$$\begin{aligned} \Delta V(k) = & V(k+1) - V(k) \\ = & \mathbf{z}^T(i_{m+1}) \mathbf{P}_j \mathbf{z}(i_{m+1}) - \mathbf{z}^T(i_m) \mathbf{P}_i \mathbf{z}(i_m). \end{aligned} \quad (18)$$

Then, along the solution of the closed-loop system (13), we have

$$\Delta V(k) = \mathbf{z}^T(i_m) \left(\mathbf{M}_i^T \mathbf{P}_j \mathbf{M}_i - \mathbf{P}_i \right) \mathbf{z}(i_m) < 0 \quad (19)$$

for any $\mathbf{z}(i_m) \neq 0$. Moreover, for $i_m < l < i_{m+1}$, we have $V(l) = \mathbf{z}^T(i_m) \bar{\mathbf{M}}_{l,i}^T \mathbf{P}_j \bar{\mathbf{M}}_{l,i} \mathbf{z}(i_m)$. From the modeling process, we know that $\eta(i_m)$ in \mathbf{M}_i takes values in Ω , while $l - i_m$ in $\bar{\mathbf{M}}_{l,i}$ takes values in $\Omega - \{N_{\text{drop}}\}$. Thus, if the condition (16) holds, we have

$$V(l) - V(i_m) = \mathbf{z}^T(l) \left(\bar{\mathbf{M}}_{l,i}^T \mathbf{P}_j \bar{\mathbf{M}}_{l,i} - \mathbf{P}_i \right) \mathbf{z}(l) < 0 \quad (20)$$

for any $\mathbf{z}(l) \neq 0$, where $i_m < l < i_{m+1}$.

From (19), we have $\lim_{i_m \rightarrow \infty} V(i_m) = 0$. From (20), we get $\lim_{l \rightarrow \infty} V(l) = 0$ for $l \neq i_m$. Summarizing the discussed two cases, we can conclude that $\lim_{l \rightarrow \infty} V(l) = 0$ for $l \in \mathbb{Z}^+$, which implies that system (13) is asymptotically stable. ■

IV. OPTIMAL STABILIZING CONTROLLER DESIGN

This section is concerned with the optimal stabilizing controller design problem. We will transform the controller design problem into an optimization problem and then solve the optimization problem.

$$\begin{aligned} \mathbf{M}_1 = & \begin{bmatrix} \mathbf{F}^{\eta(i_m)} + \sum_{j=0}^{\eta(i_m)-1} \mathbf{F}^j \mathbf{G} \mathbf{K} & 0 & 0 \\ \mathbf{I} & 0 & 0 \\ 0 & \mathbf{I} & 0 \end{bmatrix} & \mathbf{M}_4 = & \begin{bmatrix} \mathbf{F}^{\eta(i_m)} + \sum_{j=0}^{\eta(i_m)-3} \mathbf{F}^j \mathbf{G} \mathbf{K} & \mathbf{F}^{\eta(i_m)-2} \mathbf{G} \mathbf{K} & \mathbf{F}^{\eta(i_m)-1} \mathbf{G} \mathbf{K} \\ \mathbf{I} & 0 & 0 \\ 0 & \mathbf{I} & 0 \end{bmatrix} \\ \mathbf{M}_2 = & \begin{bmatrix} \mathbf{F}^{\eta(i_m)} + \sum_{j=0}^{\eta(i_m)-2} \mathbf{F}^j \mathbf{G} \mathbf{K} & \mathbf{F}^{\eta(i_m)-1} \mathbf{G} \mathbf{K} & 0 \\ \mathbf{I} & 0 & 0 \\ 0 & \mathbf{I} & 0 \end{bmatrix} & \mathbf{M}_3 = & \begin{bmatrix} \mathbf{F}^{\eta(i_m)} + \sum_{j=0}^{\eta(i_m)-3} \mathbf{F}^j \mathbf{G} \mathbf{K} & \sum_{j=\eta(i_m)-2}^{\eta(i_m)-1} \mathbf{F}^j \mathbf{G} \mathbf{K} & 0 \\ \mathbf{I} & 0 & 0 \\ 0 & \mathbf{I} & 0 \end{bmatrix} \end{aligned} \quad (15)$$

A. Optimization Variables

In the optimization problem, the optimization variables are the elements of the controller gain matrix $\mathbf{K} \in \mathbb{R}^{m \times n}$. For future use, we transform the optimization variables into a vector

$$\bar{\mathbf{K}} = (\bar{K}_1, \bar{K}_2, \dots, \bar{K}_{\bar{S}}) \quad (21)$$

where $\bar{S} = mn$ is the dimension of $\bar{\mathbf{K}}$.

B. Cost Function

Different engineering applications have their performance measures, which can be used to construct their cost functions. In this brief, the *mean square error*, the *overshoot*, and the *settling time* are used to evaluate the NCSs' performance. The corresponding cost function is defined as follows:

$$\mathbf{J} = \lambda_1 \mathbf{J}_1 + \lambda_2 \mathbf{J}_2 + \lambda_3 \mathbf{J}_3 \quad (22a)$$

$$\mathbf{J}_1 = \begin{cases} 0, & \text{MSE} \leq \text{MSE}_n \\ (\text{MSE} - \text{MSE}_n) / \text{MSE}_n, & \text{MSE} > \text{MSE}_n \end{cases} \quad (22b)$$

$$\mathbf{J}_2 = \begin{cases} 0, & |\text{O.S.}| \leq |\text{O.S.}_n| \\ (\text{O.S.} - \text{O.S.}_n) / \text{O.S.}_n, & |\text{O.S.}| > |\text{O.S.}_n| \end{cases} \quad (22c)$$

$$\mathbf{J}_3 = \begin{cases} 0, & \text{S.T.} \leq \text{S.T.}_n \\ (\text{S.T.} - \text{S.T.}_n) / \text{S.T.}_n, & \text{S.T.} > \text{S.T.}_n \end{cases} \quad (22d)$$

where

$$\text{MSE} = \frac{1}{N} \sum_{k=0}^N (\mathbf{r}(k) - \mathbf{y}(k))^2 \quad (23)$$

is the *mean square error*, MSE_n is the nominal *mean square error*, *O.S.* is the *overshoot*, O.S._n is the nominal *overshoot*, *S.T.* is the *settling time*, and S.T._n is the nominal *settling time*. MSE_n , O.S._n , and S.T._n are nominal performance measures that the system should achieve without network in the loop. If the NCS performance meets the nominal performance specifications, we have $\mathbf{J} = 0$ because $\mathbf{J}_i = 0$, $i = 1, 2, 3$. In (23), N is the appropriate time index such that the tracking has arrived at the steady state; $\mathbf{r}(k)$ and $\mathbf{y}(k)$ are the reference input and plant output at time index k . The cost \mathbf{J}_1 penalizes poor response time and convergence, the cost \mathbf{J}_2 penalizes high value of the overshoot, and the cost \mathbf{J}_3 penalizes long settling time. The problem-dependent weights λ_1 , λ_2 , and λ_3 are used to specify the relative significance of \mathbf{J}_1 , \mathbf{J}_2 , and \mathbf{J}_3 , respectively, on the overall system performance.

C. SRC

A vector set $\Psi \subset \mathbb{R}^{1 \times \bar{S}}$ is called the *stable region* of NCSs, if any vector $\bar{\mathbf{K}} \in \Psi$ can satisfy the stability conditions described in (16), i.e., Ψ is composed of $\bar{\mathbf{K}}$ that can guarantee the NCSs to be asymptotically stable. Obviously, the controller gains that can provide satisfactory control performance for NCSs must be within the *stable region* (see Fig. 4), since satisfactory performance not only requires an NCS to be asymptotically stable but also require the NCS to meet some performance specifications. Therefore, it is sufficient to search the optimal controller gain within the *stable region*, which can effectively reduce the

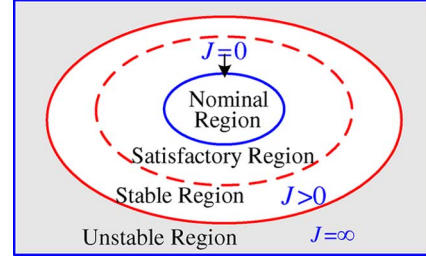


Fig. 4. Graphical representation of the regions used in this brief.

search space and guarantee the stability of NCSs from a theoretical point of view. Based on this idea, *stable region constraint* (SRC) is introduced as follows:

$$\text{SRC} : \bar{\mathbf{K}} \in \Psi \quad \forall \bar{\mathbf{K}} \quad (24)$$

which means that the optimization variables are constrained to the *stable region* throughout the optimization procedure.

In summary, the optimization problem can be expressed as

$$\begin{aligned} \text{OP} : \quad & \min \quad \mathbf{J}(\text{NCS}, \bar{\mathbf{K}}) \\ \text{s.t. SRC} : \quad & \bar{\mathbf{K}} \in \Psi \quad \forall \bar{\mathbf{K}}. \end{aligned} \quad (25)$$

Equation (25) asks for $\bar{\mathbf{K}}$ that minimizes the value of the cost function within the defined *stable region*. Obviously, the SRC 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, solving the optimization problem (25) offers a substantial advantage in that both stability and control performance can be guaranteed in the controller design stage.

For the optimization problem (25), heuristic-search algorithms such as genetic algorithm can be adopted to solve it. However, the behavior of these heuristic-search algorithms depends on many parameters. For researcher without associated experiences, the choice of suitable values for so many parameters is difficult. Unlike other evolutionary algorithms (EAs), EDA does not use crossover or mutation. Instead, EDA uses probability distribution model derived from the promising solutions to generate new population. Thus, one of the most appealing advantages of EDA over classical EAs is the reduction in the number of parameters to be tuned by the user. Moreover, it is noted that the optimization problem (25) is subjected to SRC. Therefore, to accelerate the search speed, a generic characteristic expected for the heuristic-search algorithm is that it can search the optimal solution within the defined *stable region* with high probability. As mentioned before, EDA adopts probability distribution model derived from the promising solutions to generate new population, which enables EDA to possess the required characteristic. Motivated by the earlier analyses, we adopt EDA to solve the optimization problem (25) in this brief. The optimization variables in the search space are modeled as a multivariate normal distribution $p(\bar{\mathbf{K}}) = \prod_{i=1}^{\bar{S}} p(\bar{K}_i) = \prod_{i=1}^{\bar{S}} (\mu_i, \sigma_i)$, which is a product of \bar{S} -independent univariate normal distributions $p(\bar{K}_i) = (\mu_i, \sigma_i)$. The EDA algorithm used in this brief can be summarized as follows.

TABLE I
STATISTICAL MEASURES (MINIMUM, MEDIAN, MEAN AND MAXIMUM DELAY, AND MAXIMUM CONSECUTIVE PACKET LOSS)
OF THE RTT DELAYS AND PACKET LOSSES FROM THE ADAC LABORATORY AT NCSU TO DIFFERENT DESTINATIONS

Destinations host	$\tau_{min}(sec)$	$\tau_{median}(sec)$	$\tau_{mean}(sec)$	$\tau_{max}(sec)$	N_{max}
www.utexas.edu	0.037	0.053	0.0383	0.069	2
www.ku.ac.th	0.210	0.5815	0.2802	0.953	2
www.tsinghua.edu.cn	0.261	0.9775	0.489	1.694	3

- Rule 1) **Initialization:** Generate n_1 gain candidates meeting SRC randomly to form an initial population.
- Rule 2) Repeat the following steps until the termination criterion is met.
- Selection:** Select the best n_2 ($n_2 < n_1$) gain candidates from the parent generation.
 - Updating:** Update $p(\bar{K})$ using the selected n_2 promising gain candidates.
 - Sampling:** Generate $n_1 - 1$ gain candidates meeting SRC based on the updated $p(\bar{K})$; copy the best gain candidate in the current population to the next population.

For more details on EDA, please refer to [26], [27], and the reference therein. It is noticed that, although we do not know the exact *stable region* of an NCS, we can realize the SRC based on the stability conditions described in (16). Taking step 1) for example, we can generate a gain candidate in the search space randomly and, then, check whether the gain candidate satisfies the stability conditions described in (16). If it does, the gain candidate will be selected as an individual of the population. Otherwise, we discard it and repeat the earlier steps until we obtain one satisfying the stability conditions (16). Similar arrangements are for step 2)–c).

V. ILLUSTRATIVE EXAMPLES

This section provides a comprehensive study of proposed method. As a practical application, we apply the proposed method to a networked dc-motor system. Let $\mathbf{x}_p = [\theta, \omega]^T$, where θ and ω are the motor angular position and the motor angular speed, respectively. The dc motor used in this brief can be expressed as

$$\dot{\mathbf{x}}_p(t) = \begin{bmatrix} 0 & 1 \\ 1 & -217.4 \end{bmatrix} \mathbf{x}_p(t) + \begin{bmatrix} 0 \\ 1669.5 \end{bmatrix} \mathbf{u}(t). \quad (26)$$

The networked dc-motor system is designed to drive the dc motor to a preset angle of 40° . The performance specifications are given as follows.

- 1) *Mean square error:* $MSE \leq 220^{\circ 2}$.
- 2) *Overshoot:* $O.S. \leq 0.1^\circ$.
- 3) *Settling time:* $S.T. \leq 8$ s.

Typical RTT delay and packet-loss information of packets between the Advanced Diagnosis And Control (ADAC) Laboratory at North Carolina State University (NCSU), Raleigh, U.S., and the destinations listed in Table I are measured for a 24-h period (0000–2400). The statistical information on the experimental measures are listed in Table I, and the RTT delay histograms of 60 000 packets are shown in Fig. 5. Those experimental measured data are used in numerical simulations and experimental verifications as follows: Each value of these RTT delays is divided by two and is utilized as τ_{sc} and τ_{ca} ; the packet loss is simulated either by the switch S_1 or by the switch S_2 .

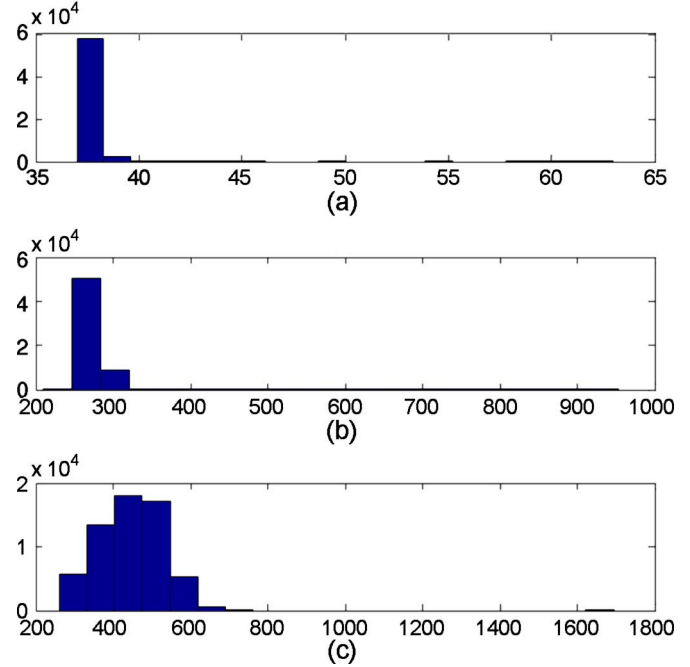


Fig. 5. Histograms of RTT delays measured from ADAC Laboratory at NCSU to (a) www.utexas.edu, (b) www.ku.ac.th, and (c) www.tsinghua.edu.cn, where the x -axis is RTT delay (in milliseconds) and the y -axis is the number of packets.

Numerical Simulations:

Case 1) NCS with network from the ADAC Laboratory at NCSU to www.utexas.edu.

We first consider the networked dc-motor system with network from the ADAC Laboratory at NCSU to www.utexas.edu. Table I shows that, in the used network, the RTT time delays range from 0.037 to 0.069 s, and the maximum consecutive packet losses is two. For the simulation study, we set the sampling period to 0.05 s. Noting that the networked system is designed to drive the dc motor to a preset angle, we introduce the reference input \mathbf{r} into networked controller (4) and modify it into $\mathbf{u} = \mathbf{K}\bar{\mathbf{x}} - \mathbf{K}(1, 1)\mathbf{r}$. For the proposed controller design method, we set n_1 and n_2 in EDA to 50 and 20, and set $MSE_n, O.S._n, S.T._n, \lambda_1, \lambda_2$, and λ_3 in the cost function to 20, 0.03, 1, 1, 1, and 1, respectively. By applying the proposed controller design method to the networked system, we obtain the optimal gain $\mathbf{K} = [-0.4216, 0.0102]$. With the initial state $\mathbf{x}_0 = [3.65, 0]^T$, a typical step response of networked dc motor is shown in Fig. 6(a), where the corresponding network condition is shown in Fig. 6(d), which is selected from the experimental measures. The MSE, O.S., and S.T. corresponding to Fig. 6(a) are $28.27^{\circ 2}$, 0° , and 1.15 s. Apparently, the

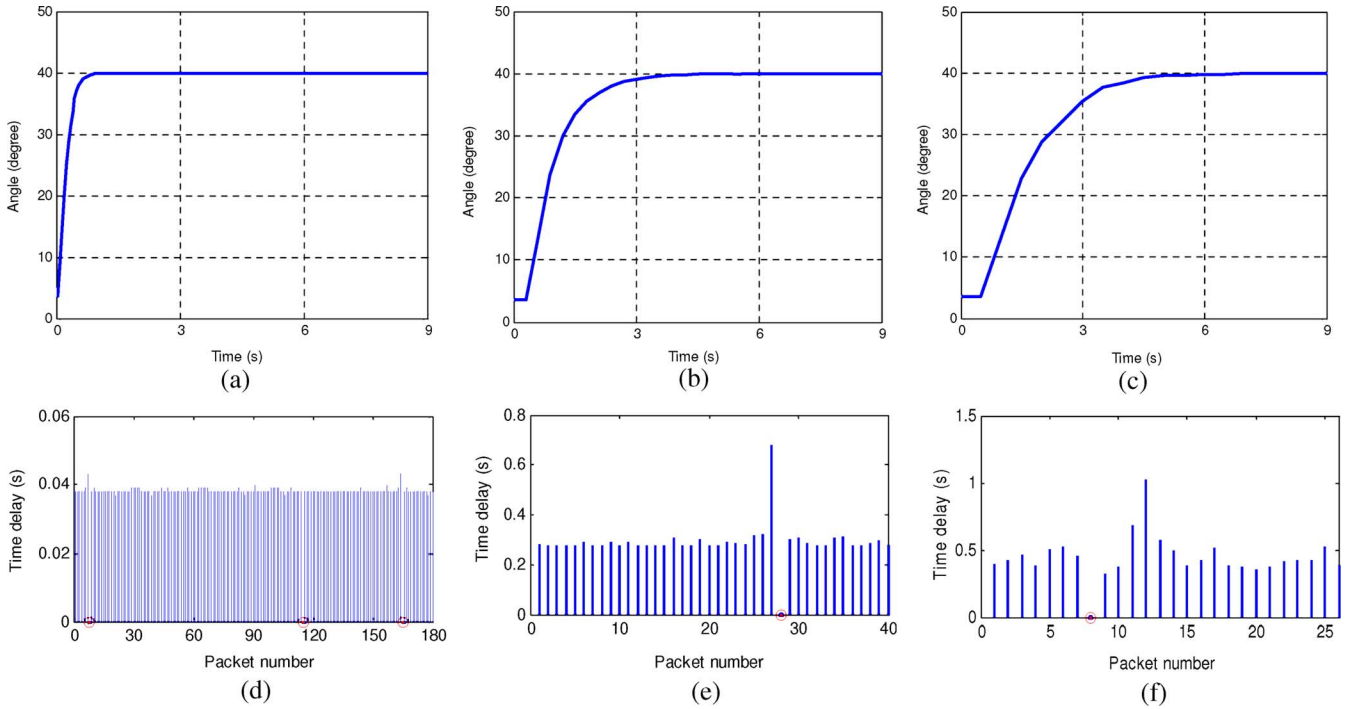


Fig. 6. Typical simulation results of the networked dc-motor control system with different network conditions.

system performance meets the given specifications very well, which demonstrates that the proposed controller provides a satisfactory performance.

Case 2) NCS with network from the ADAC Laboratory at NCSU to www.ku.ac.th.

We then consider the networked dc-motor system with network from the ADAC Laboratory at NCSU to www.ku.ac.th. To include as much data packets within $2h$ as possible, we set the sampling period to 0.3 s. By considering the packets with $\tau_k > 2h$ as lost packets, the maximum consecutive packet losses becomes three, which is obtained based on the statistics of experimental measures of RTT delays and packet losses. We apply the proposed controller design method to the earlier networked system and obtain $\mathbf{K} = [-0.1202, -0.0139]$. A typical step response of networked dc motor is shown in Fig. 6(b), where the corresponding network condition is shown in Fig. 6(e). The MSE, O.S., and S.T. corresponding to Fig. 6(b) are $122.23^{\circ 2}$, 0° , and 3.85 s. Apparently, the system performance meets the given specifications very well, which demonstrates that the proposed controller still provides a satisfactory performance even though worse network condition is considered.

Case 3) NCS with network from the ADAC Laboratory at NCSU to www.tsinghua.edu.cn

In fact, the proposed method can still stabilize the networked system with even more worse network condition. To illustrate this point, let us consider the networked dc-motor system with network from the ADAC Laboratory at NCSU to www.tsinghua.edu.cn. To include as much data

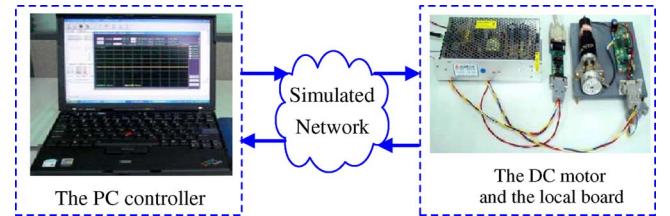


Fig. 7. Actual networked dc-motor system setup.

packets within $2h$ as possible, we set the sampling period to 0.5 s. By considering the packets with $\tau_k > 2h$ as lost packets, the maximum consecutive packet losses becomes four, which is obtained based on the statistics of experimental network measures. We apply the proposed controller design method to the earlier networked system and obtain $\mathbf{K} = [-0.0691, -0.0169]$. A typical step response of networked dc motor is shown in Fig. 6(c), where the corresponding network condition is shown in Fig. 6(f). The MSE, O.S., and S.T. corresponding to Fig. 6(c) are $203.96^{\circ 2}$, 0° , and 6.0 s. Apparently, the system performance still meets the given specifications, which demonstrates that the proposed controller can still provide a satisfactory performance even though more worse network condition is considered.

Experimental Verifications: To further illustrate the effectiveness and applicability of the proposed method, a networked dc-motor platform is constructed. The actual system setup is shown in Fig. 7, where the experimental apparatus consists of a PC controller, a local board, and a dc motor with sensors. The PC controller is used to implement the networked controller.

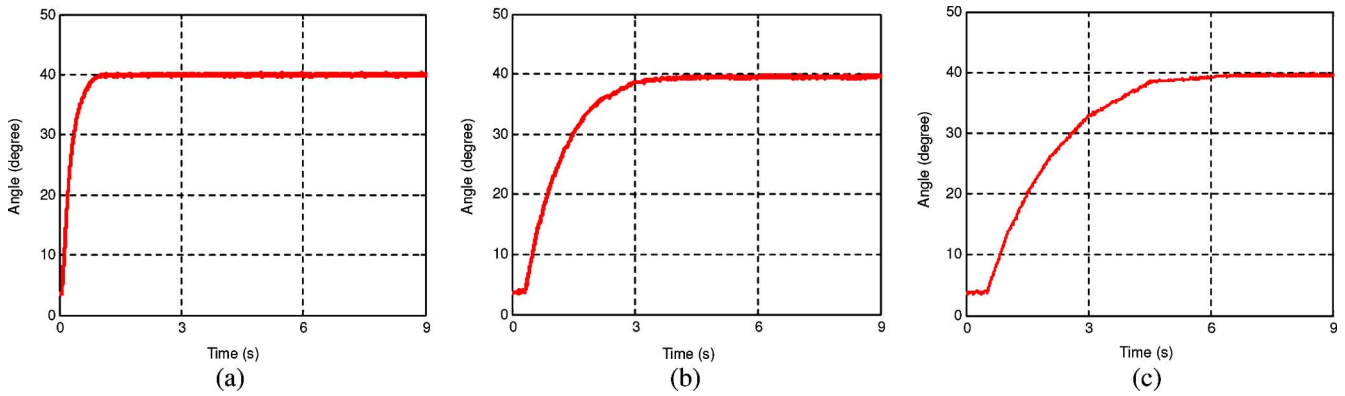


Fig. 8. Typical experimental results of the networked dc-motor control system with different network conditions.

The local board is on the plant side and used for two functions. The first function is to convert the control signal read from the buffer into a pulsewidth-modulation (PWM) signal and, then, send the PWM signal to drive the dc motor. The second function is to encapsulate the plant state and its timestamp into a packet and send it to the PC controller via the network. Both the local board and the sensor are time-driven, and they are synchronized by using the same clock signal.

In the experimental verifications, the PC controller and the dc motor were linked by a simulated network. Using real-time software and a hardware timer, the simulated network creates time delays by delaying the packets transmitted between the PC controller and the dc motor and creates packet losses by discarding the transmitted packets. The reasons for using the simulated network rather than using the actual network are as follows: 1) The experimental results can be compared with the simulation results under the same network condition and 2) the experiment is ensured to be repeatable for various investigations.

Case 4) NCS with network from the ADAC Laboratory at NCSU to www.utexas.edu.

Consider the network condition shown in Fig. 6(d). With the same initial state and controller gain as used in case 1), the experimental result of the networked dc motor is shown in Fig. 8(a). Apparently, the experimental result is consistent with the simulation result in case 1).

Case 5) NCS with network from the ADAC Laboratory at NCSU to www.ku.ac.th.

Consider the network condition shown in Fig. 6(e). With the same initial state and controller gain as used in case 2), the experimental result of the networked dc motor is shown in Fig. 8(b). Apparently, the experimental result is consistent with the simulation result in case 2).

Case 6) NCS with network from the ADAC Laboratory at NCSU to www.tsinghua.edu.cn.

Finally, consider the network condition shown in Fig. 6(f). With the same initial state and controller gain as used in case 3), the experimental result of the networked dc motor is shown in Fig. 8(c). Apparently, the experimental result is consistent with the simulation result in case 3).

The simulation and experimental results demonstrate the effectiveness of the proposed method for real-world application. Figs. 6 and 8 show that the control performance of networked system degrades when network conditions become worse. This phenomenon is reasonable, since NCS performance worsens with longer time delays and more packet losses. Note that the sampling period of the networked system increases from cases 1) to 3) and from cases 4) to 6). This demonstrates that larger sampling period is allowable in the NCSs design stage.

VI. CONCLUSION

This brief presents a controller design method for a class of NCSs with time delays and packet losses. The most appealing advantage of the proposed method is that both system stability and control performance are considered during the controller design stage. The proposed method can be easily implemented to various applications, since it has simple structure and has no assumptions on time-delay and packet-loss models. Simulation and experimental results are given to demonstrate the effectiveness of the proposed approach.

ACKNOWLEDGMENT

The authors would like to thank Associate Editor Prof. C. Hadjicostis and the anonymous reviewers for their constructive comments and suggestions, which have lead to important enhancements of the original manuscript.

REFERENCES

- [1] J. Nilsson, B. Bernhardsson, and B. Wittenmark, "Stochastic analysis and control of real-time systems with random time delays," *Automatica*, vol. 34, no. 1, pp. 57–64, Jan. 1998.
- [2] S. Hu and Q. Zhu, "Stochastic optimal control and analysis of stability of networked control systems with long delay," *Automatica*, vol. 39, no. 11, pp. 1877–1884, Nov. 2003.
- [3] 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. Control Syst. Technol.*, vol. 16, no. 1, pp. 158–168, Jan. 2008.
- [4] 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. Ind. Electron.*, vol. 51, no. 6, pp. 1228–1237, Dec. 2004.
- [5] L. Zhang, Y. Shi, T. Chen, and B. Huang, "A new method for stabilization of networked control systems with random delays," *IEEE Trans. Autom. Control*, vol. 50, no. 8, pp. 1177–1181, Aug. 2005.
- [6] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *IEEE Control Syst. Mag.*, vol. 21, no. 1, pp. 84–99, Feb. 2001.

- [7] J. Xiong and J. Lam, "Stabilization of linear systems over networks with bounded packet loss," *Automatica*, vol. 43, no. 1, pp. 80–87, Jan. 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, Oct. 2003.
- [9] P. V. Zhivoglyadov and R. H. Middleton, "Networked control design for linear systems," *Automatica*, vol. 39, no. 4, pp. 743–750, Apr. 2003.
- [10] K. Li and J. Baillieul, "Robust quantization for digital finite communication bandwidth (DFCB) control," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1573–1584, Sep. 2004.
- [11] L. A. Montestruque and P. J. Antsaklis, "Static and dynamic quantization in model-based networked control systems," *Int. J. Control*, vol. 80, no. 1, pp. 87–101, 2007.
- [12] G. C. Walsh and H. Ye, "Scheduling of networked control systems," *IEEE Control Syst. Mag.*, vol. 21, no. 1, pp. 57–65, Feb. 2001.
- [13] H. S. Park, Y. H. Kim, D. S. Kim, and W. H. Kwon, "A scheduling method for network-based control systems," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 3, pp. 318–330, May 2002.
- [14] P. Seiler and R. Sengupta, "An H_∞ approach to networked control," *IEEE Trans. Autom. Control*, vol. 50, no. 3, pp. 356–364, Mar. 2005.
- [15] 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. Control Syst. Technol.*, vol. 14, no. 4, pp. 776–787, Jul. 2006.
- [16] W. J. Kim, K. Ji, and A. Srivastava, "Network-based control with real-time prediction of delayed/lost sensor data," *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 1, pp. 182–185, Jan. 2006.
- [17] P. L. Tang and C. de Silva, "Compensation for transmission delays in an ethernet-based control network using variable-horizon predictive control," *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 4, pp. 707–718, Jul. 2006.
- [18] D. Yue, Q.-L. Han, and C. Peng, "State feedback controller design of networked control systems," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 51, no. 11, pp. 640–644, Nov. 2004.
- [19] M. Yu, L. Wang, T. Chu, and F. Hao, "An LMI approach to networked control systems with data packet dropout and transmission delays," in *Proc. 43rd IEEE Conf. Decision Control*, 2004, pp. 3545–3550.
- [20] D. Yue, Q.-L. Han, and J. Lam, "Network-based robust H_∞ control of systems with uncertainty," *Automatica*, vol. 41, no. 6, pp. 999–1007, Jun. 2005.
- [21] 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," in *Proc. 43rd IEEE Conf. Decision Control*, 2004, pp. 3539–3544.
- [22] M. Garcia-Rivera and A. Barreiro, "Analysis of networked control systems with drops and variable delays," *Automatica*, vol. 43, no. 12, pp. 2054–2059, Dec. 2007.
- [23] 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 J. Control*, vol. 10, no. 6, pp. 638–650, Nov. 2008, to be published.
- [24] H. Li, Z. Sun, M.-Y. Chow, and B. Chen, "A simple state feedback controller design method of networked control systems with time delay and packet dropout," in *Proc. IFAC World Congr.*, 2008, pp. 12575–12580.
- [25] H. Li, Z. Sun, M.-Y. Chow, and B. Chen, "State feedback controller design of networked control systems with time delay and packet dropout," in *Proc. IFAC World Congr.*, 2008, pp. 6626–6631.
- [26] C. González, "Contributions on theoretical aspects of estimation of distribution algorithms," Ph.D. dissertation, Univ. Basque Country, Bilbao, Spain, 2005.
- [27] P. A. Bosman and J. Grahl, "Matching inductive search bias and problem structure in continuous estimation-of-distribution algorithms," *Eur. J. Oper. Res.*, vol. 185, no. 3, pp. 1246–1264, Mar. 2008.