Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

State feedback controller design of networked control systems with time delay and packet dropout ⋆

Hongbo Li * **Zengqi Sun** * **Mo-Yuen Chow** ** **Badong Chen** *

* *Department of Computer Science and Technology, State Key Laboratory of Intelligent Technology and Systems, Tsinghua University, Beijing, China, 100084. (E-mail: Hb-li04@mails.tsinghua.edu.cn).*
** *Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27695-7911, USA. (E-mail: chow@ncsu.edu)*

Abstract: In this paper, the stability analysis and synthesis problems for networked control systems (NCSs) are investigated. By introducing the lifting technique into NCSs, a novel discrete-time switch model is proposed with the consideration of time delay and packet dropout during the transmission of packets. It describes NCSs as a switch system, and therefore enables us to apply the theory from switch systems to study NCSs in discrete-time domain. In terms of the given model, we give sufficient conditions for the existence of state feedback controller such that the closed-loop NCSs are asymptotically stable. Based on the obtained stability conditions, a homotopy-based iterative LMI algorithm is developed to get the state feedback gain. Simulation results are given to demonstrate the effectiveness of the proposed approaches.

1. INTRODUCTION

Networked control systems (NCSs) of which communication networks are used for the connections between spatially distributed system components have recently attracted much attention from research communities. Certain issues such as network-induced delay (Nilsson et al. [1998], Hu et al. [2003], Tipsuwan et al. [2004], Liu et al. [2006], Zhang et al. [2001]), packet dropout (Zhang et al. [2001], Xiong et al. [2007]), network constraints (Montestruque et al. [2003], Peter et al. [2003]), signal quantization (Li et al. [2004], Montestruque et al. [2007]), and scheduling (Walsh et al. [2001]), were investigated and some useful results were reported. In addition, due to the advantages of low cost, simple installation and high reliability (Xiong et al. [2007], Yue et al. [2004]), NCSs have been finding applications in DC motors (Liu et al. [2006]), vehicles (Seiler et al. [2005]) and robots (Tipsuwan et al. [2003]), etc.

In practice, controller design for NCSs with both network-induced delay and packet dropout is a very interesting and practical problem since the packets in NCSs usually suffer network-induced delay and packet dropout simultaneously during the network transmissions. Recently, some important results are developed in this field. Based on the LMI approach, state feedback control method was investigated by Yue et al. [2004] and Yu et al. [2004] respectively. In the case of H_∞ control, Yue et al. [2005] studied the controller design for NCS with external disturbance and parameter uncertainties. Note that those results investigated NCSs in continuous-time domain. In discrete-time domain, Yu et al. [2004] and Lin et al. [2005] studied the stabilization problem for NCSs. However, in those results, the controller and actuator are combined together. That means network exists only between the sensor and the controller. Xiong et al. [2007] studied the state feedback control for NCSs under a general framework. However, it assumes network-induced delay is constant, and therefore can not deal with the case when network-induced delay is random. Therefore, despite the progress made in controller design for NCSs with both network-induced delay and packet dropout, it has become evident that the state feedback control for NCSs in discrete-time domain, especially for NCSs under a general framework and with random network-induced delay, are still required.

In this paper, we focus on solving the state feedback controller design problem of NCSs in discrete-time domain and under a general framework, where random network-induced delay and arbitrary packet dropout are taken into account simultaneously. By using the lifting technique (see Li et al. [2002], Park et al. [2004]), a mathematic model is proposed for the considered NCSs. It describes NCSs as a switch system, and therefore enables us to apply the theory from switch systems to study NCSs in discrete-time domain. In terms of the given model, we give sufficient conditions for the existence of state feedback controller such that the closed-loop NCSs are asymptotically stable. Based on the obtained stability conditions, we further investigated the corresponding state feedback controller

⋆ The research work of HongBo Li and ZengQi Sun was jointly supported by the National Science Foundation (NSF) of China (Grant No:60574035, 60674053) and the National Key Project for Basic Research of China (Grant No: 2002cb312205). The research work of Mo-Yuen Chow was partially sponsored by the NSF through Grant No.IIS-0426852 (Intelligent Human-Machine Interface and Control for Highly Automated Chemical Screening Processes).

design problem. The numerical examples are provided to demonstrate the effectiveness of the proposed approaches.

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 FORMULATION

The setup of NCSs considered in this paper is depicted in Fig.1, where networks exist between sensor and controller, and between controller and actuator. The controlled plant is given by

$$\boldsymbol{x}(k+1) = \boldsymbol{F}\boldsymbol{x}(k) + \boldsymbol{G}\boldsymbol{u}(k) \tag{1}$$

where $k \in \mathbb{Z}^+$ is the time index, $\boldsymbol{x}(k) \in \mathbb{R}^n$ and $\boldsymbol{u}(k) \in \mathbb{R}^m$ are the plant state and control input respectively. $\boldsymbol{x}_0 := \boldsymbol{x}(0) \in \mathbb{R}^n$ is the initial plant state. $\boldsymbol{F}$ and $\boldsymbol{G}$ are known matrices with appropriate dimensions.

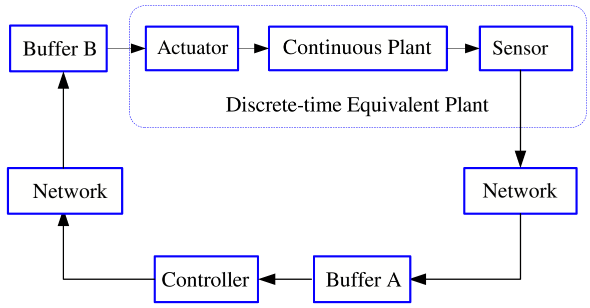

Fig. 1. The structure of the concerned networked control systems

The sensor is clock-driven, i.e., at time instant kh, it sends the most recent plant state to the controller, where h is the step length of the plant. The controller, as a receiver, has a receiving buffer denoted as buffer A. The buffer size is 1, and the packet with the latest time stamp is used to update the buffer content. The networked controller is a memoryless state feedback controller of following form:

$$\boldsymbol{u} = \boldsymbol{K}\text{buffer}(A) \tag{2}$$

where $\boldsymbol{K}$ is the feedback gain to be designed, buffer(A) is the updated content of buffer A. The controller is event-driven, i.e., whenever there is new data in buffer A, the controller starts calculating new control signal and transmits it to the actuator.

The actuator also has a buffer size of 1. When the new control packet arrives at the controller, it will be compared with the time stamp on the control signal in buffer B. The one based on the latest plant state will be put into buffer B. The actuator is clock-driven. At time instant kh, it updates the actuator output based on the value read from buffer B. Thus, the last control signal is used to control the plant when the new control signal is not available.

Remark: Note that since the actuator is clock-driven, (1) can be considered as discretized from a continuous-time system given by

$$\dot{\boldsymbol{x}}_{\rm p}(t) = \boldsymbol{A}\boldsymbol{x}_{\rm p}(t) + \boldsymbol{B}\boldsymbol{u}(t) \tag{3}$$

with sampling period h and

$$\boldsymbol{F} = e^{\boldsymbol{A}h}, \quad \boldsymbol{G} = \int_0^h e^{\boldsymbol{A}\tau} d\tau \boldsymbol{B}. \tag{4}$$

In the presence of networks, network-induced delay, packet dropout and packet out-of-order occur inevitably. As a result, not all packets from the sensor are used to control the plant. For example, the packet or the control signal based on this packet may be lost in the network, or may be discarded by the buffer because of packet out-of-order. Noting that, we adopt the following definition to capture the packet dropout in NCSs, where packet out-of-order is also considered as packet dropout.

Definition 1. A packet from the sensor is called *effective packet* under the controller (2), if the control signal based this packet is finally used to control the plant. Introduce $S := \{i_1, i_2, \cdots\} \subseteq \mathbb{Z}^+$ $(i_{m+1} > i_m,\ m \in \{1, 2, \cdots\})$ to denote the sequence of time index of the *effective packets* and let $\boldsymbol{N}_{drop} := \max_{i_m\in S}\{i_{m+1} - i_m\}$, then the packet dropout process is defined as

$$\{\eta(i_m) := i_{m+1} - i_m, \quad i_m \in S\} \tag{5}$$

which means that, from i_m to i_{m+1}, the number of packet dropout is $\eta(i_m) - 1$. Noting $i_m \in S$ and the definition of $\boldsymbol{N}_{drop}$, $\eta(i_m)$ takes values in a finite set $\Omega := \{1, 2, \cdots, \boldsymbol{N}_{drop}\}$.

Let τ_{i_m} express the RTT (Round Trip Time) delay encountered by mth effective packet, i.e., the time interval from the time instant $i_m h$ when the sensor samples the plant state to the time instant $i_m h + \tau_{i_m}$ when the control signal based on this plant state reaches the actuator. Apparently, τ_{i_m} is a combination of the sensor-to-controller delay, the controller processing delay and the controller-to-actuator delay. In this paper, without loss of generality, we assume $\tau_{i_m} \in U := [\tau_{\min},\ \tau_{\max}]$, where $\tau_{\min}$ and $\tau_{\max}$ are the lower and upper bounds of τ_{i_m} respectively. In this paper, $\tau_{\max}$ can be larger then h. From previous discussions, the time delay and packet dropout information can be embodied as

$$\{\{\eta(i_m),\ \tau_{i_m}\}:\ \ i_m \in S, \tau_{i_m} \in U, \eta(i_m) \in \Omega\} \tag{6}$$

Definition 2. (6) is said to model random time delay and arbitrary packet dropout, if τ_{i_m} takes values in U randomly, and i_m takes values in $\mathbb{Z}^+$ arbitrarily. Correspondingly, $\eta(i_m)$ takes values in Ω arbitrarily.

Let us consider $\tau_{i_m} \in [(r-1)h\ rh)$ first. Since the actuator is clock-driven, one can infer that, during the time interval between two effective packets, i.e., $[i_m h\ i_{m+1}h)$, the control input to the plant is piecewise constant and there are at most $(r+1)$ control signals, i.e., $\boldsymbol{K}\boldsymbol{x}(i_{m-j})$, $j \in \{0, 1, \cdots, r\}$. Correspondingly, there are $(r+1)$ different cases. As illustrated in Fig.2, one control signal, e.g., $\boldsymbol{K}\boldsymbol{x}(i_{m-1})$, acts on the plant when no new control signal reaches the actuator during $[i_m h\ i_{m+1}h)$. Two control signals, e.g., $\boldsymbol{K}\boldsymbol{x}(i_{m-1})$ and $\boldsymbol{K}\boldsymbol{x}(i_{m-2})$, act on the plant when one new control signal reaches the actuator during $[i_m h\ i_{m+1}h)$. By analogy, $(r+1)$ control signals, i.e., $\boldsymbol{K}\boldsymbol{x}(i_{m-j})$, $j \in \{0, 1, \cdots, r\}$, act on the plant when r control signals reaches the actuator during $[i_m h\ i_{m+1}h)$. Noting the random time delay and arbitrary packet dropout occurred in network and considering $\tau_{i_m} \in [(r-1)h\ rh)$, one can infer that the control signal $\boldsymbol{Kx}(i_m)$ can be used to control the plant only at time instant $i_m h + rh$, and the rest r control signals, i.e., $\boldsymbol{Kx}(i_{m-j})$, $j \in \{1, \cdots, r\}$, can be used to control the plant at the random time instants $i_m h + c_n^r h$, where $c_n^r \in \{0, \cdots, r-1\}$, $n \in \{1, \cdots, r\}$. For any $i, j \in \{1, \cdots, r\}$, we have

$$\begin{cases} c_i^r > c_j^r \ \ i > j, c_j^r \neq r-1 \\ c_i^r = c_j^r \ \ i > j, c_j^r = r-1 \end{cases} \tag{7}$$

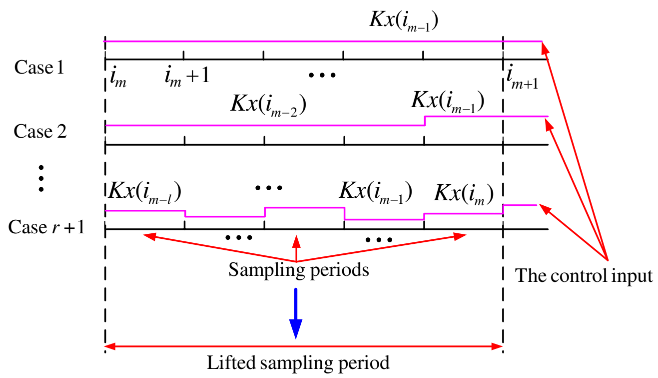

Fig. 2. A timing diagram of the concerned networked control systems.

Refer to $[i_m h\ i_{m+1} h]$ as a "lifted sampling period", then the concerned NCS can be lifted as

$$\begin{aligned} \boldsymbol{x}(i_{m+1}) = \boldsymbol{F}^{\eta(i_m)}\boldsymbol{x}(i_m) + \sum_{j=0}^{\eta(i_m)-r-1} \boldsymbol{F}^j \boldsymbol{GK}x(i_m) \\ + \sum_{j=0}^{c_1^r} \boldsymbol{F}^{\eta(i_m)-r+j}\boldsymbol{GK}x(i_{m-1}) \\ + \sum_{j=c_1^r+1}^{c_2^r} \boldsymbol{F}^{\eta(i_m)-r+j}\boldsymbol{GKx}(i_{m-2}) \\ + \cdots + \sum_{j=c_{r-1}^r+1}^{c_r^r} \boldsymbol{F}^{\eta(i_m)-r+j}\boldsymbol{GK}x(i_{m-l}) \end{aligned} \tag{8}$$

where the notation $\sum_{j=a}^{b}(\boldsymbol{F}^j)$ satisfies

$$\begin{cases} \sum_{j=a}^{b}(\boldsymbol{F}^j) = 0 \ \ b < a \\ \boldsymbol{F}^a = 0 \qquad\quad a < 0 \end{cases} \tag{9}$$

Noting the generality of r in $\tau_{i_m} \in [(r-1)h\ rh)$, the closed-loop NCS can be model as (8) with r replaced by $\lceil \tau_{i_m}/h \rceil$, where $\lceil \cdot \rceil$ denotes the nearest larger number. Let $\boldsymbol{N}_{delay} := \lceil \tau_{\max}/h \rceil$. Then one can infer that, no more than $(\boldsymbol{N}_{delay}+1)$ control signals, i.e., $\boldsymbol{Kx}(i_{m-j})$, $j \in \{0, 1, \cdots, \boldsymbol{N}_{delay}\}$, can be used to control the plant during $[i_m h\ i_{m+1} h)$. To facilitate the stability analysis of NCS, introduce $\boldsymbol{z}(i_m) = \left[\boldsymbol{x}_{i_m}^T\ \boldsymbol{x}_{i_{m-1}}^T\ \cdots\ \boldsymbol{x}_{i_{m-\boldsymbol{N}_{delay}}}^T\right]^T$ into the closed-loop NCSs model, i.e., (8) with r replaced by $\lceil \tau_{i_m}/h \rceil$. Then, the closed-loop NCSs can be expressed by the following switched system

$$\boldsymbol{z}(i_{m+1}) = \boldsymbol{M}_{\lceil \tau_{i_m}/h \rceil}\boldsymbol{z}(i_m) \tag{10}$$

where $\boldsymbol{M}_{\lceil \tau_{i_m}/h \rceil}$ is function of the switch $\lceil \tau_{i_m}/h \rceil$ which take values in a finite set $\mathbb{S} := \{1, \cdots, \boldsymbol{N}_{delay}\}$. For each possible value of $\lceil \tau_{i_m}/h \rceil = r$, $\boldsymbol{M}_r$ is of following form

$$\boldsymbol{M}_r = \begin{bmatrix} \boldsymbol{F}^{\eta(i_m)} + \boldsymbol{A}_1^r & \boldsymbol{A}_2^r & \cdots & \boldsymbol{A}_{\boldsymbol{N}_{delay}}^r & \boldsymbol{A}_{\boldsymbol{N}_{delay}+1}^r \\ I & 0 & \cdots & 0 & 0 \\ 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix} \tag{11}$$

In (11), $\boldsymbol{A}_1^r$, $\boldsymbol{A}_2^r$ and $\boldsymbol{A}_i^r$ $(i \in \{3, \cdots, r+1\})$ are given by

$$\begin{aligned} \boldsymbol{A}_1^r &= \sum_{j=0}^{\eta(i_m)-r-1} \boldsymbol{F}^j\boldsymbol{GK},\ \boldsymbol{A}_2^r = \sum_{j=0}^{c_1^r} \boldsymbol{F}^{\eta(i_m)-r+j}\boldsymbol{GK}, \\ \boldsymbol{A}_i^r &= \sum_{j=c_{i-2}^r+1}^{c_{i-1}^r} \boldsymbol{F}^{\eta(i_m)-r+j}\boldsymbol{GK}. \end{aligned} \tag{12}$$

If $r < \boldsymbol{N}_{delay}$, for (11), we also have

$$\boldsymbol{A}_i^r = 0 \qquad (i \in \{r+2, \cdots, \boldsymbol{N}_{delay}+1\}) \tag{13}$$

The objective of this paper is to investigate the stability and controller design problem for NCSs (10) such that the considered NCS is asymptotically stable.

3. STABILITY ANALYSIS OF CLOSED-LOOP NCSS.

In this section, we will give sufficient conditions for the existence of state feedback controller such that the closed-loop NCSs is asymptotically stable.

Without loss generosity, we assume that the initial control inputs are zeros, i.e., $\boldsymbol{u}(0) = 0$. Noting that the last control signal is used to control the plant when the new control signal is not available, we have $\boldsymbol{u}(l) = 0$ for $0 \leq l \leq i_1$, where i_1 is the time index of the first effective packet. With the initial plant state $\boldsymbol{x}_0 := \boldsymbol{x}(0)$, we get

$$\boldsymbol{x}(l) = \boldsymbol{F}^l\boldsymbol{x}_0 \tag{14}$$

for $0 \leq l \leq i_1$. Let $\boldsymbol{z}(0) = \left[\boldsymbol{x}_0^T\ 0\ \cdots\ 0\right]^T$. Then, for $0 \leq l \leq i_1$, we have

$$\boldsymbol{z}(l) = \hat{\boldsymbol{M}}_l\boldsymbol{z}_0 \tag{15}$$

where $\boldsymbol{z}(l) = \left[\boldsymbol{x}_l^T\ \boldsymbol{x}_{i_{m-1}}^T\ \cdots\ \boldsymbol{x}_{i_{m-\boldsymbol{N}_{delay}}}^T\right]^T$, and

$$\hat{\boldsymbol{M}}_l = \begin{bmatrix} \boldsymbol{F}^l & 0 & \cdots & 0 & 0 \\ I & 0 & \cdots & 0 & 0 \\ 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix} \tag{16}$$

Moreover, for $i_m < l < i_{m+1}$, the behavior of (10) can be expressed by

$$\boldsymbol{z}(l) = \bar{\boldsymbol{M}}_r\boldsymbol{z}(i_m) \tag{17}$$

where $i_m \in S$, $\boldsymbol{z}(l) = \left[\boldsymbol{x}_l^T\ \boldsymbol{x}_{i_{m-1}}^T\ \cdots\ \boldsymbol{x}_{i_{m-\boldsymbol{N}_{delay}}}^T\right]^T$, and $\bar{\boldsymbol{M}}_r$ is similar to (11) but with $\eta(i_m)$ replaced by $l - i_m$.

Theorem 1. With given network parameters $\boldsymbol{N}_{delay}$, $\boldsymbol{N}_{drop}$ and a given matrix $\boldsymbol{K}$, the NCS (10) in the presence of random time delay and arbitrary packet dropout is asymptotically stable, if there exists a common positive definite matrix $\boldsymbol{P}$ with appropriate dimension satisfying

$$\boldsymbol{M}_i^T \boldsymbol{P} \boldsymbol{M}_i - \boldsymbol{P} < 0 \qquad (i \in \{1, \ \cdots, \ \boldsymbol{N}_{delay}\}) \tag{18}$$

where $\boldsymbol{M}_i$ is of form (11).

Proof: Construct a Lyapunov functional as:

$$V(i_m) = \boldsymbol{z}^T(i_m)\boldsymbol{P}\boldsymbol{z}(i_m) \tag{19}$$

We first prove $\lim_{l\to\infty} \boldsymbol{z}(l) = 0$ for NCS (10). By considering (10) and by letting $\lceil \tau_{i_m}/h \rceil := i$, then we have $V(i_m) = \boldsymbol{z}^T(i_m)\boldsymbol{P}\boldsymbol{z}(i_m)$ and $V(i_{m+1}) = \boldsymbol{z}^T(i_m)\boldsymbol{M}_i^T\boldsymbol{P}\boldsymbol{M}_i\boldsymbol{z}(i_m)$. From the condition (18), one can easily show that

$$V(i_{m+1}) - V(i_m) = \boldsymbol{z}^T(i_m)(\boldsymbol{M}_i^T\boldsymbol{P}\boldsymbol{M}_i - \boldsymbol{P})\boldsymbol{z}(i_m) < 0 \tag{20}$$

for any $\boldsymbol{z}(i_m) \neq 0$.

For $i_m < l < i_{m+1}$, we have $V(l) = \boldsymbol{z}^T(i_m)\bar{\boldsymbol{M}}_i^T\boldsymbol{P}\bar{\boldsymbol{M}}_i\boldsymbol{z}(i_m)$. From (17), one can see that $\bar{\boldsymbol{M}}_i \subset \boldsymbol{M}_i$, and correspondingly, $\bar{\boldsymbol{M}}_i^T\boldsymbol{P}\bar{\boldsymbol{M}}_i - \boldsymbol{P} < 0$ is a subset of (18). Thus, if the condition (18) holds, for $i_m < l < i_{m+1}$, we can get

$$V(l) - V(i_m) = \boldsymbol{z}^T(l)\left(\bar{\boldsymbol{M}}_i^T\boldsymbol{P}\bar{\boldsymbol{M}}_i - \boldsymbol{P}\right)\boldsymbol{z}(l) < 0 \tag{21}$$

for any $\boldsymbol{z}(l) \neq 0$.

Therefore, from (20), we get $V(i_{m+1}) - V(i_m) < 0$ for any $\boldsymbol{z}(i_m) \neq 0$, which implies that $\lim_{i_m\to\infty} V(i_m) = 0$ for NCS (10). By considering (21), we can get $\lim_{l\to\infty} V(l) = 0$ for NCS (10), where $l \neq i_m$. Summarizing the above two cases, we can conclude that $\lim_{l\to\infty} V(l) = 0$ for $l \in \mathbb{Z}^+$, which implies that $\lim_{l\to\infty} \boldsymbol{z}(l) = 0$ for NCS (10).

We now prove that NCS (10) is stable if the conditions (18) hold. That is, given any $\varepsilon > 0$, we can find a $\delta(\varepsilon) > 0$ such that $\|\boldsymbol{z}(0)\| < \delta(\varepsilon)$ implies $\|\boldsymbol{z}(l)\| < \varepsilon$ for $l \in \mathbb{Z}^+$. Let $\alpha_1 = \|\boldsymbol{P}\|$ and $\alpha_2 = 1/\|\boldsymbol{P}^{-1}\|$. From the definition of Lyapunov function, we get $\alpha_2\|\boldsymbol{z}(l)\|^2 \le V(l) \le \alpha_1\|\boldsymbol{z}(l)\|^2$. In NCS (10), three cases may arise and are discussed as follows.

Case 1: $0 \le l \le i_1$. Since $\boldsymbol{z}_0 = \begin{bmatrix}\boldsymbol{x}_0^T & 0 & \cdots & 0\end{bmatrix}^T$, we have $\|\boldsymbol{z}_0\| = \|\boldsymbol{x}_0\|$. Let $\alpha_3 := \max_{l\in\{1,\cdots,\boldsymbol{N}_{drop}\}} \left\|\hat{\boldsymbol{M}}_l\right\|$, then given any $\varepsilon > 0$, if we let $\boldsymbol{x}_0 < \min\{1/\alpha_3, \ 1\}\varepsilon$, we can therefore obtain $\|\boldsymbol{z}(l)\| = \left\|\hat{\boldsymbol{M}}_l\boldsymbol{z}_0\right\| \le \left\|\hat{\boldsymbol{M}}_l\right\|\|\boldsymbol{z}_0\| < \left\|\hat{\boldsymbol{M}}_l\right\|\min\{1/\alpha_3, \ 1\}\varepsilon < \varepsilon$.

Case 2: $l = i_m$. From previous discussions, we get $V(l) < V(i_1) \le \alpha_1\|\boldsymbol{z}(i_1)\|^2 = \alpha_1\left\|\hat{\boldsymbol{M}}_{i_1}\boldsymbol{z}_0\right\|^2$ and $\|\boldsymbol{z}(l)\| \le \sqrt{V(l)/\alpha_2}$. Therefore, with any given $\varepsilon > 0$, if we let $\boldsymbol{x}_0 < (\sqrt{\alpha_2/\alpha_1}/\alpha_3)\varepsilon$, then we can get $\|\boldsymbol{z}(l)\| \le \sqrt{V(l)/\alpha_2} < \sqrt{\left\|\hat{\boldsymbol{M}}_{i_1}\right\|^2\|\boldsymbol{z}_0\|^2\alpha_1/\alpha_2} = \|\boldsymbol{z}_0\|\alpha_3\sqrt{\alpha_1/\alpha_2} < \varepsilon$.

Case 3: $i_m < l < i_{m+1}$. In this case, we have $V(l) < V(i_m) < V(i_1) \le \alpha_1\left\|\hat{\boldsymbol{M}}_{i_1}\right\|^2\|\boldsymbol{z}_0\|^2$. Let $\alpha_4 := \max_{i\in\{1,\cdots,\boldsymbol{N}_{delay}\}}\left\|\bar{\boldsymbol{M}}_i\right\|$. With any given $\varepsilon > 0$, if we let $\boldsymbol{x}_0 < \sqrt{\alpha_2/\alpha_1}/(\alpha_3\alpha_4)\varepsilon$, then we get $\|\boldsymbol{z}(l)\| = \left\|\bar{\boldsymbol{M}}_r\boldsymbol{z}(i_m)\right\| \le \left\|\bar{\boldsymbol{M}}_r\right\|\|\boldsymbol{z}(i_m)\| \le \alpha_4\|\boldsymbol{z}(i_m)\| < \|\boldsymbol{z}_0\|\alpha_3\alpha_4\sqrt{\alpha_1/\alpha_2} < \varepsilon$.

Based on the above analysis, if we let $\alpha := \min\{\min\{1/\alpha_3, \ 1\}, \ \sqrt{\alpha_2/\alpha_1}/\alpha_3, \ \sqrt{\alpha_2/\alpha_1}/(\alpha_3\alpha_4)\}$ then we can conclude that $\|\boldsymbol{z}(0)\| < \alpha\varepsilon$ implies that $\|\boldsymbol{z}(l)\| < \varepsilon$ for NCS (10), where $l \in \mathbb{Z}^+$.

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

4. STATE FEEDBACK CONTROLLER DESIGN

In this section, we consider the design of networked controller (2) such that the closed-loop system (10) is asymptotically stable.

By using Schur complement to (18), (18) is equivalent to

$$\Psi_i(\boldsymbol{Q}, \ \boldsymbol{K}) := \begin{bmatrix} -\boldsymbol{Q} & * \\ \boldsymbol{M}_i\boldsymbol{Q} & -\boldsymbol{Q} \end{bmatrix} < 0 \tag{22}$$

where $\boldsymbol{Q} = \boldsymbol{P}^{-1}$, $\boldsymbol{M}_i$ is of form (11) with variable $\boldsymbol{K}$.

From theorem 1 we see that the feasible solutions of (22) can lead to the desired networked controllers (2). Unfortunately, (22) can not be formulated into LMIs since $\boldsymbol{M}_i\boldsymbol{Q}$ involve the products between the unknown variables $\boldsymbol{K}$ and $\boldsymbol{Q}$. However, an important feature of (22) is that if $\boldsymbol{K}$ is fixed, finding $\boldsymbol{Q}$ becomes an LMI problem and vice versa. So (22) is in fact a set of bilinear matrix inequalities (BLMI). To circumvent the synthesis problem, a homotopy-based iterative LMI algorithm is developed as follows.

Define

$$\Sigma_i(\boldsymbol{Q}, \ \boldsymbol{K}, \lambda) := \begin{bmatrix} -\boldsymbol{Q} & * \\ \Sigma_i^{21} & -\boldsymbol{Q} \end{bmatrix} < 0 \tag{23}$$

where

$$\Sigma_i^{21} = \boldsymbol{H}^1\boldsymbol{Q} + \lambda\boldsymbol{H}_i^2 diag\{\boldsymbol{K}, \cdots, \boldsymbol{K}\}\boldsymbol{Q} + (1-\lambda)\boldsymbol{H}_i^2\bar{\boldsymbol{K}}$$

$$\boldsymbol{H}^1 = \begin{bmatrix} \mathbf{F}^{\eta(i_m)} & 0 & \cdots & 0 & 0 \\ I & 0 & \cdots & 0 & 0 \\ 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix}, \boldsymbol{H}_i^2 = \begin{bmatrix} \boldsymbol{A}_1^i & \boldsymbol{A}_2^i & \cdots & \boldsymbol{A}_{\boldsymbol{N}_{delay}+1}^i \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

$$\bar{\boldsymbol{K}} = diag\{\boldsymbol{K}, \cdots \boldsymbol{K}\}\boldsymbol{Q} \tag{24}$$

Note that λ is a real number varying from 0 to 1. Then

$$\Sigma_i(\boldsymbol{Q}, \ \boldsymbol{K}) = \begin{cases} \Psi_i(\boldsymbol{Q}, \ \boldsymbol{K}) & \text{if } \lambda = 1 \\ \Phi_i(\boldsymbol{Q}, \ \bar{\boldsymbol{K}}) & \text{if } \lambda = 0 \end{cases} \tag{25}$$

where

$$\Phi_i(\boldsymbol{Q}, \ \bar{\boldsymbol{K}}) := \begin{bmatrix} -\boldsymbol{Q} & * \\ \mathrm{H}^1\boldsymbol{Q} + \mathrm{H}_i^2\bar{\boldsymbol{K}} & -\boldsymbol{Q} \end{bmatrix} < 0 \tag{26}$$

Based on the above discussions, an iterative LMI algorithm can be summarized as follows.

Controller Design Procedure:

Step 1. Initiation: Set $k = 0$, select N, N_{max}. Solve $\Phi_i(\boldsymbol{Q}, \ \bar{\boldsymbol{K}})$ to get $\boldsymbol{Q}$, $\bar{\boldsymbol{K}}$, and let $\boldsymbol{Q}(0) := \boldsymbol{Q}$, $\bar{\boldsymbol{K}}(0) := \bar{\boldsymbol{K}}$.

Step 2. Set $k = k + 1$ and $\lambda_k = k/N$. Let $\boldsymbol{Q} := \boldsymbol{Q}(k-1)$, $\bar{\boldsymbol{K}} := \bar{\boldsymbol{K}}(k-1)$. If (23) upon $\boldsymbol{K}$ is feasible, then denote the feasible solution as $\boldsymbol{K}(k)$, let $\boldsymbol{Q}(k) := \boldsymbol{Q}(k-1)$, $\bar{\boldsymbol{K}}(k) := \bar{\boldsymbol{K}}(k-1)$, and go to Step 4. Otherwise, go to Step 3.

Step 3. Let $\boldsymbol{K} := \boldsymbol{K}(k-1)$ and $\bar{\boldsymbol{K}} := \bar{\boldsymbol{K}}(k-1)$. If (23) upon $\boldsymbol{Q}$ is feasible, then solve the minimization problem:

$$\begin{array}{l}\mathbf{OP}: \min \operatorname{tr}(trace(\boldsymbol{Q})) \\ \text{s.t. Inequalities (23)}\end{array} \tag{27}$$

Denote the feasible solution as $\boldsymbol{Q}(k)$, let $\boldsymbol{K}(k) := \boldsymbol{K}(k-1)$ and $\bar{\boldsymbol{K}}(k) := \bar{\boldsymbol{K}}(k-1)$, then go to Step 4. Otherwise, set $N = 2N$. If $N > N_{max}$, then the algorithm fails in giving feasible solution, else set $k = 0$, go to Step 2.

Step 4. If $k < N$, go to Step 2. If $k = N$, the obtained solutions $\boldsymbol{K}(k)$ and $\boldsymbol{Q}(k)$ are a set of feasible solutions of (22).

5. NUMERICAL EXAMPLE

In this section, a numerical example is used to validate the proposed approaches. Let us consider a NCS with setup shown in Fig.1, where the controlled plant is borrowed from Xiong et al. [2007] and given by

$$\dot{\boldsymbol{x}}_p(t) = \begin{bmatrix} -1 & 0 & -0.5 \\ 1 & -0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} \boldsymbol{x}_p(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \boldsymbol{u}(t) \tag{28}$$

When the sampling period is specified to 0.1s, the discretized system is of form (1) with

$$\boldsymbol{F} = \begin{bmatrix} 0.9048 & 0.0000 & -0.0488 \\ 0.0928 & 0.9512 & -0.0024 \\ 0.0000 & 0.0000 & 1.0513 \end{bmatrix} \quad \boldsymbol{G} = \begin{bmatrix} -0.0025 \\ -0.0001 \\ 0.1025 \end{bmatrix} \tag{29}$$

To carry out numerical simulations, we assume the RTT delay τ_{i_m} is random, ranging from 0ms to 195ms, and in three consecutive packets from sensor, there is at least one effective packet. From the above assumptions, we get $\boldsymbol{N}_{delay} = 2$ and $\boldsymbol{N}_{drop} = 3$. In the following text, we will consider three controller design methods, and compare the corresponding NCSs performance. For fair comparisons, the three controller design methods are selected with the same form $\boldsymbol{u} = \boldsymbol{K}\boldsymbol{x}$. The main difference between them resides in that they are developed with different network effects taken into account.

Case 1: The controller considering only packet dropout

Let us consider the state feedback controller design method proposed by Xiong et al. [2007], where the controller is designed with only packet dropout being considered. We apply the Theorem 11 in Xiong et al. [2007] to the concerned NCS (i.e., we design the networked controller considering only packet dropout, without considering time delay), and therefore obtain $\boldsymbol{K} = [0.5716, 0.1591, -4.6618]$. When this feedback gain matrix is used to control the NCS in the presence of time delay and packet dropout, we can conclude that it leads to a unstable NCS by using the obtained stability criterion equation (18). With the initial state $\boldsymbol{x}_0 = [-5, 0, 5]^T$, the simulation result of NCS under $\boldsymbol{u} = [0.5716, 0.1591, -4.6618]\,\boldsymbol{x}$ is shown in Fig.3, which indicates that the networked controller considering only packet dropout is not capable of guaranteeing a stable NCS for this special example. Clearly the simulation result is consistent with the analytical stability criterion.

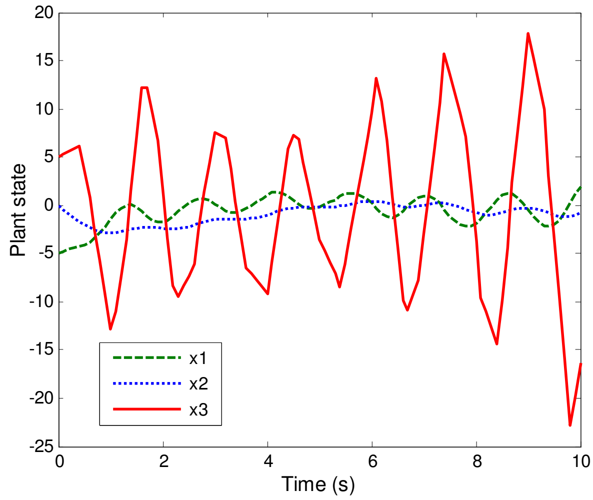

Fig. 3. Typical NCS performance under a networked controller considering only packet dropout.

Case 2: The controller considering only time delay

We apply the proposed controller design method considering only time delay, without considering packet dropout, to the concerned NCS, and therefore obtain $\boldsymbol{K} = [-0.5530, -0.4318, -0.1040]$. Using the obtained closed-loop stability criterion, we see that when time delay and packet dropout are present, the designed feedback matrix results in an unstable NCS. With the same initial state, the simulation result of NCS under $\boldsymbol{u} = [-0.5530, -0.4318, -0.1040]\,\boldsymbol{x}$ is depicted in Fig.4, which implies that the networked controller considering only time delay is also not capable of guaranteeing a stable NCS in this case. Apparently, the simulation result is also consistent with the analytical stability criterion.

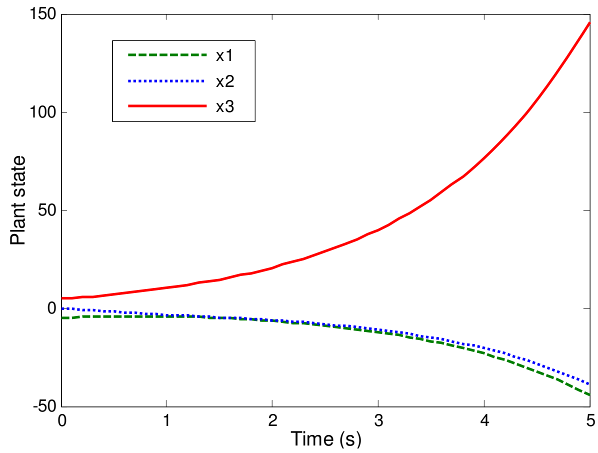

Fig. 4. Typical NCS performance under a networked controller considering only time delay.

Case 3: The proposed state feedback controller

Now, we apply the proposed controller design method considering both time delay and packet dropout to the concerned NCS. Solving the Design Procedure, we get $\boldsymbol{K} = [-0.1943, -0.1373, -1.3080]$. With the same initial state, the simulation result of NCS under $\boldsymbol{u} = [-0.1943, -0.1373, -1.3080]\,\boldsymbol{x}$ is depicted in Fig.5, which demonstrates that the proposed controller can stabilize the concerned NCS very well.

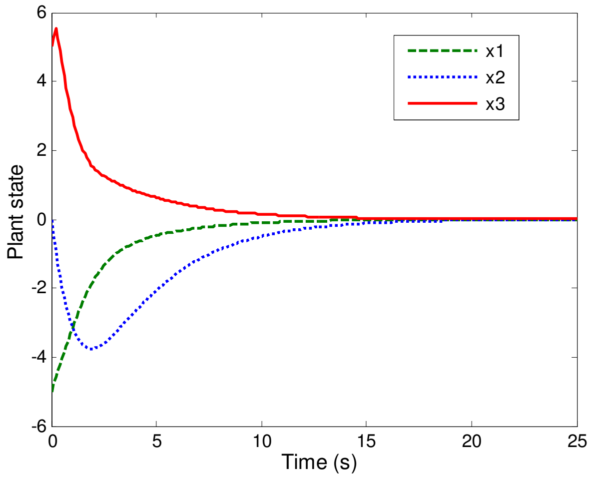

Fig. 5. Typical NCS performance under the proposed controller.

As can be seen in this example, the discretized plant is unstable because of eig($\boldsymbol{F}$) $= 0.9512, 0.9048, 1.0513$. The networked controller considering only network-induced delay or packet dropout can not stabilize the unstable NCS. However, under the proposed controller, the NCS is not only stable but also shows good performance. This demonstrates demonstrate the effectiveness of the proposed approaches.

6. CONCLUSIONS

This paper has investigated the state feedback controller design and stability analysis problems for NCSs under effects of network-induced delay and packet dropout. A discrete-time switch model is proposed by introducing the lifting technique into the considered NCSs, which enables us to apply the theory from switch systems to study NCSs in discrete-time domain. In the proposed framework, sufficient conditions for the existence of state feedback controller such that the closed-loop NCSs is asymptotically stable is derived and the corresponding controller design problem is also addressed. Simulation results are given to demonstrate the effectiveness of the proposed approaches.

REFERENCES

J. Nilsson, B. Bernhardsson and B. Wittenmark. Stochastic analysis and control of real–time systems with random time delays. *Automatica*, 34:57–64, 1998.

S.S. Hu and Q.X. Zhu. Stochastic optimal control and analysis of stability of networked control systems with long delay. *Automatica*, 39:1877–1884, 2003.

Y. Tipsuwan and M.Y. Chow. On the Gain Scheduling for Networked PI Controller Over IP Network. IEEE/ASME Transaction on Mechatronics, 9(3):491–498, 2004.

G.P. Liu, J. X. Mu, D. Rees and S.C. Chai. Design and stability analysis of networked control systems with random communication time delay using the modified MPC. *International Journal of Control*, 79(4):287–296, 2006.

W. Zhang, M.S. Branicky and S.M. Phillips. "Stability of networked control systems. *IEEE Control Systems Magazine*, 21(1):84–99, 2001.

J.L. Xiong and J. Lam. Stabilization of linear systems over networks with bounded packet loss. *Automatica*, 43(1): 80–87, 2007.

L. A. Montestruque and P. J. Antsaklis. On the model–based control of networked systems. *Automatica*, 39: 1837–1843, 2003.

V. Z. Peter and H. M. Richard. Networked control design for linear systems. *Automatica*, 39:743–750, 2003.

K.Y. Li and J. Baillieul. Robust Quantization for Digital Finite Communication Bandwidth (DFCB) Control. *IEEE Transactions on Automatic Control*, 49(9):1573–1584, 2004.

L.A. Montestruque and P.J. Antsaklis. Static and dynamic quantization in model–based networked control systems. *International Journal of Control*, 80(1):87–101, 2007.

G.C. Walsh and H. Ye. "Scheduling of networked control systems." *IEEE Control Systems Magazine*, 21(1):57–65, 2001.

D. Yue, Q.L. Han and P. Chen. State feedback controller design of networked control systems. *IEEE Transactions on Circuits and Systems*, 51(11):640–644, 2004.

P. Seiler and R. Sengupta. An H_∞ Approach to Networked Control. *IEEE Transactions on Automatic Control*, 50(3):356 –364, 2005.

Y. Tipsuwan and M.Y. Chow. Control methodologies in networked control systems. *Control Engineering Practice*, 11:1099–1111, 2003.

M. Yu, L. Wang, T. Chu, F. Hao. An LMI Approach to Networked Control Systems with Data Packet Dropout and Transmission Delays. *Proceedings of the 43th IEEE Conference on Decision and Control*, pages 3545–3550, 2004.

D. Yue, Q.L. Han and J. Lam. Network–based robust H_∞ control of systems with uncertainty. *Automatica*, 41(6): 999–1007, 2005.

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. *Proceedings of the 43th IEEE Conference on Decision and Control*, pages 3539–3544, 2004.

H. Lin and P. J. Antsaklis. Stability and persistent disturbance attenuation properties for a class of networked control systems:switched system approach. *International Journal of Control*, 78(18):1447–1458, 2005.

D.G. Li, S. L. Shah and T.W. Chen. Analysis of dual–rate inferential control systems. *Automatica*, 38(6):1053–1059, 2002.

K. S. Park, Y. H. Choi, Z. Li and N.H. Kim. Design of H2 Controllers for Sampled–Data Systems with Input Time Delays. *Real–Time Systems*, 26(3):231–261, 2004.