Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

WeC06.2

Optimization and Stabilization of Networked Control Systems: an Estimation of Distribution Algorithm approach

HongBo Li*,**, Mo-Yuen Chow*, and ZengQi Sun**

hli7@ncsu.edu, chow@ncsu.edu, szq-dcs@mail.tsinghua.edu.cn

Abstract— This paper addresses the optimization and stabilization problem for networked control systems (NCSs). The memoryless state feedback controller is considered, and the resulting closed-loop NCS is modeled as a discrete-time switch system. By defining a state-dependent 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 Estimation of Distribution Algorithm (EDA) is used to select the optimal state feedback stabilizing gain. It is shown that the proposed method can be easily implemented to various applications, since it is simple and has no assumptions on time delays and packet losses. Simulation results are given to demonstrate the effectiveness of the proposed approach.

I. Introduction

Networked control systems (NCSs) are spatially distributed control systems with the control loops closed over communication networks. Recently, NCSs have attracted significant attention from research communities and a wealth of literature have appeared. For example, the discussions of NCSs under effects of time delays [1]–[6], packet losses [7], [8], network constraints [9], [10], signal quantization [11], [12], and scheduling [13] were presented and some useful results were reported. In addition, due to the advantages of reduced system wiring, simple installation, increased system flexibility, and the great benefits from sharing of the resources [6], [8], NCSs have been finding applications in DC motors [4], robots [5] and vehicles [14], etc.

Time delays and packet losses are two major causes for the NCSs performance deterioration and potential NCSs instability. On the other hand, memoryless state feedback control is a simple and effective control method. It has a simple structure and does not require controller memory to store a large amount of past information such as historical plant state values. Therefore, memoryless state feedback control for NCSs with both time delays and packet losses has received considerable attention in the past few years. The memoryless state feedback control for NCSs, either in the continuous-time domain ([15], [16], [17]), or in the discrete-time domain ([8], [19]–[22]), have been investigated with results reported in the literature. It is noticed that, for the memoryless state feedback controller design methods in the aforementioned results, most of them are derived from the stability conditions, and none of them considered the control

performance of NCSs. As a results, the designed memoryless state feedback controller can only guarantee the NCSs to be asymptotically stable while the control performance of NCSs may be not satisfied. However, in control practice, it not only requires a NCS to be asymptotically stable, but also requires 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 the present study.

In this paper, we investigate the memoryless state feedback control for NCSs in the discrete-time domain, where both stability and performance are considered during the controller design. For the NCSs with time delays and packet losses, this paper proposes a new discrete-time switch system. In terms of the given model, the stability conditions are derived for NCSs in terms of LMIs. Based on the obtained stability conditions, the corresponding controller design problem is solved, and Estimation of Distribution Algorithm (EDA) is used to select the optimal state feedback stabilizing gain. It is shown that the proposed method can be easily implemented to various applications, since it is simple and has no assumptions on the time delays and the packet losses. Simulation results demonstrate that the proposed controller shows better performance than the existing memoryless state feedback stabilizing controllers.

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. 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

The NCS considered in this paper is depicted in Fig.1, where the plant is modeled by a linear discrete-time system:

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

where $\boldsymbol{x}(k) \in \mathbb{R}^n$ and $\boldsymbol{u}(k) \in \mathbb{R}^m$ are the plant state and the control input, respectively. $\boldsymbol{F}$ and $\boldsymbol{G}$ are known matrices with appropriate dimensions.

The sensor is time-driven. At each sampling period, the sampled plant state and the time it is sampled (i.e., timestamp) are encapsulated into a packet and sent to the controller via the network. The timestamp will ensure the controller determine the order of the sensor packets, and
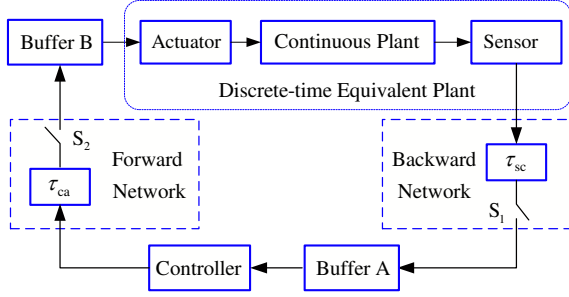
Fig. 1.   The structure of the concerned networked control systems

correspondingly select the right one to compute the control signal. In practice, the data packets in NCSs usually suffer time delays and packet losses during network transmissions. As shown in Fig.1, for the considered NCSs, we use τ_{sc} and τ_{ca} to denote the sensor-to-controller delay and the controller-to-actuator delay, and use two switches S_1 and S_2 to model the packet losses in the backward network and the forward network. For example, when S_1 is open, the sensor packet is lost during the 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.

The controller has a buffer denoted as buffer A. The buffer size used in this paper is 1, which implies that the sensor packet with the latest timestamp is used to update the buffer content. In more details, when a sensor packet arrives at the controller, it will be compared with the timestamp of the plant state in buffer A. The recent plant state (i.e., the one with the recent timestamp) and its timestamp will be put into buffer A. The networked controller is a memoryless state feedback controller with the following form:

$$\boldsymbol{u} = \boldsymbol{K}\mathrm{buffer}(A), \qquad (2)$$

where $\boldsymbol{K}$ is the feedback gain to be designed, $\mathrm{buffer}(A)$ is the updated plant state in buffer A. The controller is event-driven, i.e., whenever there is new data in buffer A, 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 in a packet and sent to the actuator via the network. The timestamp will ensure the actuator select the right control signal to control the plant.

The actuator is time-driven. It is assumed that the actuator and the sensor have the same sampling period h and they are synchronized. As shown in Fig.1, the actuator also has a buffer size of 1, which means that the latest control packet is used to control the plant. For analysis purpose, we let $\tau_k \triangleq \tau_{sc}(k) + \tau_{ca}(k)$ express the Round Trip Time (RTT) delay encountered by kth 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 paper can be easily extended to the case $\tau_k > 2h$. The actuator uses a zero-order hold (ZOH). At each sampling period, the ZOH reads the control signal from buffer B and uses it to control the plant.

Remark 1: Note that the actuator and the sensor are at the same location. Theretofore, the synchronization between the actuator and the sensor can be easily archived by hardware synchronization, for instance, using special wiring to distribute a global clock signal to the sensor and the actuator. It also worths noting that, the plant (1) can be considered as discretized from a continuous-time system given by

$$\dot{\boldsymbol{x}}_{\mathrm{p}}(t) = \boldsymbol{A}\boldsymbol{x}_{\mathrm{p}}(t) + \boldsymbol{B}\boldsymbol{u}(t), \qquad (3)$$

with sampling period h and

$$\boldsymbol{F} = e^{\boldsymbol{A}h}, \qquad \boldsymbol{G} = \int_0^h e^{\boldsymbol{A}\tau} d\tau \boldsymbol{B}. \qquad (4)$$

III. MODELING OF NCSs

This section will present a discrete-time switch model for NCSs. To address this problem, we introduce the definition of an *effective sensor packet* as follows.

Definition 1. A packet from the sensor is called an *effective sensor packet* for the controller (2), if its RTT delay is no longer than $2h$.

Let $S \triangleq \{i_1, i_2, \cdots\}$, a subsequence of $\{0, 1, 2, \cdots\}$, denote the sequence of time index of *effective sensor packets*. As mentioned in the previous section, for the NCSs under investigation, only *effective sensor packets* are used to control the plant. Therefore, the sensor packets between two *effective sensor packets*, i.e., the sensor packets with time index $k \in (i_m, i_{m+1})$, can all be considered as dropped packets, where $i_m \in S$. Based on this idea, the following mathematical model is used to capture the nature of packet losses in NCSs.

Definition 2. The packet losses process in the NCSs is defined as

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

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 $\boldsymbol{N}_{drop} \triangleq \max_{i_m \in S} \{\eta(i_m)\}$. Then we can conclude that $\eta(i_m)$ takes values in a finite set $\Omega \triangleq \{1, 2, \cdots, \boldsymbol{N}_{drop}\}$.

Let τ_{i_m} express the RTT delay encountered by the mth *effective sensor packet*. As illustrated in Fig.2, three cases may arise in NCSs during the time interval between two *effective sensor packets*, i.e., $[i_m h, i_{m+1} h]$. As discussed in [21], the dynamics of NCSs corresponding to the three different cases can be described as three different subsystems. If we introduce the augmented state $\boldsymbol{z}(i_m) = \left[\boldsymbol{x}_{i_m}^T, \boldsymbol{x}_{i_{m-1}}^T, \boldsymbol{x}_{i_{m-2}}^T\right]^T$ into the three different subsystems, the three subsystems can
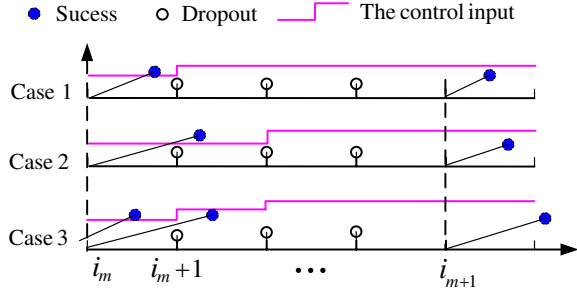
Fig. 2. The illustration of time delays and packet losses of NCSs.

be synthesized into a general framework, which is described by the following *discrete-time switch system*:

$$z(i_{m+1}) = M_{r(i_m)}z(i_m) \tag{6}$$

where $r(i_m)$ is a piecewise constant function called a switching signal, which takes values in a finite set $\mathbb{S} \triangleq \{1,2,3\}$. For $r(i_m) \in \mathbb{S}$, we have

$$M_1 = \begin{bmatrix} \Upsilon + \sum_{j=0}^{\eta(i_m)-2} F^r GK & F^{\eta(i_m)-1}GK & 0 \\ I & 0 & 0 \\ 0 & I & 0 \end{bmatrix} \tag{7}$$

$$M_2 = \begin{bmatrix} \Upsilon + \sum_{j=0}^{\eta(i_m)-3} F^r GK & \sum_{j=\eta(i_m)-2}^{\eta(i_m)-1} F^r GK & 0 \\ I & 0 & 0 \\ 0 & I & 0 \end{bmatrix} \tag{8}$$

$$M_3 = \begin{bmatrix} \Upsilon + \sum_{j=0}^{\eta(i_m)-3} F^r GK & F^{\eta(i_m)-2}GK & F^{\eta(i_m)-1}GK \\ I & 0 & 0 \\ 0 & I & 0 \end{bmatrix} \tag{9}$$

where $\Upsilon = F^{\eta(i_m)}$, $\eta(i_m) \in \Omega$.

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

$$z(l) = \bar{M}_{r(i_m)}z(i_m) \tag{10}$$

where $r(i_m) \in \mathbb{S}$, $z(l) \triangleq \begin{bmatrix} x_l^T & x_{i_{m-1}}^T & x_{i_{m-2}}^T \end{bmatrix}^T$, $\bar{M}_{r(i_m)}$ is similar to (7)-(9) but with $\eta(i_m)$ replaced by $l - i_m$.

For more details of the modeling process and the used notations, please refer to our previous work [21]. [21] has also shown that networked DC motor system (6) in the presence of time delays and packet losses is asymptotically stable, if for $i, j \in \mathbb{S}$, there exist positive definite matrices $P_i \in \mathbb{R}^{3n \times 3n}$, $P_j \in \mathbb{R}^{3n \times 3n}$ satisfying

$$M_i^T P_j M_i - P_i < 0 \tag{11}$$

where M_i is of form (7)-(9).

IV. THE OPTIMAL STABILIZING GAIN SELECTION

This section will present a optimal state feedback stabilizing gain selection method. In this paper, the optimal

stabilizing gain design problem is transformed into an optimization problem, and then solved based on a heuristic search algorithm.

• The optimization variables

In the optimization problem, the optimization variables are the elements of the controller gain K. For future use, we transform the optimization variables into a vector:

$$V = (v_1, v_2, \ldots, v_{\bar{S}}) \tag{12}$$

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

• The cost function

Different engineering applications have their performance measures, which can be used to construct their cost function. In this paper, the *mean square error* is used to evaluate the NCSs step response. The corresponding cost function is defined as follows:

$$J = \frac{1}{N}\sum_{k=0}^{N}\left(r(k) - y(k)\right)^2 \tag{13}$$

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

• Stable Region Constraint (SRC)

A set $\Psi \subset \mathbb{R}^{1 \times \bar{S}}$ is called the *Stable Region* of the concerned NCSs, if any element in Ψ can satisfy the stability conditions described in (11), i.e., Ψ is composed of controller gains 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 defined *Stable Region*, since satisfactory control performance not only requires the system to be asymptotically stable, but also require the system to meet some performance specifications. Therefore, it is sufficient to search the optimal controller gain within the *Stable Region*, which can effectively reduce the search space and guarantee the stability of NCSs during the optimization procedure. Based on this idea, *Stable Region Constraint* (SRC) is introduced as follows:

$$\text{SRC}: V \in \Psi, \quad \forall V \tag{14}$$

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

In summary, the optimization problem can be expressed as:

$$\begin{aligned} &\text{OP}: \min J \\ &\text{s.t. SRC}: V \in \Psi, \quad \forall V \end{aligned} \tag{15}$$

Obviously, the SRC can guarantee the stability of NCSs. 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 (15).

In this paper, a population-based, heuristic search algorithm namely Estimation of Distribution Algorithm (EDA) is used to solve the optimization problem (15). EDA has recently been recognized as a new computing paradigm in

evolutionary computation. Unlike other Evolutionary Algorithms, EDA does not use crossover or mutation. Instead, it explicitly extracts global statistical information from the selected solutions, and subsequently builds a probability model of promising solutions based on the extracted information. Thus, the relationships between the variables are explicitly and effectively captured and exploited. For more details on EDA, please refer to [23] and [24]. A typical EDA algorithm is used in this paper, where the optimization variables in the search space are modeled as a multivariate normal distribution $p(q) = \prod_{i=1}^{\bar{S}} p(q_i)$, which is a product of $\bar{S}$ independent univariate normal distributions $p(q_i)$. As a result, the solutions of EDA can be replaced by two vectors, the mean values of Gaussian normal distribution, μ_i, and the standard deviation, σ_i. No interactions among variables are considered. The EDA algorithm used in this paper can be summarized as follows:

1. Under *Stable Region Constraint*, generate W_1 individuals randomly in search space to form an initial population.

2. Repeat the following steps:

　　a. Select the best W_2 individuals from the parent generation, where $W_2 \leq W_1$.

　　b. Update the multivariate Gaussian distribution using the selected individuals according to

$$\mu_i = \frac{1}{W_2} \sum_{j=1}^{W_2} v_i^j \qquad (16)$$

$$\sigma_i = \sqrt{\frac{1}{W_2} \sum_{j=1}^{W_2} (v_i^j - \bar{v}_i^j)^2} \qquad (17)$$

where the selected W_2 individuals $\bar{V}_i$ $(i = 1, \ldots, W_2)$ are denoted as $\bar{V}_i = (v_1^i, \ldots, v_{\bar{S}}^i)$, $\bar{v}_i^j$ is the mean of v_i^j, $j = 1, \ldots W_2$.

　　c. Generate the next population: The best individual is copied to the next population. Under *Stable Region Constraint*, the rest $W_1 - 1$ individuals are generated with the new multivariate normal distribution.

Remark 2: Although we do not know the exact *Stable Region* of a NCS, we can realize the *Stable Region Constraint* based on the stability conditions described in (11). Taking Step 1 for example, we can generate an individual candidate in the search space randomly, and then check whether it satisfies the stability conditions described in (11). If it does, it will be selected as an individual of the population. Otherwise, we discard the candidate point and repeat the above steps until we obtain one satisfying the stability conditions described in (11). Similar arrangement is for Step 2-c.

V. ILLUSTRATIVE EXAMPLES

Consider a NCS with setup shown in Fig.1, where the controlled plant was used in [15] and [19], and is given by

$$\dot{x}_p(t) = \begin{bmatrix} 0 & 1 \\ 0 & -0.1 \end{bmatrix} x_p(t) + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} u(t). \qquad (18)$$

When the plant is sampled with a sampling period $h = 0.6s$, the discretized plant is system (1) with

$$F = \begin{bmatrix} 1.0000 & 0.5824 \\ 0 & 0.9418 \end{bmatrix}, \quad G = \begin{bmatrix} 0.0176 \\ 0.0582 \end{bmatrix}. \qquad (19)$$

Due to the application variations of comparative control methods, two operating conditions are considered and discussed as follows.

Case 1.1: The RTT delays are less than h.

In this scenario, we assume that $\tau_k \in (0s, 0.6s]$ and $N_{drop} = 5$, which means that the RTT delays encountered by the sensor packets are less than h and up to 4 consecutive packets can be lost during the network transmissions. For the above NCS, [19] designed a non-networked controller $u = [-3.75 - 11.5] x$. With the initial state $x_0 = [-2, 0]^T$ and output equation $y(k) = [1, 0] x(k)$, the simulation result of the NCS under this controller is depicted by the dot-dashed line in Fig.3. The corresponding network condition is shown in Fig.4, where the time delays are represented by the height of stem and the packet losses are indicated with circles on the packet number axes. Then, by using the Theorem 11 in [8] to this NCS, we obtain a discrete-time state feedback controller $u = [-0.6989, -3.6859] x$. For fair comparisons, the network condition shown in Fig.4 is used and the simulation result of the NCS with this networked controller is depicted by the dashed line in Fig.3. Now, we apply the Theorem 1 in [15] to this NCS, and obtain a continuous-time state feedback controller $u = [-0.2183, -0.7774] x$. With the network condition shown in Fig.4, the simulation result of the NCS using this controller is depicted by dot-solid line in Fig.3. Finally, let us consider the proposed OSFC controller design method. This simulation uses a population size of 50 (i.e., $W_1 = 50$ in the EDA algorithm), with the best 20 individuals (i.e., $W_2 = 20$ in the EDA algorithm) selected from the parent generation to update the multivariate Gaussian distribution for the next generation. Applying the OFSC controller design method to the above NCS, we obtain $u = [-0.9965 - 4.2252] x$. Fig.5 depicts the corresponding population distribution of EDA at different generations and scales, where the circles represent the individuals of population and the crosses represent the discarded individual candidates based on the *Stable Region Constraint* criteria. Fig.5 clearly shows that, from generation to generation, the population evolves towards the optimal solution and the search space gets smaller and smaller. From Fig.5(a), we can also approximate the boundary of the *Stable Region* for the NCS. With the network condition shown in Fig.4, the simulation result of the NCS using the OFSC controller is depicted by the solid line in Fig.3. Fig.3 clearly shows that, the proposed OSFC controller achieves much better performance than the three comparative ones. This comes no surprise since the optimization of the control performance is only considered in the OSFC controller design.

Remark 3: As shown in Fig.5, by sampling from a probability model built on the promising solutions in the parent generation, EDA can generate a new population in the *Stable Region* for the next generation. It avoids unnecessary
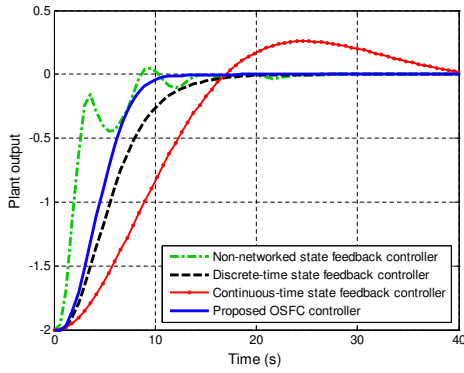
Fig. 3. Typical NCS performance using different networked controllers.
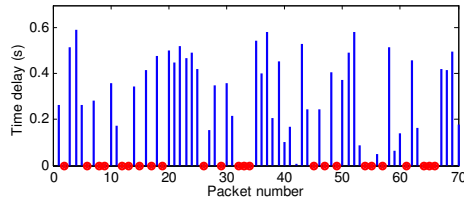

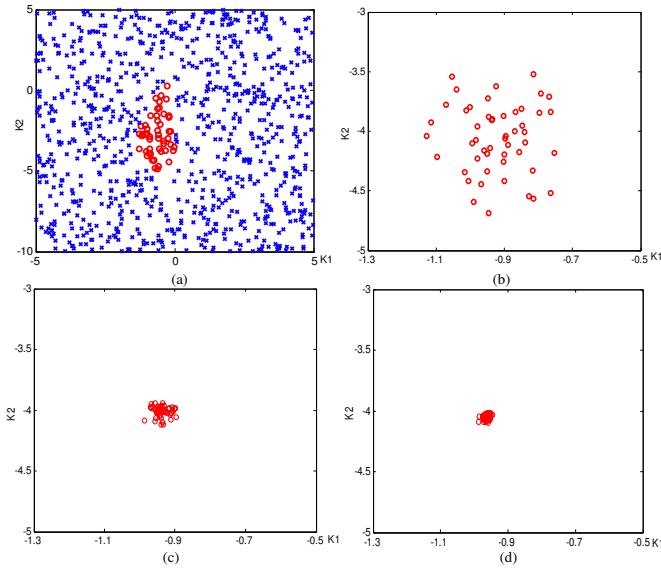Fig. 4. The network condition used in the simulation.

Fig. 5. The population distribution of EDA at different generations and scales, (a) the initial generation, (b) the 5th generation, (c) the 10th generation, (d) the 40th generation.

repeating steps as discussed in Remark 2, and provides faster search speed by making the search space smaller and smaller. This is the main reason of using EDA in this paper.

Case 1.2: The RTT delays can be larger than h.

In this scenario, we assume that $N_{drop} = 5$ and $\tau_k \in (0.05s, 1.9s]$, where τ_k can be larger than h. Note that the methods in [8] and [19] assumed τ_k is either equal to h or random but smaller than h. Therefore, the methods in [8] and [19] can not be applied to this NCS. We apply the Theorem 1 in [15] to the above NCS, and obtain a continuous-time state feedback controller $u = [-0.0591, -0.1190]\, x$.

With the initial state $x_0 = [-2, 0]^T$ and output equation $y(k) = [1, 0]\, x(k)$, the typical simulation result of the NCS using this controller is depicted by dashed line in Fig.6, where the corresponding network condition is shown in Fig.7. Now, we apply the proposed OSFC controller design method to this NCS, and obtain $u = [-0.9406, -4.4168]\, x$. With the network condition shown in Fig.7, the simulation result of the NCS using the OSFC controller is depicted by the solid line in Fig.6. Fig.6 clearly shows that the proposed SFSC controller achieves much better performance than the published one in [15].
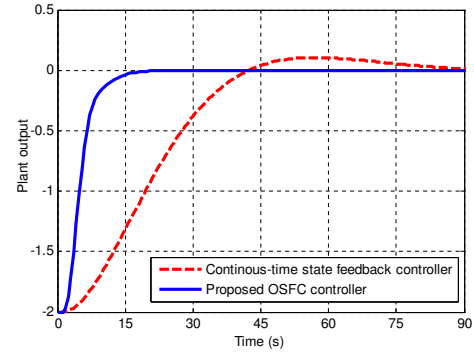

Fig. 6. Typical NCS performance using different networked controllers.
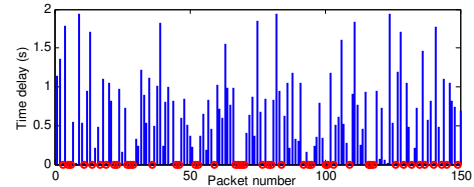

Fig. 7. The network condition used in the simulation.

Case 1.1 and Case 1.2 demonstrate that the proposed method shows more generality than the ones in [8] and [19], and shows much better control performance than the ones in [8], [19], and [15].

VI. CONCLUSIONS

This paper has investigated the optimization and stabilization problem for NCSs with time delays and packet losses. The memoryless state feedback controller is considered, and the resulting closed-loop NCS is modeled as a discrete-time switch system. By defining a state-dependent Lyapunov function, the stability conditions are derived for NCSs in terms of LMIs. A controller design technique with both system stability and control performance taken into account is proposed to design the corresponding controller. Simulation results demonstrate the effectiveness of the proposed approach.

REFERENCES

[1] J. Nilsson, B. Bernhardsson, and B. Wittenmark, "Stochastic analysis and control of real-time systems with random time delays," *Automatica*, vol. 34, pp. 57-64, 1998.

[2] S. Hu and Q. Zhu, "Stochastic optimal control and analysis of stability of networked control systems with long delay," *Automatica*, vol. 39, pp. 1877-1884, 2003.

[3] Y. Tipsuwan and M.-Y. Chow, "On the gain scheduling for networked PI controller over IP network," *Mechatronics, IEEE/ASME Transactions on*, vol. 9, pp. 491-498, 2004.

[4] 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*, vol. 79, pp. 288C297, 2006.

[5] Y. Tipsuwan and M.-Y. Chow, "Gain scheduler middleware: a methodology to enable existing controllers for networked control and teleoperation-part II: teleoperation," *Industrial Electronics, IEEE Transactions on*, vol. 51, pp. 1218-1227, 2004.

[6] L. Zhang, Y. Shi, T. Chen, and B. Huang, "A New Method for Stabilization of Networked Control Systems With Random Delays," *Automatic Control, IEEE Transactions on*, vol. 50, pp. 1177-1181, 2005.

[7] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *Control Systems Magazine, IEEE*, vol. 21, pp. 84-99, 2001.

[8] J. Xiong and J. Lama, "Stabilization of linear systems over networks with bounded packet loss," *Automatica*, vol. 43, pp. 80-87, 2007.

[9] L. A. Montestruque and P. J. Antsaklis, "On the model-based control of networked systems," *Automatica*, vol. 39, pp. 1837-1843, 2003.

[10] P. V. Zhivoglyadov and R. H. Middleton, "Networked control design for linear systems," *Automatica*, vol. 39, pp. 743-750, 2003.

[11] K. Li and J. Baillieul, "Robust quantization for digital finite communication bandwidth (DFCB) control," *Automatic Control, IEEE Transactions on*, vol. 49, pp. 1573-1584, 2004.

[12] L. A. Montestruque and P. J. Antsaklis, "Static and dynamic quantization in model-based networked control systems," *International Journal of Control*, vol. 80, pp. 87-101, 2007.

[13] G. C. Walsh and H. Ye, "Scheduling of networked control systems," *Control Systems Magazine, IEEE*, vol. 21, pp. 57-65, 2001.

[14] P. Seiler and R. Sengupta, "An H_∞ Approach to Networked Control," *Automatic Control, IEEE Transactions on*, vol. 50, pp. 356-364, 2005.

[15] D. Yue, Q.-L. Han, and C. Peng, "State feedback controller design of networked control systems," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 51, pp. 640-644, 2004.

[16] 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*, pp. 3545-3550, 2004.

[17] D. Yue, Q.-L. Han, and J. Lam, "Network–based robust H_∞ control of systems with uncertainty," *Automatica*, vol. 41, pp. 999-1007, 2005.

[18] 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*, pp. 3539-3544, 2004.

[19] M. G. Rivera and A. Barreiro, "Analysis of networked control systems with drops and variable delays." *Automatica*, vol. 43, pp. 2054-2059, 2007.

[20] 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*. (to be published).

[21] 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," *Proc. IFAC World Congress*, 2008.

[22] H. Li, Z. Sun, M.-Y. Chow, and B. Chen. "State feedback controller design of networked control systems with time delay and packet dropout," *Proc. IFAC World Congress*, 2008.

[23] P. Larranaga, J. A. Lozano, "Estimation of Distribution Algorithms. A New Tool for Evolu-tionary Computation," Kluwer Academic Publishers, 2001.

[24] C. González, "Contributions on Theoretical Aspects of Estimation of Distribution Algorithms," Ph.D. dissertation, University of the Basque Country, Spain, 2005.