# Input Design-Based Compensation Control for Networked Nonlinear Systems With Random Delays and Packet Dropouts

Zhong-Hua Pang, *Member, IEEE,* Guo-Ping Liu, *Fellow, IEEE,*
Donghua Zhou, *Senior Member, IEEE,* and Dehui Sun, *Member, IEEE*

***Abstract*—This brief investigates the data-based networked control problem of a class of nonlinear systems, where random network-induced delays and packet dropouts in the feedback and forward channels are considered simultaneously and further treated as random round-trip time (RTT) delays. The main contributions of this brief are as follows: 1) to actively compensate for RTT delays, a novel compensation control scheme is proposed based on the control input design, and thus, only one control command needs to be transmitted to the actuator through network; 2) an explicit sufficient condition is derived to ensure the stability of the resulting closed-loop system as well as a zero steady-state output error for a constant reference input; and 3) numerical simulation and comparison with existing methods are carried out to show the effectiveness of the proposed method.**

***Index Terms*—Networked control systems (NCSs), nonlinear systems, data-based control, network-induced delay, packet dropout, compensation, stability analysis.**

## I. Introduction

IN the last two decade, considerable attention has been paid to networked control systems (NCSs), due to their advantages such as long-distance data exchange and sharing, low installation and maintenance cost, high flexibility and reliability, easy reconfigurability, and increasing mobility. However, the utilization of network in the control loop also brings various communication constraints such as random network-induced delays and packet dropouts, which may seriously degrade the system performance or even make the closed-loop system unstable. Therefore, various interesting approaches have been presented for the analysis and design of NCSs with network-induced delays or/and packet dropouts [1]-[3].

One typical approach to effectively cope with the above network-induced constraints is model-based networked predictive control (MBNPC) methods (see, e.g., [4]-[18]), which take full advantage of the feature of NCSs such as packet-based transmission mechanism, as well as smart sensor and actuator. However, in practical applications, most of the existing MBNPC methods suffer from the following drawbacks: (i) Most of the available results are focused on linear plants [4]-[13], and very limited results are for nonlinear plants [14]-[18]. (ii) The performance of these MBNPC methods significantly depends on the accurate model or necessary uncertainty knowledge of the controlled plant. (iii) To compensate for all possible network-induced delays and packet dropouts, a large number of candidate control commands are needed to be transmitted to the actuator in one packet through network.

To overcome the above drawbacks (i) and (ii), a complementary approach is data-based control (DBC) methods, which have received a great deal of attention in recent years [19]-[26]. However, most of the existing DBC methods are developed for traditional control systems equipped with dedicated hardwired links, and quite few results are available for NCSs, which are reviewed as follows. In [27], a data-driven predictive control scheme was designed for linear NCSs by using the subspace matrices technique, but it is difficult to analyze the system stability. In [28], a model-free adaptive control (MFAC) algorithm in [29] was extended to nonlinear systems with data dropouts. However, only the data dropouts in the feedback channel were considered. In [30], to simultaneously compensate for random network-induced delays and packet dropouts in both the feedback and forward channels, a data-based networked predictive control (DBNPC) method was proposed for networked nonlinear systems. Nevertheless, the above drawback (iii) still remains unsolved, which motivates the present study.

This brief presents a new data-based networked control method for a class of nonlinear systems, where random network-induced delays and packet dropouts in both the feedback and forward channels are considered simultaneously. The round-trip time (RTT) delay is redefined so as to describe the total effect of the two-channel network-induced delays and packet dropouts. To compensate for the random RTT delays, an input design-based compensation control (IDBCC) method is proposed, where a simple yet effective compensation strategy is presented based on the control input design. Then an explicit sufficient condition is established to guarantee the closed-loop

This work was supported in part by the National Natural Science Foundation of China under Grants 61273104, 61333003, 61210012, 61490701, 61290324 and 61673023, in part by the Beijing Municipal Natural Science Foundation under grant 4152014, in part by the Outstanding Young Scientist Award Foundation of Shandong Province of China under grant BS2013DX015, in part by the Great Wall Scholar Candidate Training Program of North China University of Technology (NCUT), in part by the Scientific Research Foundation of NCUT, and in part by the Research Fund for the Taishan Scholar Project of Shandong Province of China.

Z.-H. Pang and D. Sun are with the Key Laboratory of Fieldbus Technology and Automation of Beijing, North China University of Technology, Beijing 100144, China (e-mail: zhonghua.pang@ia.ac.cn; sundehui@ncut.edu.cn).

G.-P. Liu is with the School of Engineering, University of South Wales, Pontypridd CF37 1DL, U.K., and is also with the CTGT Center, Harbin Institute of Technology, Harbin 150001, China (e-mail: guoping.liu@southwales.ac.uk).

D. Zhou is with the College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China, and is also with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: zdh@mail.tsinghua.edu.cn).

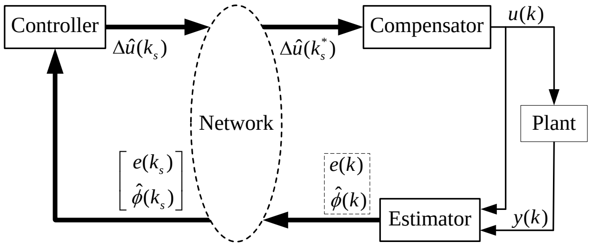


Fig. 1. IDBCC scheme.

stability and the output error convergence.

Compared with the existing DBC methods for NCSs in [27]-[30], the main advantages of the proposed IDBCC method include: (i) Like the DBNPC method in [30], the IDBCC method is proposed to solve the DBC problem of a class of nonlinear systems with random network-induced delays and packet dropouts, which was not considered in [27]-[29]; (ii) Only one control command needs to be transmitted to the actuator through network, which thus leads to less burden over network traffic; (iii) The proposed compensation strategy can be flexibly designed by using various methods so that the DBNPC method in [30] is just a special case of the IDBCC method, and thus, a better control performance can be expected.

*Notation:* The notation used here is fairly standard. $\Delta x(k)$ is defined as $\Delta x(k) = x(k) - x(k-1)$. $|x|$ means the absolute value of the scalar $x$. $\text{sign}(\cdot)$ denotes the signum function.

## II. IDBCC Scheme

Consider the NCS setup shown in Fig. 1, which includes five parts: a physical plant, an estimator in the sensor, a communication network, a controller, and a compensator in the actuator. Each part will be described in the following. It is assumed that the sensor and actuator are time-driven and synchronous, whereas the controller is event-driven.

The physical plant is a single-input single-output nonlinear system described by

$$y(k+1) = f\big(y(k), \cdots, y(k-n_y), u(k), \cdots, u(k-n_u)\big), \quad (1)$$

where $y(k)$ and $u(k)$ are the output and input of the plant, respectively, $f(\cdot)$ is an unknown nonlinear function, and $n_y$ and $n_u$ are unknown output and input orders. The following two assumptions are made for the controlled plant:

*Assumption 1:* The partial derivative of $f(\cdot)$ with respect to $u(k)$ is continuous.

*Assumption 2:* System (1) is generalized Lipschitz, i.e., $|\Delta y(k+1)| \leq \bar{\phi}|\Delta u(k)|$ for any $k$ and $\Delta u(k) \neq 0$, where $\bar{\phi}$ is a positive constant.

If Assumptions 1 and 2 are satisfied, according to [19], system (1) can be transformed into the following equivalent dynamic linearization data model

$$y(k+1) = y(k) + \phi(k)\Delta u(k), \quad (2)$$

where $|\phi(k)| \leq \bar{\phi}$. It is assumed that $\phi(k)$ satisfies $\phi(k) > 0$ (or $\phi(k) < 0$) for all time $k$ [23].

In general, the parameter $\phi(k)$ is time-varying and unknown for nonlinear system (1). To estimate $\phi(k)$ online, a parameter estimator is designed in the sensor (see Fig. 1), and the following estimation algorithm in [19] is adopted:

$$\hat{\phi}(k) = \hat{\phi}(k-1) + \frac{\Delta u(k-1)}{\mu + \Delta u(k-1)^2}\big(\Delta y(k) - \hat{\phi}(k-1)\Delta u(k-1)\big), \quad (3)$$

$$\hat{\phi}(k) = \hat{\phi}(0), \quad \text{if } |\hat{\phi}(k)| \leq \varepsilon, \text{or } |\Delta u(k-1)| \leq \varepsilon, \quad \text{or } \text{sign}\big(\hat{\phi}(k)\big) \neq \text{sign}\big(\hat{\phi}(0)\big), \quad (4)$$

where $\hat{\phi}(k)$ is the estimate of $\phi(k)$ with the initial value $\hat{\phi}(0)$, $\mu > 0$ is the estimation weighting factor, and $\varepsilon$ is a small positive constant. The sensor sends the data $[e(k) \;\; \hat{\phi}(k)]^T$ and the timestamp $k$ in one packet to the controller with

$$e(k) = r(k) - y(k), \quad (5)$$

where $r(k)$ is the reference input.

For the NCS depicted in Fig. 1, data packets travel through the network from the sensor to the controller and from the controller to the actuator, respectively. As a result, network-induced delays and packet dropouts are inevitable during the packet transmission, which are usually random with unknown distribution. Since the controller is event-driven, it calculates a control command only when receiving a feedback packet, of which the the timestamp is supposed to be $k_s \leq k$.

Our goal is to design a control scheme to drive the system output $y(k)$ to track the reference input $r(k)$. To generate control commands in the controller, the following performance index is adopted:

$$J = \big(r(k_s+1) - y(k_s+1)\big)^2 + \lambda\Delta u(k_s)^2. \quad (6)$$

By substituting (2) into (6) and then minimizing (6), we have

$$\Delta u(k_s) = \frac{\phi(k_s)}{\lambda + \phi(k_s)^2}\big(r(k_s+1) - y(k_s)\big), \quad (7)$$

where $\lambda > 0$ is the control weighting factor.

In practical applications, the reference input $r(k+1)$ is not always known in advance. Therefore, in this brief, the following modified version of control law (7) is used to calculate the control increment:

$$\Delta\hat{u}(k_s) = \frac{\hat{\phi}(k_s)}{\lambda + \hat{\phi}(k_s)^2}\big(r(k_s) - y(k_s)\big). \quad (8)$$

Then the controller transmits it together with the timestamp $k_s$ to the actuator through the forward channel.

*Remark 1:* For the constant reference input $r(k_s)$, it can be obtained from (2) and (7) that

$$\Delta u(k_s+1) = \frac{\lambda\phi(k_s+1)}{\big(\lambda + \phi(k_s+1)^2\big)\phi(k_s)}\Delta u(k_s). \quad (9)$$

Equation (9) indicates that, with $\phi(k_s) > 0$, $\phi(k_s+1) > 0$, and $\lambda > 0$, we can obtain that $\text{sign}\big(\Delta u(k_s+1)\big) = \text{sign}\big(\Delta u(k_s)\big)$, and further $|\Delta u(k_s+1)| < |\Delta u(k_s)|$ if $\phi(k_s+1) \approx \phi(k_s)$, which will inspire us in the following to design a compensator in the actuator.

In the actuator, a compensator is designed (see Fig. 1) to (i) buffer the received packets and only store the latest packet through the comparison of timestamps, and (ii) to compensate for random network-induced delays and packet dropouts based on the latest packet. For our purpose, the effect of the network-induced delays and packet dropouts in both channels is described by redefining the RTT delay $\tau_k$ as

$$\tau_k = k - k_s^*, \tag{10}$$

where $k_s^*$ is the timestamp of the latest packet with $k_s^* \leq k_s \leq k$. Thus, the latest control increment available in the actuator can be expressed as $\Delta\hat{u}(k_s^*)$.

In view of the definition in (10), it is obvious that the RTT delay satisfies $\tau_{k+1} \leq \tau_k + 1$. It is assumed that the RTT delay $\tau_k$ is bounded by $\bar{\tau}$, i.e., $\tau_k \leq \bar{\tau}$ for all $k$, which means that at least one packet can arrive at the actuator during $\bar{\tau}$ sampling periods. According to the fact mentioned in Remark 1, a network delay compensation strategy based on the control input design is presented as follows:

$$\Delta u(k) = \beta_k^{\tau_k} \Delta\hat{u}(k_s^*), \tag{11}$$

where $\beta_k > 0$ is the compensation factor. Then, to compensate for the RTT delay $\tau_k$, the following control signal is applied to system (1) at time $k$:

$$u(k) = u(k-1) + \Delta u(k). \tag{12}$$

*Remark 2:* It is noted that the above design procedure of the IDBCC scheme only involves input and output data of the controlled plant. Neither the dynamic model nor the structure information of the controlled plant is required. Therefore, like the DBNPC method in [30], the IDBCC method is also a DBC method for the addressed networked nonlinear system. However, different from the DBNPC method, the IDBCC method only needs to transmit one control command to the actuator through the forward channel, which thus overcomes the aforementioned three drawbacks of MBNPC methods. In addition, it should be pointed out that there is an error in the description for the design of the DBNPC scheme in [30]. In fact, the control increment $\Delta u(k)$ in (16) of [30] is applied to the plant, and thus the equation (15) in [30] for calculating the control signal $u(k)$ should be $u(k) = u(k-1) + \Delta u(k|k_s^*)$, where $\Delta u(k|k_s^*)$ is generated by (11) in [30].

## III. Stability Analysis

This section is concerned with the stability analysis of the resulting closed-loop system. Without loss of generality, we assume $\phi(k) > 0$ in this brief. Thus, it is clear from (4) that $\hat{\phi}(k) > \varepsilon > 0$.

Before proceeding, the following lemma is first presented.

*Lemma 1 [30]:* Consider the following discrete-time scalar linear system:

$$\begin{aligned} x(k+1) &= x(k) - a(k)x(k-\tau_k), \\ x(k) &= \psi(k), k = -\bar{\tau}, -\bar{\tau}+1, \cdots, 0, \end{aligned} \tag{13}$$

where $x(k)$ is the scalar state, $a(k)$ is the time-varying parameter, and $\psi(k)$ is the initial condition. System (13) is stable if $0 < a(k) < 2/(2\bar{\tau}+1)$.

Next, Lemma 1 will be used to derive the condition for the stability and convergence of the closed-loop IDBCC system.

*Theorem 1:* If $\beta_k$ is chosen as

$$\beta_k < \sqrt[\tau_k]{\frac{2\big(\lambda + \hat{\phi}(k-\tau_k)^2\big)}{(2\bar{\tau}+1)\bar{\phi}\hat{\phi}(k-\tau_k)}}, \tag{14}$$

the closed-loop IDBCC system is not only stable but also guarantees a zero steady-state output error for a constant reference input.

*Proof:* From (8), (10), and (11), it is obtained that

$$\Delta u(k) = \beta_k^{\tau_k} \frac{\hat{\phi}(k-\tau_k)}{\lambda + \hat{\phi}(k-\tau_k)^2} e(k-\tau_k). \tag{15}$$

Thus, from (2), (5), and (15), we obtain the following closed-loop system:

$$\begin{aligned} e(k+1) &= e(k) - \Delta y(k+1) \\ &= e(k) - \phi(k)\Delta u(k) \\ &= e(k) - \alpha(k)e(k-\tau_k), \end{aligned} \tag{16}$$

where

$$\alpha(k) = \phi(k)\beta_k^{\tau_k} \frac{\hat{\phi}(k-\tau_k)}{\lambda + \hat{\phi}(k-\tau_k)^2}.$$

With $0 < \phi(k) \leq \bar{\phi}$, $\beta_k > 0$, $\lambda > 0$, and $\hat{\phi}(k-\tau_k) > 0$, we have

$$0 < \alpha(k) \leq \bar{\phi}\beta_k^{\tau_k} \frac{\hat{\phi}(k-\tau_k)}{\lambda + \hat{\phi}(k-\tau_k)^2}. \tag{17}$$

Then, according to Lemma 1, it is clear that system (16) is stable if

$$0 < \alpha(k) \leq \bar{\phi}\beta_k^{\tau_k} \frac{\hat{\phi}(k-\tau_k)}{\lambda + \hat{\phi}(k-\tau_k)^2} < \frac{2}{2\bar{\tau}+1}. \tag{18}$$

That is,

$$\beta_k < \sqrt[\tau_k]{\frac{2\big(\lambda + \hat{\phi}(k-\tau_k)^2\big)}{(2\bar{\tau}+1)\bar{\phi}\hat{\phi}(k-\tau_k)}}. \tag{19}$$

Moreover, it is easy to observe from (16) that a zero steady-state output error can be achieved for the constant reference input. The proof is completed. ■

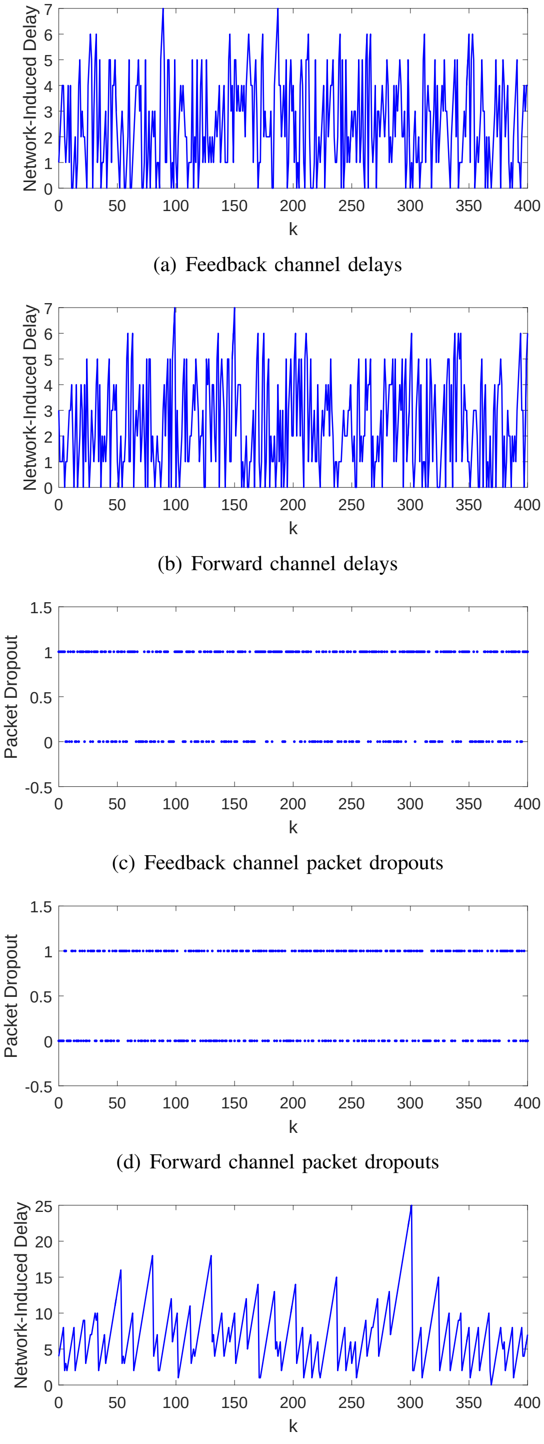
*Remark 3:* It is important to note that from the comparison of the closed-loop forms of the DBNPC method in [30] and the IDBCC method, i.e., (28) in [30] and (16) in this brief, it is easy to find that, when future reference signals cannot be known beforehand and are set to be $r(k+\bar{\tau}+1) = \cdots = r(k+1) = r(k)$, the DBNPC method is just a special case of the IDBCC method. Thus in this case, by carefully designing the compensation factor, the IDBCC method can provide a better control performance than the DBNPC method.

## IV. Simulation Study

In this section, a numerical example is given to illustrate the effectiveness of the IDBCC method for networked nonlinear systems. The following nonlinear plant is considered:

$$\begin{aligned} y(k) =& \frac{y(k-1)y(k-2)y(k-3)u(k-2)\big(y(k-3)-1\big)}{1 + y(k-2)^2 + y(k-3)^2} \\ &+ \frac{2.5u(k-1) + 0.5u(k-3)^2}{1 + y(k-2)^2 + y(k-3)^2}. \end{aligned} \tag{20}$$

Suppose that the feedback and forward channels are subject to random network-induced delays and packet dropouts, as shown in Figs. 2(a)-2(d). The two-channel network-induced delays vary from 0 to 7 steps, and the packet dropout rates of



(a) Feedback channel delays

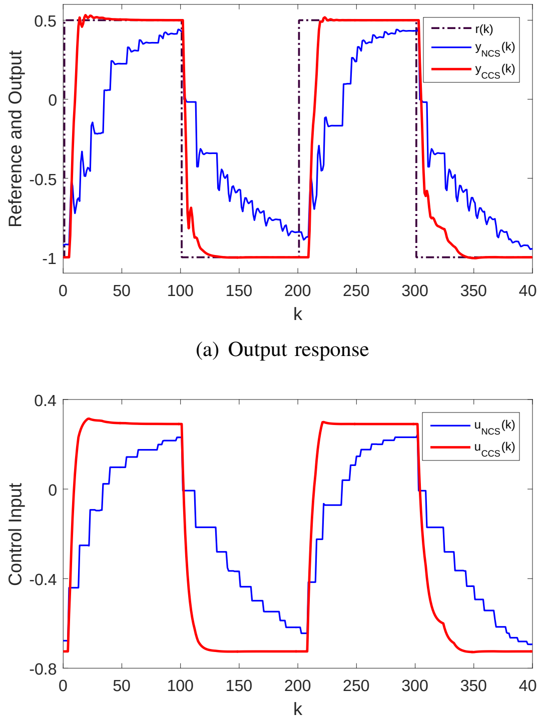
(b) Forward channel delays

(c) Feedback channel packet dropouts

(d) Forward channel packet dropouts

(e) RTT delays

Fig. 2. Network-induced delays and packet dropouts in two channels.



(a) Output response

(b) Control input

Fig. 3. Simulation results of IDBCC method.

the feedback and forward channels are $40.40\%$ and $46.88\%$, respectively, which lead to the random RTT delays shown in Fig. 2(e). In Figs. 2(c) and 2(d), $1$ and $0$ denote the success and failure of packet transmission, respectively.

Under the network-induced constraints shown in Fig. 2, the parameters are set to be $\mu = 1$, $\lambda = 5$, $\hat{\phi}(0) = 1$, and $\varepsilon = 10^{-5}$. The simulation results are shown in Fig. 3. It can be seen from Fig. 3(a) that without network delay compensation, the output performance of the NCS is very poor (thin blue line). The reason is that, due to the presence of random network-induced delays and packet dropouts, the control inputs applied to the plant are delayed control signals, i.e., $u(k) = u(k-\tau_k)$, as shown in Fig. 3(b) (thin blue line). When the proposed IDBCC method with $\beta_k = 0.8$ is applied to the plant, a much better system performance is obtained, as shown in Fig. 3(a) (thick red line). To quantitatively evaluate the system performance, a output error index $E = \sum_{k=0}^{400} |e(k)|$ is defined. It is obtained from Fig. 3(a) that $E_{\text{NCS}} = 177.4213$ and $E_{\text{IDBCCS}} = 54.1618$, which indicate that the proposed IDBCC method is effective.

In addition, for comparison, the performance of the DBNPC method in [30] is tested, where the future reference signals are set to be $r(k+\bar{\tau}+1) = \cdots = r(k+1) = r(k)$. It should be noted that, in this case, the DBNPC method is only a special case of the IDBCC method, and thus, the IDBCC method can obtain a better performance than the DBNPC method by carefully designing the compensation factor $\beta_k$. Here, to achieve as good a system performance as possible, the parameter of the DBNPC method is chosen as $\lambda = 10$. The simulation result is shown in Fig. 4, and the output error index is $E_{\text{DBNPCS}} = 51.9960$. It can be seen that the DBNPC method provides a similar performance to the IDBCC method with $\beta_k = 0.8$.

## V. Conclusion

This brief has investigated a data-based networked control method for a class of nonlinear systems subject to random network-induced delays and packet dropouts in both the

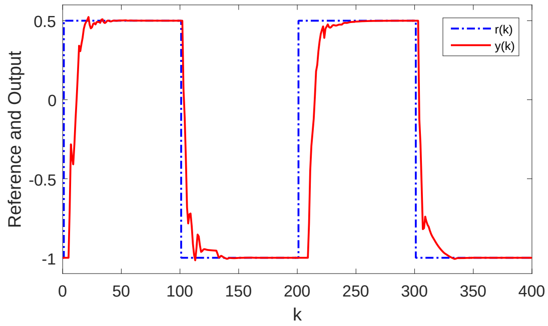


Fig. 4. Simulation result of DBNPC method in [30].

feedback and forward channels. An MFAC scheme has been employed to calculate the control command and transmit it to the actuator through network, and a compensator has been designed in the actuator to generate the control input applied to the plant based on the latest control command so that the two-channel network-induced delays and packet dropouts can be effectively compensated. Then an explicit sufficient condition has been presented for the stability of the closed-loop system. Finally, the simulation results have been given to demonstrate the effectiveness of the proposed method.

It is worth mentioning that, in [23], two novel MFAC schemes were proposed for a class of nonlinear systems. Different from the MFAC scheme used in this brief, they use the dynamic linearization approach not only on the controlled plant but also on the ideal controller. It is conjectured that the two MFAC schemes can also be extended to deal with the control problem of networked nonlinear systems addressed in this brief, although there would exist various challenging issues.

## References

[1] R. A. Gupta and M.-Y. Chow, “Networked control system: Overview and research trends,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 7, pp. 2527–2535, Jul. 2010.

[2] L. Zhang, H. Gao, and O. Kaynak, “Network-induced constraints in networked control systems–A survey,” *IEEE Trans. Ind. Inf.*, vol. 9, no. 1, pp. 403–416, Feb. 2013.

[3] Y. Q. Xia, Y. L. Gao, L. P. Yan, and M. Y. Fu, “Recent progress in networked control systems - A survey,” *Int. J. Autom. Comput.*, vol. 12, no. 4, pp. 343–367, Aug. 2015.

[4] G. P. Liu, “Predictive controller design of networked systems with communication delays and data loss,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 6, pp. 481–485, Jun. 2010.

[5] A. Ulusoy, O. Gurbuz, and A. Onat, “Wireless model-based predictive networked control system over cooperative wireless network,” *IEEE Trans. Ind. Inf.*, vol. 7, no. 1, pp. 41–51, Feb. 2011.

[6] A. Onat, T. Naskali, E. Parlakay, and O. Mutluer, “Control over imperfect networks: Model-based predictive networked control systems,” *IEEE Trans. Ind. Electron.*, vol. 58, no. 3, pp. 905–913, Mar. 2011

[7] B. Rahmani and A. H. Markazi, “Variable selective control method for networked control systems,” *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 3, pp. 975–982, Mar. 2013.

[8] Y. M. Liu and I. K. I-Kong Fong, “Robust predictive tracking control of networked control systems with time-varying delays and data dropouts,” *IET Control Theory Appl.*, vol. 7, no. 5, pp. 738–748, May 2013.

[9] Y. Shi, J. Huang, and B. Yu, “Robust tracking control of networked control systems: Application to a networked DC motor,” *IEEE Trans. Ind. Electron.*, vol. 60, no. 12, pp. 5864–5874, Dec. 2013.

[10] X. M. Sun, D. Wu, C. Wen, and W. Wang, “A novel stability analysis for networked predictive control systems,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 61, no. 6, pp. 453–457, Jun. 2014.

[11] Z. H. Pang, G. P. Liu, D. Zhou, and M. Chen, “Output tracking control for networked systems: A model-based prediction approach,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 9, pp. 4867–4877, Sep. 2014.

[12] H. Li, H. Yang, F. Sun, and Y. Xia, “Sliding-mode predictive control of networked control systems under a multiple-packet transmission policy,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6234–6243, Nov. 2014.

[13] Z. H. Pang, G. P. Liu, and D. Zhou, “Design and performance analysis of incremental networked predictive control systems,” *IEEE Trans. Cybern.*, vol. 46, no. 6, pp. 1400–1410, Jun. 2016.

[14] T. C. Yang, C. Peng, D. Yue, and M. R. Fei, “New study of controller design for networked control systems,” *IET Control Theory Appl.*, vol. 4, no. 7, pp. 1109–1121, Jul. 2010.

[15] G. Pin and T. Parisini, “Networked predictive control of uncertain constrained nonlinear systems: Recursive feasibility and input-to-state stability analysis,” *IEEE Trans. Autom. Control*, vol. 56, no. 1, pp. 72–87, Jan. 2011.

[16] H. Zhang, Y. Shi, and M. Liu, “$H_\infty$ step tracking control for networked discrete-time nonlinear systems with integral and predictive actions,” *IEEE Trans. Ind. Inf.*, vol. 9, no. 1, pp. 337–345, Feb. 2013.

[17] H. Li and Y. Shi, “Network-based predictive control for constrained nonlinear systems with two-channel packet dropouts,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 3, pp. 1574–1582, Mar. 2014.

[18] T. Wang, H. Gao, and J. Qiu, “A combined adaptive neural network and nonlinear model predictive control for multirate networked industrial process control,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 2, pp. 416–425, Feb. 2016.

[19] Z. Hou and Z. Wang, “From model-based control to data-driven control: Survey, classification and perspective,” *Inf. Sci.*, vol. 235, pp. 3–35, Jun. 2013.

[20] S. Yin, X. Li, H. Gao, and O. Kaynak, “Data-based techniques focused on modern industry: An overview,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 1, pp. 657–667, Jan. 2015.

[21] S. Formentin, P. De Filippi, M. Corno, M. Tanelli, and S. M. Savaresi, “Data-driven design of braking control systems,” *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 1, pp. 186-193, Jan. 2013.

[22] Z. Wang and D. Liu, “A data-based state feedback control method for a class of nonlinear systems,” *IEEE Trans. Ind. Inf.*, vol. 9, no. 4, pp. 2284-2292, Nov. 2013.

[23] Z. Hou and Y. Zhu, “Controller-dynamic-linearization-based model free adaptive control for discrete-time nonlinear systems,” *IEEE Trans. Ind. Inf.*, vol. 9, no. 4, pp. 2301–2309, Nov. 2013.

[24] Y. Zhu and Z. Hou, “Data-driven MFAC for a class of discrete-time nonlinear systems with RBFNN,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 25, no. 5, pp. 1013–1020, May 2014.

[25] M.-B. Rădac, R.-E. Precup, E. M. Petriu, and S. Preitl, “Iterative data-driven tuning of controllers for nonlinear systems with constraints,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6360–6368, Nov. 2014.

[26] Z. Wang and T. Chen, “Data and event-driven control of a class of networked non-linear control systems,” *IET Control Theory Appl.*, vol. 9, no. 7, pp. 1034–1041, Apr. 2015.

[27] Y. Xia, W. Xie, B. Liu, and X. Wang, “Data-driven predictive control for networked control systems,” *Inf. Sci.*, vol. 235, pp. 45-54, Jun. 2013.

[28] X. Bu, F. Yu, Z. Hou, and H. Zhang, “Model-free adaptive control algorithm with data dropout compensation,” *Math. Prob. Eng.*, vol. 2012, pp. 1–14, 2012.

[29] Z. Hou and X. Bu, “Model free adaptive control with data dropouts,” *Expert Syst. Appl.*, vol. 38, no. 8, pp. 10709–10717, Aug. 2011.

[30] Z. H. Pang, G. P. Liu, D. Zhou, and D. Sun, “Data-based predictive control for networked nonlinear systems with network-induced delay and packet dropout,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 2, pp. 1249–1256, Feb. 2016.