Letter

Comparison of Three Data-Driven Networked Predictive Control Methods for a Class of Nonlinear Systems

Zhong-Hua Pang, *Senior Member, IEEE*, Xue-Ying Zhao, Jian Sun, *Senior Member, IEEE*, Yuntao Shi, and Guo-Ping Liu, *Fellow, IEEE*

Dear Editor,

In this letter, in order to deal with random network delays and packet losses in a class of networked nonlinear systems, three data-driven networked predictive control methods are designed. Their closed-loop systems and control increments are derived, respectively. Although the expressions of their control increments are obviously different, they are similar in form and composition, which are helpful to evaluate the effects of control actions. A comparison of the control performance of the three methods is carried out by a simulation example so as to show their advantages and disadvantages.

Networked control systems (NCSs) have found practical applications in many fields such as smart microgrids and urban traffic [1], [2]. However, the utilization of networks inevitably brings communication problems such as network delays and packet losses, which would damage system performance. The past two decades have witnessed several typical control approaches to handle them, see, e.g., [3]–[5]. One of them, networked predictive control (NPC), can effectively compensate for those communication constraints, which fully utilize the packet-based transmission mechanism of networks.

Random network delays and packet losses in the backward and forward channels can be treated as destination-based lumped delays (DBLDs) defined in [6]. Up to date, there are two ways to deal with random DBLDs in NPC methods. One is to compensate for random round-trip time delays (RTTDs) [7], and the other is to separately handle one-way time delays (OWTDs) in the two channels [8]. For an NCS with an accurate model, the two ways can provide the same dynamic performance as that of the control system with no delays, although they have different closed-loop stability conditions.

In practice, however, it is not easy to create a precise model for a complex system, due to intrinsic properties such as nonlinearity, time-variance, and so on. In this case, a data-driven NPC approach is necessary. In [9] and [10], a data-driven NPC method was proposed to compensate for random RTTDs. According to delay compensation ways, another two data-driven NPC methods can also be designed. Thus, two questions arise: what are the differences between the three data-driven NPC methods, and which one is superior? The answers will be given by theoretical analysis and numerical simulation in the following, which are also the main contributions of this letter.

Three data-driven NPC methods: Consider a discrete-time nonlinear system:

$$y(t+1) = f\big(y(t),\ldots,y(t-n_y),u(t),\ldots,u(t-n_u)\big) \tag{1}$$

where $y(t)\in\mathbb{R}$ and $u(t)\in\mathbb{R}$ are the system output and input with unknown orders n_y and n_u, respectively, and $f(\cdot)$ is an unknown nonlinear function, of which the partial derivative with respect to $u(t)$ is continuous. It is assumed that system (1) satisfies $|\Delta y(t+1)| \le \bar{\psi}|\Delta u(t)|$ for any t and $\Delta u(t)\neq 0$, where $\bar{\psi}>0$ is a constant, and Δ is the difference operator, e.g., $\Delta u(t) = u(t)-u(t-1)$. Then, system (1) can be converted into the data model [11]

$$\Delta y(t+1) = \psi(t)\Delta u(t) \tag{2}$$

where $\psi(t)$ is a time-varying parameter with $|\psi(t)|\le\bar{\psi}$. To real-time estimate $\psi(t)$, the following recursive algorithm is used:

$$\hat{\psi}(t) = \hat{\psi}(t-1)+\gamma(t)(\Delta y(t)-\hat{\psi}(t-1)\Delta u(t-1)) \tag{3}$$

where $\hat{\psi}(t)$ is the estimate of $\psi(t)$, and $\gamma(t)\triangleq \Delta u(t-1)/(\mu+\Delta u(t-1)^2)$ with the weighting factor $\mu>0$.

The random DBLDs in the backward and forward channels of an NCS are denoted by $\tau_t^{\rm sc}$ and $\tau_t^{\rm ca}$ with upper bounds $\bar{\tau}^{\rm sc}$ and $\bar{\tau}^{\rm ca}$, respectively. In order to mitigate adverse effects of the two-channel random DBLDs on system performance, by using data model (2), three data-driven NPC methods are designed, which are called R-NPC1, R-NPC2, and O-NPC, respectively. The former two methods are based on RTTDs (see Fig. 1), and the last one is based on OWTDs (see Fig. 2). Obviously, the three methods consist of three same control components, i.e., a data buffer, a control predictor, and a delay compensator, which are assumed to be time-driven and synchronous. Although the three methods have similar structures, there are differences in the control schemes, which are given below.

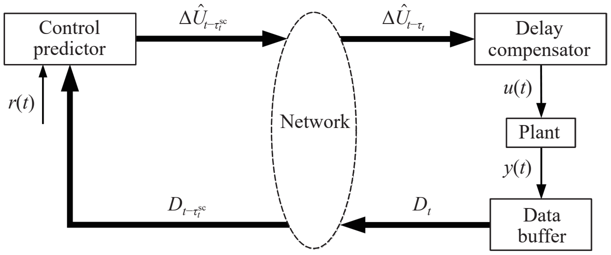

Fig. 1. R-NPC scheme.

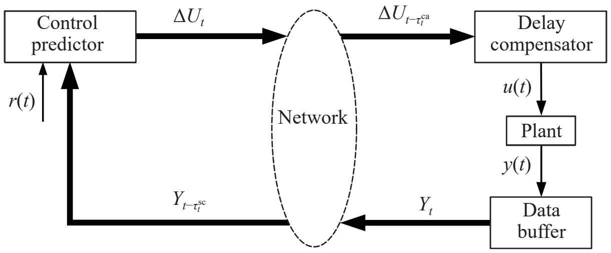

Fig. 2. O-NPC scheme.

1) Design of R-NPC1: As shown in Fig. 1, the data sequence $D_t = \{y(t),\ldots,y(t-\bar{\tau}^{\rm sc}),u(t-1),\ldots,u(t-\bar{\tau}^{\rm sc}-1),t\}$ is sent to the control predictor at each sampling instant t, where the last element of D_t, i.e., t, is the timestamp. In the control predictor, the latest data sequence at time t is $D_{t-\tau_t^{\rm sc}}$. By using $D_{t-\tau_t^{\rm sc}}$ and an incremental control law in [9], the following control increment predictions are generated:

$$\Delta\hat{u}(t-\tau_t^{\rm sc}+i|t-\tau_t^{\rm sc}) = a(t-\tau_t^{\rm sc})(r(t-\tau_t^{\rm sc}+1+i)-\hat{y}(t-\tau_t^{\rm sc}+i|t-\tau_t^{\rm sc})) \tag{4}$$

Corresponding author: Yuntao Shi.

Citation: Z.-H. Pang, X.-Y. Zhao, J. Sun, Y. T. Shi, and G.-P. Liu, "Comparison of three data-driven NPC methods for a class of nonlinear systems," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 9, pp. 1714–1716, Sept. 2022.

Z.-H. Pang, X.-Y. Zhao, and Y. T. Shi are with the Key Laboratory of Fieldbus Technology and Automation of Beijing, North China University of Technology, Beijing 100144, China (e-mail: shiyuntao@ncut.edu.cn; zxy612398@163.com; shiyuntao@ncut.edu.cn).

J. Sun is with the State Key Laboratory of Intelligent Control and Decision of Complex Systems, School of Automation, Beijing Institute of Technology, Beijing 100081, China (e-mail: sunjian@bit.edu.cn).

G.-P. Liu is with the Center for Control Science and Technology, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: liugp@sustech.edu.cn).

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

Digital Object Identifier 10.1109/JAS.2022.105830

for $i=0,1,2,\ldots,\bar{\tau}$, where $\bar{\tau}=\bar{\tau}^{\mathrm{sc}}+\bar{\tau}^{\mathrm{ca}}$ is the upper bound of RTTDs, $a(t-\tau_t^{\mathrm{sc}})\triangleq\hat{\psi}(t-\tau_t^{\mathrm{sc}})/(\lambda+\hat{\psi}(t-\tau_t^{\mathrm{sc}})^2)$, $\lambda>0$ is a control parameter, $r(\cdot)$ is the reference signal, $\hat{y}(t-\tau_t^{\mathrm{sc}}|t-\tau_t^{\mathrm{sc}})=y(t-\tau_t^{\mathrm{sc}})$, and

$$\begin{aligned}\hat{y}(t-\tau_t^{\mathrm{sc}}+j|t-\tau_t^{\mathrm{sc}})&=\hat{y}(t-\tau_t^{\mathrm{sc}}+j-1|t-\tau_t^{\mathrm{sc}})\\&\quad+\hat{\psi}(t-\tau_t^{\mathrm{sc}})\Delta\hat{u}(t-\tau_t^{\mathrm{sc}}+j-1|t-\tau_t^{\mathrm{sc}})\end{aligned}\tag{5}$$

for $j=1,2,\ldots,\bar{\tau}$. As a result, the control increment sequence $\Delta\hat{U}_{t-\tau_t^{\mathrm{sc}}}=\{\Delta\hat{u}(t-\tau_t^{\mathrm{sc}}|t-\tau_t^{\mathrm{sc}}),\Delta\hat{u}(t-\tau_t^{\mathrm{sc}}+1|t-\tau_t^{\mathrm{sc}}),\ldots,\Delta\hat{u}(t-\tau_t^{\mathrm{sc}}+\bar{\tau}|t-\tau_t^{\mathrm{sc}}),t-\tau_t^{\mathrm{sc}}\}$ is obtained, which is transmitted to the delay compensator. In the delay compensator, the real-time RTTD is

$$\tau_t=\tau_t^{\mathrm{ca}}+\tau_{t-\tau_t^{\mathrm{ca}}}^{\mathrm{sc}}\tag{6}$$

and the latest control increment sequence is $\Delta\hat{U}_{t-\tau_t}=\{\Delta\hat{u}(t-\tau_t|t-\tau_t),\Delta\hat{u}(t-\tau_t+1|t-\tau_t),\ldots,\Delta\hat{u}(t-\tau_t+\bar{\tau}|t-\tau_t),t-\tau_t\}$. To compensate for τ_t, the applied control signal is

$$u(t)=u(t-1)+\Delta\hat{U}_{t-\tau_t}\{\tau_t\}=u(t-1)+\Delta\hat{u}(t|t-\tau_t)\tag{7}$$

where $\Delta\hat{U}_{t-\tau_t}\{\tau_t\}$ denotes the $\{\tau_t+1\}$-th control action of $\Delta\hat{U}_{t-\tau_t}$.

2) Design of R-NPC2: For this method, the only difference from R-NPC1 lies in that the applied control signal is

$$u(t)=u(t-\tau_t-1)+\sum_{i=0}^{\tau_t}\Delta\hat{u}(t-\tau_t+i|t-\tau_t).\tag{8}$$

3) Design of O-NPC: As illustrated in Fig. 2, the data buffer transmits the data sequence $Y_t=\{y(t),y(t-1),\ldots,y(t-\bar{\tau}^{\mathrm{sc}}),t\}$ to the control predictor at each time instant t. In the control predictor at time t, the received output data sequence $Y_{t-\tau_t^{\mathrm{sc}}}$ and the historical control commands buffered here are employed to produce the following output predictions:

$$\begin{aligned}\hat{y}(t-\tau_t^{\mathrm{sc}}+j|t-\tau_t^{\mathrm{sc}})&=\hat{y}(t-\tau_t^{\mathrm{sc}}+j-1|t-\tau_t^{\mathrm{sc}})\\&\quad+\hat{\psi}(t-\tau_t^{\mathrm{sc}})\Delta u(t-\tau_t^{\mathrm{sc}}+j-1)\end{aligned}\tag{9}$$

for $j=1,2,\ldots,\tau_t^{\mathrm{sc}}+\bar{\tau}^{\mathrm{ca}}$. Then, the control increment at time $t+\bar{\tau}^{\mathrm{ca}}$ is obtained as

$$\Delta u(t+\bar{\tau}^{\mathrm{ca}})=a(t-\tau_t^{\mathrm{sc}})(r(t+\bar{\tau}^{\mathrm{ca}}+1)-\hat{y}(t+\bar{\tau}^{\mathrm{ca}}|t-\tau_t^{\mathrm{sc}})).\tag{10}$$

The control increment sequence $\Delta U_t=\{\Delta u(t),\Delta u(t+1),\ldots,\Delta u(t+\bar{\tau}^{\mathrm{ca}}),t\}$ is sent to the delay compensator. In the delay compensator, the latest control increment packet at time t is $\Delta U_{t-\tau_t^{\mathrm{ca}}}=\{\Delta u(t-\tau_t^{\mathrm{ca}}),\Delta u(t+1-\tau_t^{\mathrm{ca}}),\ldots,\Delta u(t+\bar{\tau}^{\mathrm{ca}}-\tau_t^{\mathrm{ca}}),t-\tau_t^{\mathrm{ca}}\}$, of which the $\{\tau_t^{\mathrm{ca}}+1\}$-th element is used, i.e.,

$$u(t)=u(t-1)+\Delta U_{t-\tau_t^{\mathrm{ca}}}\{\tau_t^{\mathrm{ca}}\}.\tag{11}$$

Remark 1: There are two main differences between R-NPC1/2 and O-NPC. One lies in the control predictors (see (4) and (10)). The former two methods provide a sequence of candidate control increment predictions, and which one will be applied in the delay compensator cannot be known in advance in the control predictor. Nevertheless, the O-NPC method only gives a single future control command, which will definitely be used in the delay compensator at time $t+\bar{\tau}^{\mathrm{ca}}$. The other difference refers to Remark below.

Closed-loop stability analysis: The stability of the three data-driven NPC methods is analyzed as follows.

1) Stability of R-NPC1: For a constant reference input $\bar{r}$, define

$$e(t)=\bar{r}-y(t)\tag{12}$$

as the output tracking error. According to [9], it is obtained that

$$\Delta u(t)=\Delta u(t|t-\tau_t)=a(t-\tau_t)\sigma(t-\tau_t)^{\tau_t}e(t-\tau_t)\tag{13}$$

$$e(t+1)=e(t)-\psi(t)a(t-\tau_t)\sigma(t-\tau_t)^{\tau_t}e(t-\tau_t)\tag{14}$$

where $\sigma(t-\tau_t)=\lambda/(\lambda+\hat{\psi}(t-\tau_t)^2)$.

Theorem 1 [9]: If $\lambda\geq(2\bar{\tau}+1)^2\bar{\psi}^2/16$, the closed-loop system for R-NPC1 is stable with $\lim_{t\to\infty}|e(t)|=0$.

2) Stability of R-NPC2: From (4) and (5), we have

$$\begin{aligned}e(t|t-\tau_t)&\triangleq\bar{r}-\hat{y}(t|t-\tau_t)\\&=\sigma(t-\tau_t)^{\tau_t}e(t-\tau_t).\end{aligned}\tag{15}$$

Substituting (15) into (4) yields

$$\Delta\hat{u}(t|t-\tau_t)=a(t-\tau_t)\sigma(t-\tau_t)^{\tau_t}e(t-\tau_t).\tag{16}$$

It is obtained from (8) that

$$\Delta u(t)=a(t-\tau_t)\sum_{i=0}^{\tau_t}\sigma(t-\tau_t)^{i}e(t-\tau_t)-\sum_{i=1}^{\tau_t}\Delta u(t-i).\tag{17}$$

From (2) and (12), we get

$$e(t+1)=e(t)-\psi(t)\Delta u(t).\tag{18}$$

Then, from (17) and (18), we obtain

$$X(t+1)=\Phi(t)X(t)\tag{19}$$

where

$$X(t)=\begin{bmatrix}E(t)\\\Delta U(t-1)\end{bmatrix},\quad\Phi(t)=\begin{bmatrix}A+B(t)D(\tau_t)&B(t)C(\tau_t)\\D(\tau_t)&C(\tau_t)\end{bmatrix}$$

$$E(t)=\begin{bmatrix}e(t)\\e(t-1)\\\vdots\\e(t-\bar{\tau})\end{bmatrix},\quad\Delta U(t)=\begin{bmatrix}\Delta u(t)\\\Delta u(t-1)\\\vdots\\\Delta u(t-\bar{\tau}+1)\end{bmatrix}$$

$$A=\begin{bmatrix}1&0_{1\times\bar{\tau}}\\I_{\bar{\tau}\times\bar{\tau}}&0_{\bar{\tau}\times1}\end{bmatrix},\quad B(t)=\begin{bmatrix}-\psi(t)&0_{1\times(\bar{\tau}-1)}\\\multicolumn{2}{c}{0_{\bar{\tau}\times\bar{\tau}}}\end{bmatrix}$$

$$C(\tau_t)=\begin{bmatrix}\xi(\tau_t)&0_{1\times(\bar{\tau}-\tau_t)}\\I_{(\bar{\tau}-1)\times(\bar{\tau}-1)}&0_{(\bar{\tau}-1)\times1}\end{bmatrix}$$

$$\xi(\tau_t)=\begin{bmatrix}\underbrace{-1,-1,\ldots,-1}_{\tau_t}\end{bmatrix}$$

$$D(\tau_t)=\begin{bmatrix}0_{1\times\tau_t}&d(\tau_t)&0_{1\times(\bar{\tau}-\tau_t)}\\\multicolumn{3}{c}{0_{\bar{\tau}\times(\bar{\tau}+1)}}\end{bmatrix}$$

$$d(\tau_t)=a(t-\tau_t)\sum_{i=0}^{\tau_t}\sigma(t-\tau_t)^{i}.$$

Thus, a result is obtained as follows.

Theorem 2: The closed-loop system for R-NPC2 is stable if and only if the matrix $\Phi(t)$ is stable, and also $\lim_{t\to\infty}|e(t)|=0$.

3) Stability of O-NPC: It is obtained from (9) that

$$e(t|t-\tau_t')=e(t-\tau_t')-\hat{\psi}(t-\tau_t')\sum_{i=1}^{\tau_t'}\Delta u(t-i)\tag{20}$$

where

$$\tau_t'\triangleq\bar{\tau}^{\mathrm{ca}}+\tau_{t-\bar{\tau}^{\mathrm{ca}}}^{\mathrm{sc}}.\tag{21}$$

From (10) and (20), we have

$$\Delta u(t)=a(t-\tau_t')e(t-\tau_t')-(1-\sigma(t-\tau_t'))\sum_{i=1}^{\tau_t'}\Delta u(t-i).\tag{22}$$

Combing (18) and (22), we have

$$X(t+1)=\Phi'(t)X(t)\tag{23}$$

where

$$\Phi'(t)=\begin{bmatrix}A+B(t)D'(\tau_t')&B(t)C'(\tau_t')\\D'(\tau_t')&C'(\tau_t')\end{bmatrix}$$

$$C'(\tau_t')=\begin{bmatrix}\xi'(\tau_t')&0_{1\times(\bar{\tau}-\tau_t')}\\I_{(\bar{\tau}-1)\times(\bar{\tau}-1)}&0_{(\bar{\tau}-1)\times1}\end{bmatrix}$$

$$\xi'(\tau_t')=(1-\sigma(t-\tau_t'))\begin{bmatrix}\underbrace{-1,-1,\ldots,-1}_{\tau_t'}\end{bmatrix}$$

$$D'(\tau_t')=\begin{bmatrix}0_{1\times\tau_t'}&a(t-\tau_t')&0_{1\times(\bar{\tau}-\tau_t')}\\\multicolumn{3}{c}{0_{(\bar{\tau}-1)\times(\bar{\tau}+1)}}\end{bmatrix}.$$

Similar to (19), we obtain the following result.

Theorem 3: The closed-loop system for O-NPC is stable if and only if $\Phi'(t)$ is stable, which also guarantees $\lim_{t\to\infty}|e(t)|=0$.

Remark 2: The other difference between R-NPC1/2 and O-NPC is in RTTDs. From (6) and (21), it can be seen that the RTTD of R-NPC1/2, τ_t, consists of the real-time DBLDs in both channels, while the RTTD of O-NPC, τ_t', is composed of the real-time DBLD in the backward channel and the upper bound of DBLDs in the forward channel. It is clear that $\tau_t'\geq\tau_t$, which would affect the performance of O-NPC if data model (2) cannot be accurately obtained.

Remark 3: From (13), (17), and (22), it is easy to see that the control increments of the three methods are similar in form and composition. That is, they are all combined with the actions of a delayed tracking error or/and historical control increments. Moreover, (13) is one part of the first term of (17), and if $\tau_t = \tau'_t$, the first and second terms of (22) are one part of the first and second terms of (17), respectively. Thus, without loss of generality, for a constant reference input $\bar{r} > 0$ and with zero initial system outputs and control increments, the control increments satisfy $\Delta u(t)_{\text{R-NPC2}} > \Delta u(t)_{\text{O-NPC}} > \Delta u(t)_{\text{R-NPC1}}$ at the start-up stage, and then the control increment in (17) decreases fastest as $y(t)$ tends to $\bar{r}$. As a result, the R-NPC2 method would have the fastest response time and the smallest overshoot, and thus is superior to the other two methods.

Simulation results: The following system is used for simulation:

$$\begin{aligned}&p(t) = 0.5u(t)^3 - 1.5u(t)^2 + 1.5u(t),\\ &y(t+1) = 0.6y(t) - 0.1y(t-1) + 1.2p(t) - 0.1p(t-1).\end{aligned} \tag{24}$$

The initial inputs and outputs are zero, and the parameters are chosen as $\mu = 1$, $\lambda = 10$, and $\hat{\psi}(0) = 1$. Two simulation cases are taken into account for comparison.

Case 1: Random DBLDs $\tau_t^{\text{sc}} \in [2,8]$ are considered. The output responses of the three methods are given in Fig. 3. Obviously, the tracking performance of R-NPC2 is best, and those of R-NPC1 and O-NPC are similar, which can be intuitively observed from (13), (17), and (22). That is, in this case, $\tau_t = \tau'_t = \tau_t^{\text{sc}}$, and with zero historical control increments, three start-up control increments satisfy $\Delta u(t)_{\text{R-NPC2}} > \Delta u(t)_{\text{O-NPC}} > \Delta u(t)_{\text{R-NPC1}}$, and then $\Delta u(t)_{\text{R-NPC2}}$ rapidly decreases as the system output approaches the reference signal, which thus yield the output responses in Fig. 3.

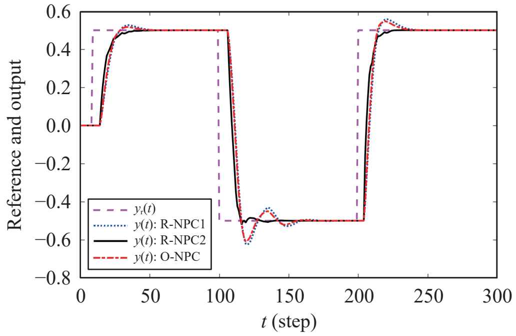

Fig. 3. Simulation results for backward channel DBLDs $\tau_t^{\text{sc}} \in [2,8]$.

Case 2: The same random DBLDs as in Case 1 are considered in the forward channel, i.e., $\tau_t^{\text{ca}} \in [2,8]$. The simulation results are illustrated in Fig. 4, indicating that the output responses of R-NPC1/2 are the same as those in Case 1, while the output response of O-NPC becomes much worse than that in Case 1. The reason is that, for this case, $\tau_t = \tau_t^{\text{ca}} \in [2,8]$ still holds, while $\tau'_t = \bar{\tau}^{\text{ca}} = 8$. Moreover, the inevitable errors between $\hat{\psi}(t)$ and $\psi(t)$ lead to that the performance of O-NPC becomes worse than that of R-NPC1 and even that of the networked control without compensation (green point line).

Conclusion: In this letter, three data-driven NPC methods have been presented for a class of nonlinear NCSs. In order to compare them, the control scheme design, closed-loop system analysis, and numerical simulation have been conducted. The main similarities and differences between them have been discussed. Simulation results have shown that among them, the R-NPC2 method can provide the best performance.

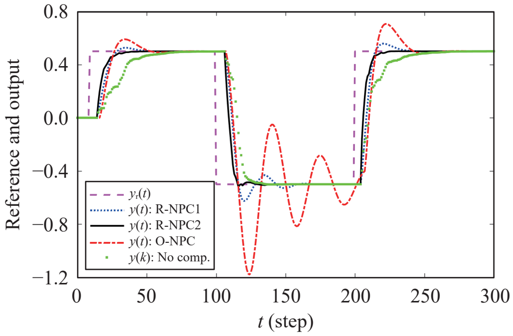

Fig. 4. Simulation results for forward channel DBLDs $\tau_t^{\text{ca}} \in [2,8]$.

Acknowledgments: This work was supported in part by the National Natural Science Foundation of China (62173002, 61925303, 62088101, U20B2073, 61720106011, 62173255), the National Key R&D Program of China (2018YFC0809700), and the Beijing Natural Science Foundation (4222045).

References

[1] X.-M. Zhang, Q.-L. Han, X. Ge, D. Ding, L. Ding, D. Yue, and C. Peng, "Networked control systems: A survey of trends and techniques," *IEEE/CAA J. Autom. Sinica.*, vol. 7, no. 1, pp. 1–17, 2020.

[2] Z.-H. Pang, L.-Z. Fan, Z. Dong, Q.-L. Han, and G.-P. Liu, "False data injection attacks against partial sensor measurements of networked control systems," *IEEE Trans. Circuits Syst. II Expr. Briefs*, vol. 69, no. 1, pp. 149–153, 2022.

[3] L. Zou, Z. Wang, Q.-L. Han, and D. Zhou, "Moving horizon estimation for networked time-delay systems under round-robin protocol," *IEEE Trans. Autom. Control*, vol. 64, no. 12, pp. 5191–5198, 2019.

[4] F. Xiao, Y. Shi, and T. Chen, "Robust stability of networked linear control systems with asynchronous continuous-time and discrete-time event-triggering schemes," *IEEE Trans. Autom. Control*, vol. 66, no. 2, pp. 932–939, 2021.

[5] Z.-H. Pang, C.-B. Zheng, C. Li, G.-P. Liu, and Q.-L. Han, "Cloud-based time-varying formation predictive control of multi-agent systems with random communication constraints and quantized signals," *IEEE Trans. Circuits Syst. II Expr. Briefs*, vol. 69, no. 3, pp. 1282–1286, 2022.

[6] Z. Pang, C. Bai, G. Liu, Q. Han, and X. Zhang, "A novel networked predictive control method for systems with random communication constraints," *J. Syst. Sci. Complex.*, vol. 34, no. 4, pp. 1364–1378, 2021.

[7] Z.-H. Pang, C.-G. Xia, W.-F. Zhai, G.-P. Liu, and Q.-L. Han, "Networked active fault-tolerant predictive control for systems with random communication constraints and actuator/sensor faults," *IEEE Trans. Circuits Syst. II Expr. Briefs*, vol. 69, no. 4, pp. 2166–2170, 2022.

[8] J. Zhang, Y. Lin, and P. Shi, "Output tracking control of networked control systems via delay compensation controllers," *Automatica*, vol. 57, pp. 85–92, 2015.

[9] 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. 1249–1257, 2016.

[10] H. Tan, Y. Wang, M. Wu, Z. Huang, and Z. Miao, "Distributed group coordination of multiagent systems in cloud computing systems using a model-free adaptive predictive control strategy," *IEEE Trans. Neural Networks Learn. Syst.*, 2021. DOI: 10.1109/TNNLS.2021.3053016

[11] Z. Hou and S. Xiong, "On model-free adaptive control and its stability analysis," *IEEE Trans. Autom. Control*, vol. 64, no. 11, pp. 4555–4569, 2019.