

Research Article

Event-Triggered-Based External Consensus Protocol of RBF-ARX-Model-Based Networked Multiagent Systems with Nonlinear Dynamics and Communication Delays

Qi Lei^{1,2} and Ying Luo¹

¹School of Information Science and Engineering, Central South University, Changsha 410083, China

²Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems, Wuhan 430074, China

Correspondence should be addressed to Qi Lei; leiqi@csu.edu.cn

Received 22 January 2021; Revised 2 March 2021; Accepted 20 March 2021; Published 5 April 2021

Academic Editor: Sundarapandian Vaidyanathan

Copyright © 2021 Qi Lei and Ying Luo. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An event-triggered-based external consensus control protocol with delay compensation for networked multiagent systems (NMASs) with nonlinear dynamics and network-induced delays is discussed in this paper. An RBF-ARX modelling method is adopted to approximate a nonlinear system. By utilizing the RBF-ARX model, the locally linearized time series model can be obtained to describe the behaviour of agents with nonlinear characteristics. An event-triggered control protocol with communication delay compensation is proposed to reduce the effects of network-induced delays and the times of the agent's control update and communication between others by applying the idea of a prediction strategy. The event-triggered mechanism for each agent only depends on the deviation between its own output and the delay-compensated output of neighboring agents. Then, a distributed event-triggered-based and delay-compensated external consensus control protocol is given. Under this proposed control protocol, both the consensus and stability of the system can be proved. Finally, a numerical simulation is performed to verify the availability of our results.

1. Introduction

Currently, multiagent systems generally exchange data with each other over communication networks because of the advantages of networks in terms of flexibility and cost, which has led to the emergence of networked multiagent systems (NMASs). NMASs have been widely applied in many fields [1–3], and the coordinative control has been attracting increasing interest. The problem of consensus plays a key role in the research of coordination of NMASs.

It is said that NMASs achieve consensus if agents reach an agreement under a communication network [4]. A consensus protocol can be divided into the following categories according to different requirements. The average consensus protocol [5] is one of the most common consensus protocols, and its consensus value is defined as the average of the initial value of every agent. The max-min consensus protocol [6] and the function consensus protocol

[7] are also dependent on the initial value of agents. These consensus values are calculated by the maximum (minimum) value of the initial value or by a function related to the initial value. However, these types of consensus protocols are limited to NMASs with a nonzero initial value [8].

Occasionally, we need the output of system tracking by inputting any selected external reference value. For example, an industrial process contains some subsystems that have their own goals, such as stable temperature and expected water level. These subsystems should be set to given external references. Therefore, the consensus value is independent of the initial value and is only determined by external reference inputs given to systems. A consensus protocol like this is called an external consensus protocol. A detailed discussion of the external consensus protocol can be found in [9, 10]. From the abovementioned definition, the external consensus protocol of NMASs has widespread applications, such as in industrial fields.

In recent years, an increasing number of related research studies about consensus problems on NMASs have been conducted since it was proposed [11–15]. For practical systems, two unavoidable problems must be considered. Ji et al. and Qu et al. [16, 17] proposed several graphical necessary conditions for controllability of NMASs. When applying communication network in a practical system, the influence of network-induced delays is inevitable [18]. The existence of network-induced delays possibly leads to degradation in system performance and, at its worst, lead to destabilization [19–21]. Although various methods have been developed to overcome the effects of delays [22, 23], some strict assumptions generally have been made on the considered systems [24]. Meanwhile, it has been proved by [24–26] that the prediction strategy can not only compensate for the network-induced delays effectively but also in an active way that predicts the control sequences and sends them to the network. In [25], by considering the communication delays, the stability and output consensus can be reached for the underlying NMASs using a networked multiagent predictive control scheme. In [26], under the background of cloud computing, to deal with the impression of communication delays in the NMASs, a predictive control method was considered in order, and its stability and consensus on the NMASs with network-induced delays were analyzed.

The other problem is the nonlinear characteristics of an agent in practical applications. In practical systems, each agent itself is almost a nonlinear system or has nonlinear characteristic; thus, the dynamic model of the multiagent system is often complex, such as a temperature control system and a three-tank water system. Then it is necessary to consider agents with nonlinear dynamics instead of linear dynamics. The point to address nonlinear characteristics is how to build a reliable nonlinear system model. A Neural network (NN) is commonly applied due to difficulties in modelling nonlinear dynamics. Shi et al. and Xu et al. [27, 28] introduced an adaptive NN method that could obtain the unknown parameters of the system. However, asymptotic stability cannot be achieved because the adaptive NN cannot be asymptotically close to the nonlinear term. The BP algorithm has a strong nonlinear mapping ability and flexible structure [29], but its learning speed is extremely slow and inefficient.

Radial basis function (RBF) NN is another NN method and has the ability of local approximation and can be used for function learning. It has the advantages of quick speed and simple network structure. The state-dependent AutoRegressive model with eXogenous variable (ARX) is a time sequence analysis method and can be expressed using a linear regression equations form, which is simple to calculate [30]. The RBF-ARX model is a model structure that combines the NN and ARX models [31, 32]. In this modelling approach, the coefficients of an ARX model are approximated by using a series of RBF NN. The advantage of this approach is that a nonlinear system can be locally linearized and each model is an ARX model. Thus, it is easy to design the predictive control scheme using the RBF-ARX model [33].

It is known that a periodic control scheme is used among the aforementioned results. However, if the system works in the ideal conditions or has gradually stabilized, the agent's control update and communication between others at every sampling time will definitely cause unnecessary resource loss, such as energy, computational cost, communication cost, and so on [34]. This is another problem that needs to be addressed. To further reduce the times of the agent's control update and communication, an event-triggered mechanism has been developed [35, 36]. The event-triggered mechanism shows that the agent's communication and control update between others proceed only when the predefined event is satisfied; if the trigger condition is not satisfied, the controller does nothing and the information does not need to be transmitted between neighboring agents. Thus, the times and cost of the agent's operations can be reduced greatly. The event-triggered method was first applied in multiagent systems in [37]. Since then, many results of event-triggered-based NMASs have been reported. For example, considering the consensus problem of NMASs with fixed/switching topologies, Wu et al. [38] proposed a distributed event-triggered control protocol, Sun et al. [39] proposed a leader-following consensus of the NMASs under the fixed/switching topologies, and Jia and Tang [40] proposed an event-triggered control protocol in an NMAS where a nonlinear-function-type couple among neighboring agents must be considered. In an NMAS, an event-triggered mechanism can simplify calculations and decrease the wear of the actuator. However, a few research studies combine the prediction strategy and event-triggered mechanism in the multiagent system. The well-known ability of prediction strategies to overcome delay effects and event-triggered mechanisms to avoid unnecessary resource losses motivates us to write this paper.

From the abovementioned discussion, when the control system is designed for an NMAS in practical applications, not only network-induced delays but also nonlinear dynamics existing in agents should be considered. Furthermore, an event-triggered control scheme is another factor that should be considered to reduce the cost of the control system. To date, there are a few related studies on these aspects simultaneously, which motivate the current study. We study an event-triggered-based external consensus control protocol for NMASs with a nonlinear characteristic and network-induced delays. In the research, the nonlinear dynamics of the agent is modeled by using the RBF-ARX approach. To solve the problem resulted from network-induced delays, a prediction strategy is designed based on the constructed RBF-ARX model. An event triggering mechanism is designed to reduce actuator wear and calculation cost. To achieve the external consensus for the underlying system, a distributed event-triggered-based control protocol for NMASs with a nonlinear characteristic and network-induced delay is proposed.

The remainder of this paper is organized as follows. Section 2 presents some main knowledge about algebraic graph theory used in system analysis and shows the problems that we considered. Section 3 presents the key methods to address these problems. System stability and external

consensus are proven in section 4. Some numerical simulations to verify the effectiveness of our results are presented in section 5. A brief conclusion and our future works are presented in section 6.

2. Preliminaries and Problem Formulation

It is necessary to describe the NMASs with algebraic graph theory. In this paper, a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ is used to describe the communication topology structure of the NMASs. In the graph \mathcal{G} , $\mathcal{V} = \{1, \dots, n\}$ is the set of nodes in the NMASs. $\mathcal{E} = \{(i, j) \mid i, j = 1, \dots, n; i \neq j\} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges. $\mathcal{A} = [a_{ij}]_{n \times n}$ represents the adjacency matrix, where a_{ij} is used to describe whether two agents are connected and data can be transmitted. If there is a connection for data transmission between agent i and agent j , $a_{ij} > 0$; otherwise, $a_{ij} = 0$. An adjacency matrix is said to be unweighted if $a_{ij} = 1$. The set of neighbours of agent i is denoted by $N_i = \{j \mid (i, j) \in \mathcal{E}\}$. If $(i, j) \in \mathcal{E}$ is equivalent to $(j, i) \in \mathcal{E}$, the graph is undirected. Also, the undirected graph is connected if paths from every agent to every other agent exist. $\mathbf{L} = [l_{ij}]_{n \times n}$ is the Laplacian matrix of graph \mathcal{G} , where

$$l_{ij} = \begin{cases} \sum_{j=1, j \neq i}^n a_{ij}, & i = j, \\ -a_{ij}, & i \neq j. \end{cases} \quad (1)$$

For a connected and undirected graph, it is clear that the Laplacian matrix L is positive semidefinite that has no negative eigenvalue, i.e., $\mathbf{L} = \mathbf{L}^T \geq 0$. In this paper, NMAS with n nonlinear SISO agents are considered. Each agent has the following discrete-time nonlinear model formulation:

$$u_i(k) = \begin{cases} G_i(q^{-1}) \left(R(k) - y_i(k) - K_i \sum_{j \in N_i} (y_i(k-\tau) - y_j(k-\tau)) \right), & i \in \mathcal{F}, \\ -G_i(q^{-1}) \left(K_i \sum_{j \in N_i} (y_i(k-\tau) - y_j(k-\tau)) \right), & i \in \mathcal{Q}, \end{cases} \quad (4)$$

where \mathcal{F} is the chosen agents given external reference input and \mathcal{Q} is the agents without reference input. Also, the external reference input is represented by $R(k)$, and the positive gain is K_i . Then, $y_i(k-\tau)$ is the output of agent i at time $k-\tau$, and $y_j(k-\tau)$ is the output of neighbouring agent j at time $k-\tau$.

However, control protocol (4) is no longer suitable because of the network-induced delays. In addition, the uninterrupted agent's control update and data transmission between others will definitely cause unnecessary resource loss, and the nonlinear dynamics of the agent will cause difficulties in system design. To solve these problems, an RBF-ARX modelling approach is used to approximate agents' nonlinear dynamics. A prediction strategy based on the RBF-ARX model is adopted to compensate communication delay. An

$$\begin{aligned} y_i(k+1) &= f_i(y_i(k), \dots, y_i(k-n_{yi}), u_i(k), \dots, u_i(k-n_{ui})), \\ &\quad i = 1, 2, \dots, n, \end{aligned} \quad (2)$$

where $u_i(k)$ and $y_i(k)$ are the control input and output of agent i , respectively. n_{yi} and n_{ui} are orders. $f_i(\cdot)$ is a nonlinear function. Suppose nonlinear function $f_i(\cdot)$ satisfies the Lipschitz condition or there exists a Lipschitz constant ρ for any $x, y \in \mathbb{R}$ such that $\|f(x) - f(y)\| \leq \rho \|x - y\|$.

Moreover, local controller $i(i = 1, 2, \dots, n)$ of each agent makes the system achieve stability, which is designed as the following discrete-time model:

$$\begin{aligned} C_i(q^{-1})u_i(k+1) &= D_i(q^{-1})\tilde{e}_i(k), \\ C_i(q^{-1}) &= 1 + c_{i1}q^{-1} + \dots + c_{in_{ci}}q^{-n_{ci}}, \\ D_i(q^{-1}) &= d_{i0} + d_{i1}q^{-1} + \dots + d_{in_{di}}q^{-n_{di}}, \end{aligned} \quad (3)$$

where q^{-1} denotes the unit delay operator, $\tilde{e}_i(k)$ is the input signal of the local controller, n_{ci} and n_{di} are orders, and $c_{i1}, \dots, c_{in_{ci}}$ and $d_{i1}, \dots, d_{in_{di}}$ are coefficients of the discrete-time model (3). The main task of each agent's own controller is to ensure that the system can obtain stabilization without a network-induced delay. Let $G_i(q^{-1}) = D_i(q^{-1})/C_i(q^{-1})$.

The NMAS is said to reach consensus if $\lim_{k \rightarrow \infty} \|y_i(k) - y_j(k)\| = 0, i, j \in V$. Furthermore, the NMAS is said to achieve the external consensus if $\lim_{k \rightarrow \infty} \|y_i(k)\| = \lim_{k \rightarrow \infty} \|R(k)\|$, and the delay τ is a known positive integer.

The general external consensus control protocol with communication delay τ is introduced as follows from [8]:

event-triggered control scheme with the prediction strategy is used to reduce the times of control update and data transmission. Then, to achieve the external consensus of the NMASs, an event-triggered-based and delay-compensated external consensus control mechanism is proposed.

3. Event-Triggered-Based External Consensus Control Protocol

To address the nonlinear dynamics of agents, an RBF-ARX modelling approach is applied. This approach uses a cluster of RBF networks to approximate the coefficients of an ARX model. The common formulation of the RBF-ARX model can be described as

$$\begin{aligned}
y(k) &= \varphi_0(\mathbf{W}(k-1)) + \sum_{m=1}^{n_y} \varphi_{ym}(\mathbf{W}(k-1))y(k-m) + \sum_{m=0}^{n_u} \varphi_{um}(\mathbf{W}(k-1))u(k-m) + \xi(k), \\
\varphi_0(\mathbf{W}(k-1)) &= r_0^0 + \sum_{h=1}^{n_m} r_h^0 \exp\left\{-\rho_h^y \|\mathbf{W}(k-1) - \Phi_h^y\|_2^2\right\}, \\
\varphi_{ym}(\mathbf{W}(k-1)) &= r_{0m}^y + \sum_{h=1}^{n_m} r_{hm}^y \exp\left\{-\rho_h^y \|\mathbf{W}(k-1) - \Phi_h^y\|_2^2\right\}, \\
\varphi_{um}(\mathbf{W}(k-1)) &= r_{0m}^u + \sum_{h=1}^{n_m} r_{hm}^u \exp\left\{-\rho_h^u \|\mathbf{W}(k-1) - \Phi_h^u\|_2^2\right\}, \\
\mathbf{W}(k-1) &= \begin{bmatrix} \omega(k-1) \\ \omega(k-2) \\ \vdots \\ \omega(k-n_\omega) \end{bmatrix}, \\
\Phi_h^y &= \begin{bmatrix} \phi_{h1}^y \\ \phi_{h2}^y \\ \vdots \\ \phi_{hn_\omega}^y \end{bmatrix}, \\
\Phi_h^u &= \begin{bmatrix} \phi_{h1}^u \\ \phi_{h2}^u \\ \vdots \\ \phi_{hn_\omega}^u \end{bmatrix},
\end{aligned} \tag{5}$$

where n_u, n_y, n_m, n_ω represent the order of input, order of output, order of the RBF network, and order of state information, respectively; $\Phi_h^y, \Phi_h^u, h = 1, 2, \dots, n_m$ are RBF network centres; $\mathbf{W}(k-1)$ is the selected state information, which can be an information of control input, output, or both of them; $\rho_h^y, \rho_h^u, h = 1, 2, \dots, n_m$ are the scaling parameters; $r_h^0, h = 0, 1, \dots, n_m$, $r_{hm}^y, h = 0, 1, \dots, n_m; m = 1, 2, \dots, n_y$, and $r_{hm}^u, h = 0, 1, \dots, n_m; m = 0, 1, \dots, n_u$ are the linear weighting coefficients; and $\xi(k)$ represents the modelling deviation.

Remark 1. The RBF-ARX model is constructed as a global nonlinear model. The parameters of the RBF-ARX model

can be estimated by SNPOM [30] offline from process data, which can avoid some potential problems caused by the failure of online estimation.

Remark 2. The RBF-ARX model has a traditional autoregressive structure, which has advantages in system design and does well in combining prediction strategies. In addition, a locally linearized model can easily be acquired by fixing $\mathbf{W}(k-1)$ at any sampling time.

Then, based on the RBF-ARX model, a discrete-time nonlinear agent (2) can be approximated:

$$\begin{aligned}
A_{i,k}(q^{-1})y_i(k+1) &= \alpha_{i0,k} + B_{i,k}(q^{-1})u_i(k), \quad i = 1, 2, \dots, n, \\
A_{i,k}(q^{-1}) &= 1 + \alpha_{i1,k}q^{-1} + \dots + \alpha_{in_{yi},k}q^{-n_{yi}}, \\
B_{i,k}(q^{-1}) &= \beta_{i0,k} + \beta_{i1,k}q^{-1} + \dots + \beta_{in_{ui},k}q^{-n_{ui}}, \\
\alpha_{i0,k} &= r_{0,i}^0 + \sum_{h=1}^{n_m} r_{h,i}^0 \exp\left\{-\rho_{h,i}^y \|\mathbf{W}(k) - \Phi_{h,i}^y\|_2^2\right\}, \\
\alpha_{im,k} &= r_{0m,i}^y + \sum_{h=1}^{n_m} r_{hm,i}^y \exp\left\{-\rho_{h,i}^y \|\mathbf{W}(k) - \Phi_{h,i}^y\|_2^2\right\}, \quad m = 1, 2, \dots, n_{yi}, \\
\beta_{im,k} &= r_{0m,i}^u + \sum_{h=1}^{n_m} r_{hm,i}^u \exp\left\{-\rho_{h,i}^u \|\mathbf{W}(k) - \Phi_{h,i}^u\|_2^2\right\}, \quad m = 0, 1, \dots, n_{ui},
\end{aligned} \tag{6}$$

where $\alpha_{i0,k}, \alpha_{i1,k}, \dots, \alpha_{in_{yi},k}$ and $\beta_{i0,k}, \dots, \beta_{in_{ui},k}$ are coefficients of model (6); n_{yi} and n_{ui} are orders; and the other parameters have the same definition as in model (5) mentioned above, where the subscript i means the i -th agent.

The distributed event-triggered mechanism with network-induced delay compensation is designed.

$$\begin{aligned} \|e_i(k)\|_2^2 &> \sigma_{i,k} \|z_i(k)\|_2^2, \\ e_i(k) &= y_i(k_l^i) - y_i(k), \\ z_i(k) &= \sum_{j \in N_i} (y_i(k_l^i) - \bar{y}_j(k_{l'}^j)), \end{aligned} \quad (7)$$

where $e_i(k)$ is the deviation between agent i 's actual output at its triggering time and output at time k , k_l^i ($l = 0, 1, \dots$) represents the l -th triggering time of agent i , $k_{l'}^j$ is the last triggering time of agent j , $\bar{y}_j(k_{l'}^j)$ denotes the delay-compensated output of neighbouring agent j , and $\sigma_{i,k}$ represents a positive variable. How to calculate $\bar{y}_j(k_{l'}^j)$ and the selection of $\sigma_{i,k}$ are discussed later.

Remark 3. When the event-triggered control protocol (7) is designed, the delay-compensated output is considered; thus, it has the ability to acquire a more accurate triggering interval when a network-induced delay exists. The agent's control signal always updates at its own triggering time $k_l^i, k_{l+1}^i, k_{l+2}^i, \dots$.

With an event-triggered mechanism (7), an event is triggered only when the event condition is satisfied, $k = k_l^i$. At its triggering time, the control signal will update immediately of agent i and broadcast information of the output prediction to its neighbours, and then, the output deviation will reset to zero. Otherwise, no operations will be performed, and the control signal remains unchanged until the next triggering time. Therefore, the definition of the next triggering time k_{l+1}^i is shown by

$$k_{l+1}^i = \inf \left\{ |k| k > k_l^i, \|e_i(k)\|_2^2 > \sigma_{i,k} \|z_i(k)\|_2^2 \right\}. \quad (8)$$

Note that an agent has an unchanged event condition within a sampling period. When an event is triggered, the output deviation will reset to zero. It can avoid repetitive triggers. The triggering time interval is guaranteed within lower bound by T , which means that the triggering time is $nT, n = 1, 2, \dots$. Then, uninterrupted agent control updates and communication between others can be reduced. Here, T represents the sampling period.

On the other hand, it is of great importance to obtain accurate information of $\bar{y}_j(k_l^j)$. When an event is triggered, according to the RBF-ARX model (6), local controller model (3), and external consensus control protocol (4), an RBF-ARX-model-based prediction strategy is applied. This recursive prediction strategy is adopted to generate the agents' output prediction sequence from time $k - \tau + 1$ to k .

For simplicity, from (4), it is defined that

$$\begin{aligned} y_{ei}(k - \tau) &= K_i \sum_{j \in N_i} (y_j(k - \tau) - y_j(k_{l'}^j - \tau)), \\ y_{ri}(k - \tau) &= R(k - \tau) - y_i(k - \tau). \end{aligned} \quad (9)$$

The received information about neighbours is given by

$$\hat{Y}_j = [y_j(k_{l'}^j | k_{l'}^j - \tau), y_j(k_{l'}^j - 1 | k_{l'}^j - \tau), \dots, y_j(k_{l'}^j - \tau + 1 | k_{l'}^j - \tau)]^T, \quad j \in N_i. \quad (10)$$

From time $k - \tau$, the one-step-ahead output prediction of agent i can be described as

$$\begin{aligned} y_i(k - \tau + 1 | k - \tau) &= - \sum_{\eta=1}^{n_{yi}} \alpha_{i\eta, k-\tau} y_i(k - \eta - \tau + 1) + \beta_{i0, k-\tau} u_i(k - \tau | k - \tau) \\ &\quad + \sum_{\eta=1}^{n_{ui}} \beta_{i\eta, k-\tau} u_i(k - \eta - \tau). \end{aligned} \quad (11)$$

The one-step-ahead control signal can be described as

$$\begin{aligned} u_i(k - \tau + 1 | k - \tau) &= \begin{cases} -c_{i1} u_i(k - \tau | k - \tau) - \sum_{\eta=2}^{n_{ci}} c_{i\eta} u_i(k - \eta - \tau + 1) \\ + d_{i0} y_{ri}(k - \tau + 1 | k - \tau) + \sum_{\eta=1}^{n_{di}} d_{i\eta} y_{ri}(k - \eta - \tau + 1) \end{cases} \\ &= \begin{cases} -d_{i0} y_{ei}(k - \tau + 1 | k - \tau) - \sum_{\eta=1}^{n_{di}} d_{i\eta} y_{ei}(k - \eta - \tau + 1), & i \in \mathcal{F} \\ -c_{i1} u_i(k - \tau | k - \tau) - \sum_{\eta=2}^{n_{ci}} c_{i\eta} u_i(k - \eta - \tau + 1) \\ -d_{i0} y_{ei}(k - \tau + 1 | k - \tau) - \sum_{\eta=1}^{n_{di}} d_{i\eta} y_{ei}(k - \eta - \tau + 1), & i \in \mathcal{Q}. \end{cases} \end{aligned} \quad (12)$$

The control input and output predictions from time $k - \tau + 1$ to k can be calculated recursively according to the one-step-ahead prediction equation derivation (11) and (12). The

results of p -step-ahead prediction (13) and (14) are proposed here, $p = 1, 2, \dots, \tau$.

$$\begin{aligned} y_i(k - \tau + p | k - \tau) &= - \sum_{\eta=1}^{\min\{n_{yi}, p-1\}} \alpha_{i\eta, k-\tau} y_i(k - \eta - \tau + p | k - \tau) - \sum_{\eta=p}^{n_{yi}} \alpha_{i\eta, k-\tau} y_i(k - \eta - \tau + p) \\ &\quad + \sum_{\eta=0}^{\min\{n_{ui}, p-1\}} \beta_{i\eta, k-\tau} u_i(k - \eta - \tau - 1 + p | k - \tau) + \sum_{\eta=p}^{n_{ui}} \beta_{i\eta, k-\tau} u_i(k - \eta - \tau - 1 + p), \end{aligned} \quad (13)$$

$$u_i(k - \tau + p | k - \tau) = \left\{ \begin{array}{l} - \sum_{\eta=1}^{\min\{n_{ci}, p-1\}} c_{i\eta} u_i(k - \eta - \tau + p | k - \tau) - \sum_{\eta=p}^{n_{ci}} c_{i\eta} u_i(k - \eta - \tau + p) \\ + \sum_{\eta=0}^{\min\{n_{di}, p-1\}} d_{i\eta} y_{ri}(k - \eta - \tau + p | k - \tau) + \sum_{\eta=p}^{n_{di}} d_{i\eta} y_{ri}(k - \eta - \tau + p) \\ - \sum_{\eta=0}^{\min\{n_{de}, p-1\}} d_{i\eta} y_{ei}(k - \eta - \tau + p | k - \tau) + \sum_{\eta=p}^{n_{de}} d_{i\eta} y_{ei}(k - \eta - \tau + p), \quad i \in \mathcal{F} \\ - \sum_{\eta=1}^{\min\{n_{ci}, p-1\}} c_{i\eta} u_i(k - \eta - \tau + p | k - \tau) - \sum_{\eta=p}^{n_{ci}} c_{i\eta} u_i(k - \eta - \tau + p) \\ - \sum_{\eta=0}^{\min\{n_{de}, p-1\}} d_{i\eta} y_{ei}(k - \eta - \tau + p | k - \tau) + \sum_{\eta=p}^{n_{de}} d_{i\eta} y_{ei}(k - \eta - \tau + p), \quad i \in \mathcal{Q} \end{array} \right\}. \quad (14)$$

Then, at time k , the output prediction of agent i is chosen to be $\bar{y}_i(k) = y_i(k | k - \tau)$ to compensate for the delay τ . The saved output prediction sequence of agent i is represented as \hat{Y}_i , where

$$\hat{Y}_i = [y_i(k | k - \tau), y_i(k - 1 | k - \tau), \dots, y_i(k - \tau + 1 | k - \tau)]^T, \quad i = 1, 2, \dots, n. \quad (15)$$

Remark 4. The prediction functions (13) and (14) have two separate parts: one is calculated by the sequence of prediction information, and the other is calculated by the sequence of current information that is available at time $k - \tau$. These functions have a simple construction that is easy for computing. The prediction strategy is performed only when

the event is triggered, $k = k_l^i$. In this way, \hat{Y}_i will be sent through the network and become \hat{Y}_j received by the neighbouring agents.

Therefore, by introducing an event-triggered mechanism (7) into external consensus control protocol (4), an event-triggered-based and delay-compensated external consensus control protocol is proposed as (16), where $k \in [k_l^i, k_{l+1}^i]$. The network-induced delay occurs only when information is transmitted through communication networks; thus, the actual value of agent i at its triggering time k_l^i is effective here. Considering the control protocol (16) is the same as the time-based control scheme proposed in [8] if the event-triggered control protocol is not used.

$$u_i(k) = \begin{cases} G_i(q^{-1}) \left(R(k_l^i) - y_i(k_l^i) - K_i \sum_{j \in N_i} (y_i(k_l^i) - \bar{y}_j(k_{l'}^j)) \right), & i \in \mathcal{F}, \\ -G_i(q^{-1}) \left(K_i \sum_{j \in N_i} (y_i(k_l^i) - \bar{y}_j(k_{l'}^j)) \right), & i \in \mathcal{Q}. \end{cases} \quad (16)$$

Remark 5. From the abovementioned description, the effects of network-induced delay are compensated by using τ -step-ahead output prediction of agents instead of the information at time $t - \tau$. Control protocol (16) can achieve consensus of nonlinear multiagent systems due to the RBF-ARX-model-based prediction strategy.

Remark 6. For a connected and undirected graph, the NMASs can achieve consensus under the control protocol (16).

4. Stability and Consensus Analyses

The aforementioned control protocol (16) is discussed, and it is essential to analyze the external consensus and stability of the NMASs in this section. For simplicity, we denote $A_{i,k}(q^{-1})$ as $A_{i,k}$, $B_{i,k}(q^{-1})$ as $B_{i,k}$, and $G_i(q^{-1})$ as G_i . Then, the following theorem is established. Suppose agent $i = 1$ is the only one which is given the external reference input.

Theorem 1. For an NMAS with a nonlinear characteristic composed of agents with model (6), external consensus and stability can be guaranteed with control protocol (16) if $\sigma_{i,k}$ in (7) satisfies the following inequality:

$$0 < \sigma_{i,k}^2 < \frac{v_{i,k}^2(v_{i,k} + 1)}{(2v_{i,k} + 1)(\lambda_L l_L)^2}, \quad (17)$$

$$0 < \sigma_{i,k} \leq \sigma_k, 0 < v_{i,k} \leq v_k, \quad i = 1, 2, \dots, n,$$

where $l_L = \max\{l_{ii}\}$, l_{ii} is the diagonal element of the Laplacian matrix \mathbf{L} ; λ_L is the maximum eigenvalue of \mathbf{L} ; and $v_k, v_{i,k}, \sigma_k, \sigma_{i,k}$ are positive constants.

Proof. For $k \in [k_l^i, k_{l+1}^i]$, substituting control protocol (16) into model (6), there are

$$\begin{aligned} y_i(k+1) &= \tilde{A}_{i,k}y_i(k) + B_{i,k}u_i(k) + a_{i0,k} \\ &= \begin{cases} (\tilde{A}_{i,k} - B_{i,k}G_i)y_i(k) + B_{i,k}G_iR(k), \\ -B_{i,k}G_iK_i \sum_{j \in N_i} (y_i(k_l^i) - \bar{y}_j(k_l^j)) + a_{i0,k}, \quad i = 1, \\ \tilde{A}_{i,k}y_i(k) - B_{i,k}G_iK_i \sum_{j \in N_i} (y_i(k_l^i) - \bar{y}_j(k_l^j)) + a_{i0,k}, \\ i = 2, \dots, n, \end{cases} \end{aligned} \quad (18)$$

where $\tilde{A}_{i,k} = -\alpha_{i1,k} - \alpha_{i2,k}q^{-1} - \dots - \alpha_{in,k}q^{-(n_{y_i}-1)}$. The value of $\bar{y}_j(k_l^j)$ is able to be acquired from the received neighbours' information that is described in (10). The value of $y_j(k_l^j)$ can also be acquired from agent j at time k_l^j . Therefore, prediction deviation ε_j is available.

Note that when k is known, the parameters $\alpha_{i0,k}$, $\tilde{A}_{i,k}$, $B_{i,k}$, and $B_{i,k}G_iR(k) = H$, $i = 1$ can be calculated and treated as constants.

We introduce prediction deviation $\varepsilon_j = \bar{y}_j(k_l^j) - y_j(k_l^j)$, rewrite (18), and output deviation $e_i(k)$ in a matrix form.

$$\begin{aligned} y(k+1) &= \tilde{A}_k y(k) - \tilde{B}_k L y(k) - \tilde{B}_k L e(k) + \tilde{B}_k \hat{L} \varepsilon + H_k \\ &= (\tilde{A}_k - \tilde{B}_k L) y(k) - \tilde{B}_k L e(k) + \tilde{B}_k \hat{L} \varepsilon + H_k, \\ e(k+1) &= \mathbf{y}(k_l^i) - \mathbf{y}(k+1) \\ &= \mathbf{y}(k_l^i) - \mathbf{y}(k) + \mathbf{y}(k) - \mathbf{y}(k+1) \\ &= (\mathbf{I} - \tilde{A}_k + \tilde{B}_k L) \mathbf{y}(k) + (\mathbf{I} + \tilde{B}_k L) e(k) - \tilde{B}_k \hat{L} \varepsilon - H_k, \end{aligned} \quad (19)$$

where \mathbf{I} is the unit diagonal matrix,

$$\begin{aligned} y(k) &= \begin{bmatrix} y_1(k) \\ y_2(k) \\ \vdots \\ y_n(k) \end{bmatrix}, \\ e(k) &= \begin{bmatrix} e_1(k) \\ e_2(k) \\ \vdots \\ e_n(k) \end{bmatrix}, \\ H_k &= \begin{bmatrix} H + a_{10,k} \\ a_{20,k} \\ \vdots \\ a_{n0,k} \end{bmatrix}, \\ \varepsilon(k) &= \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}, \\ \tilde{A}_k &= \begin{bmatrix} \tilde{A}_{1,k} - B_{1,k}G_1 & & & \\ & \tilde{A}_{2,k} & & \\ & & \ddots & \\ & & & \tilde{A}_{n,k} \end{bmatrix}, \\ \tilde{B}_k &= \begin{bmatrix} B_{1,k}G_1K_1 & & & \\ & B_{2,k}G_2K_2 & & \\ & & \ddots & \\ & & & B_{n,k}G_nK_n \end{bmatrix}, \\ L &= \begin{bmatrix} l_{11} & l_{12} & \dots & l_{1n} \\ l_{21} & l_{22} & & \vdots \\ \vdots & & \ddots & \\ l_{n1} & \dots & & l_{nn} \end{bmatrix}, \\ \hat{L} &= \begin{bmatrix} 0 & -l_{12} & \dots & -l_{1n} \\ -l_{21} & 0 & & \vdots \\ \vdots & & \ddots & -l_{(n-1)n} \\ -l_{n1} & \dots & -l_{n(n-1)} & 0 \end{bmatrix}. \end{aligned} \quad (20)$$

Letting $w(k) = [\mathbf{y}^T(k), \mathbf{e}^T(k), \varepsilon^T(k), \mathbf{H}_k^T]^T$, the considered system can be described as follows:

$$w(k+1) = G_k w(k), \quad (21)$$

$$V(k) = w^T(k) P_k w(k). \quad (22)$$

where $G_k = \begin{bmatrix} \tilde{A}_k - \tilde{B}_k L & -\tilde{B}_k L & +\tilde{B}_k \tilde{L} & \mathbf{I} \\ \mathbf{I} - \tilde{A}_k + \tilde{B}_k L & \mathbf{I} - \tilde{A}_k + \tilde{B}_k L & -\tilde{B}_k \tilde{L} & -\mathbf{I} \\ & & \mathbf{I} & \\ & & & \mathbf{I} \end{bmatrix}$.

Then, the Lyapunov function can be constructed:

When the event-triggered mechanism (7) is considered, the forward difference of (22) is as follows:

$$\Delta V(k) = V(k+1) - V(k)$$

$$= w^T(k+1) P_k w(k+1) - w^T(k) P_k w(k)$$

$$< w^T(k) (G_k^T P_k G_k - P_k) w(k) + e^T(k) e(k) - \sigma_k z^T(k) z(k)$$

$$\leq w^T(k) (G_k^T P_k G_k - P_k) w(k) + w^T(k) \begin{bmatrix} \mathbf{0} & \sigma_k \lambda_L l_L \mathbf{I} \\ \mathbf{I} & \sigma_k \lambda_L l_L \mathbf{I} \\ \sigma_k \lambda_L l_L \mathbf{I} & \sigma_k \lambda_L l_L \mathbf{I} & \mathbf{0} \\ \mathbf{0} \end{bmatrix} w(k), \quad (23)$$

where $z(k) = [z_1(k), z_2(k), \dots, z_n(k)]^T = \mathbf{Ly}(k) + \mathbf{Le}(k) - \tilde{L}\xi$; \mathbf{L} is positive semidefinite, and $0\mathbf{I} \leq \mathbf{L} \leq \lambda_L \mathbf{I}$; $\tilde{L} = -(\mathbf{L} - \tilde{\mathbf{L}})$, $\tilde{\mathbf{L}} = \text{diag}(l_{ii})$, and it, thus, has $-\lambda_L \mathbf{I} \leq \tilde{L} \leq l_L \mathbf{I}$; $l_L = \max\{l_{ii}\}$; and $0 < \sigma_k < 1$. It can be concluded from (23) that if a positive

definite matrix P_k exists that makes $\Delta V(k) < 0$ at every triggering time, the considered system can be stable.

Let

$$G_k^T P_k G_k - P_k + \begin{bmatrix} \mathbf{0} & \sigma_k \lambda_L l_L \mathbf{I} \\ \mathbf{I} & \sigma_k \lambda_L l_L \mathbf{I} \\ \sigma_k \lambda_L l_L \mathbf{I} & \sigma_k \lambda_L l_L \mathbf{I} & \mathbf{0} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} -v_k \mathbf{I} & & & \\ & -v_k \mathbf{I} & & \\ & & -v_k \mathbf{I} & \\ & & & -v_k \mathbf{I} \end{bmatrix} < 0. \quad (24)$$

Performing a matrix elementary transformation in (25) yielded (26).

$$G_k^T P_k G_k - P_k = \begin{bmatrix} -v_k \mathbf{I} & -\sigma_k \lambda_L l_L \mathbf{I} \\ (-v_k - 1) \mathbf{I} & -\sigma_k \lambda_L l_L \mathbf{I} \\ -\sigma_k \lambda_L l_L \mathbf{I} & -\sigma_k \lambda_L l_L \mathbf{I} & -v_k \mathbf{I} \\ & & -v_k \mathbf{I} \end{bmatrix} < 0, \quad v_k > 0, \quad (25)$$

$$\begin{bmatrix} -v_k \mathbf{I} & -\sigma_k \lambda_L l_L \mathbf{I} \\ (-v_k - 1) \mathbf{I} & -\sigma_k \lambda_L l_L \mathbf{I} \\ -\sigma_k \lambda_L l_L \mathbf{I} & -\sigma_k \lambda_L l_L \mathbf{I} & -v_k \mathbf{I} \\ & & -v_k \mathbf{I} \end{bmatrix} \rightarrow \begin{bmatrix} -v_k \mathbf{I} & (-v_k - 1) \mathbf{I} \\ -\sigma_k \lambda_L l_L \mathbf{I} & -\sigma_k \lambda_L l_L \mathbf{I} & (-v_k^2(v_k + 1) + (2v_k + 1)(\lambda_L l_L)^2) \mathbf{I} \\ & & -v_k \mathbf{I} \end{bmatrix} < 0. \quad (26)$$

Therefore, the stability of the considered system can be proven if $-v_k^2(v_k + 1) + (2v_k + 1)(\lambda_L l_L)^2 < 0$ holds. That is,

$$0 < \sigma_k^2 < \frac{v_k(v_k + 1)}{(2v_k + 1)(\lambda_L l_L)^2}, \quad 0 < \sigma_k < 1, v_k > 0. \quad (27)$$

□

Remark 7. The introduction of σ_k and v_k leads to more flexibility in the design of event-triggered mechanisms. Proper parameters of $\sigma_{i,k}$ and $v_{i,k}$ can be selected according to the requirements, where $0 < \sigma_{i,k} \leq \sigma_k$ and $0 < v_{i,k} \leq v_k$.

Furthermore, when the system is stable, $\lim_{k \rightarrow \infty} (y_i(k+1) - y_i(k)) = 0$. Then, it leads to $\lim_{k \rightarrow \infty} (u_i(k+1) - u_i(k)) = 0$ and $\lim_{k \rightarrow \infty} \tilde{e}_i(k) = 0$. From equation (16), the external consensus of the system is that $\lim_{k \rightarrow \infty} R(k) = \lim_{k \rightarrow \infty} y_i(k) = \lim_{k \rightarrow \infty} \bar{y}_j(k) = \lim_{k \rightarrow \infty} y_j(k)$, which can be proven.

Thus, the proof of Theorem 1 has been accomplished.

5. Numerical Simulation

The stability and external consensus of the control protocol (16) proposed in Section 4 has been analyzed and proved. In this section, a comparison is also performed between our results (16) and those of the time-based control protocol proposed in [8].

Consider three different nonlinear agents with a discrete-time model as follows. The topology of the NMAS constructed by them is shown in Figure 1.

(i) Agent 1:

$$n_{u1} = 2;$$

$$n_{y1} = 3;$$

$$(r_{0,1}^0, r_{1,1}^0; r_{01,1}^y, r_{11,1}^y; r_{02,1}^y, r_{12,1}^y; r_{03,1}^y, r_{13,1}^y; r_{00,1}^u, r_{10,1}^u; r_{01,1}^u, r_{11,1}^u)$$

$$= (0.317, 2.587; 0.226, -0.682; 0.124, -0.442; 0.064, -0.081; 0.270, 2.406; 1.271, -0.525),$$

$$(\rho_{11}^y, \phi_{111}^y, \phi_{121}^y; \rho_{11}^u, \phi_{111}^u, \phi_{121}^u)$$

$$= (0.188, 2.500, 2.500; 0.188, 2.500, 2.500),$$

$$C_1(q^{-1}) = 1 - q^{-1},$$

$$D_1(q^{-1}) = 0.2 + 0.3q^{-1}.$$

For agent 2, we choose the order of input $n_{u2} = 1$, the order of output $n_{y2} = 2$, the order of RBF network n_m

$$\begin{aligned} y(k+1) = & \frac{2.5y(k)y(k-1)}{1 + y^2(k) + y^2(k-1)} \\ & + 0.7 \sin(0.5(y(k) + y(k-1))) \\ & + 1.2u(k) + 1.4u(k-1). \end{aligned} \quad (28)$$

(ii) Agent 2:

$$y(k+1) = \frac{y(k)}{1 + y(k)} + 0.2y(k-1) + u(k). \quad (29)$$

(iii) Agent 3:

$$\begin{aligned} y(k+1) = & \frac{5y(k)y(k-1)}{1 + y^2(k) + y^2(k-1) + y^2(k-2)} \\ & + u(k) + 1.1u(k-1). \end{aligned} \quad (30)$$

Next, by using the RBF-ARX modelling approach, the agents' RBF-ARX model is constructed similar to (6). The parameters of the RBF-ARX model are determined by the Akaike Information Criterion in this paper. We choose the order of input $n_{u1} = 2$, the order of output $n_{y1} = 3$, the order of RBF network $n_m = 1$, and the order of state information $n_\omega = 2$. $\mathbf{W}(k-1) = [y(k), y(k-1)]^T$ is selected as state information.

(i) Parameters of agent 1:

[

= 1, and the order of state information $n_\omega = 2$. $\mathbf{W}(k-1) = [y(k), y(k-1)]^T$ is selected as state information.

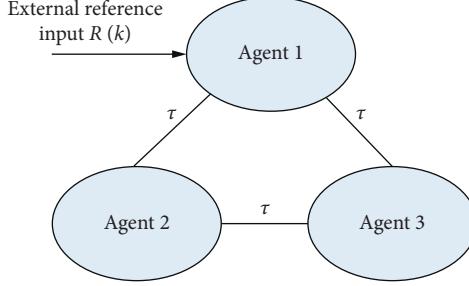


FIGURE 1: Topology structure of an NMAS.

(ii) Parameters of agent 2:

$$\begin{aligned}
n_{u2} &= 1; \\
n_{y2} &= 2; \\
(r_{0,2}^0, r_{1,2}^0; r_{01,2}^y, r_{11,2}^y; r_{02,2}^y, r_{12,2}^y; r_{00,2}^u, r_{10,2}^u) \\
&= (0.259, -0.369; 0.698, 0.993; \\
&\quad -0.035, -0.609; -0.449, -0.492), \\
(\rho_{1,2}^y, \phi_{11,2}^y, \phi_{12,2}^y; \rho_{1,2}^u, \phi_{11,2}^u, \phi_{12,2}^u) \\
&= (0.322, 2.475, 2.479; 0.339, 2.506, 2.502), \\
C_2(q^{-1}) &= 1 - q^{-1}, \\
D_2(q^{-1}) &= 0.1 - 0.09q^{-1}.
\end{aligned} \tag{32}$$

For agent 3, we choose the order of input $n_{u3} = 2$, the order of output $n_{y3} = 2$, the order of RBF network $n_m = 1$, and the order of state information $n_\omega = 2$. $\mathbf{W}(k-1) = [y(k), y(k-1)]^T$ is selected as state information.

(iii) Parameters of agent 3:

$$\begin{aligned}
n_{u3} &= 2; \\
n_{y3} &= 2; \\
(r_{0,3}^0, r_{1,3}^0; r_{01,3}^y, r_{11,3}^y; r_{02,3}^y, r_{12,3}^y; r_{00,3}^u, r_{10,3}^u; r_{01,3}^u, r_{11,3}^u) \\
&= (1.961, 2.396; 0.209, 0.038; 0.176, -0.543; \\
&\quad 1.556, -0.287; -0.906, 0.558), \\
(\rho_{1,3}^y, \phi_{11,3}^y, \phi_{12,3}^y; \rho_{1,3}^u, \phi_{11,3}^u, \phi_{12,3}^u) \\
&= (0.253, 2.071, 3.114; 0.314, 4.062, 2.872), \\
C_3(q^{-1}) &= 1 - q^{-1}, \\
D_3(q^{-1}) &= 0.15 + 0.15q^{-1}.
\end{aligned} \tag{33}$$

K_i in (16) is a positive gain that loosens or strengthens the connection between agents by tuning its value. A lower

or higher value of K_i will decide the loose or tight of the connection. The decision of triggering time depends on the variable $\sigma_{i,k}$, according to (7). It is clear that when lower values of $\sigma_{i,k}$ and $v_{i,k}$ are chosen, the triggering condition is more accurate to be reached. For simplicity, we introduce parameters v_i and σ_i , where $v_i = \min\{\sigma_{i,k}\}$ and $\sigma_i = \min\{\sigma_{i,k}\}$. Overall, the parameters are selected as follows: $K_1 = 0.1$; $K_2 = 3.8$; $K_3 = 1.1$; $v_1 = v_2 = v_3 = 1$; $\sigma_1 = 0.0001$; $\sigma_2 = 0.001$; and $\sigma_3 = 0.001$, and the sampling time $T = 0.1s$.

The control consensus protocol (4) and the control consensus protocol (16) based on event trigger with delay compensation introduced are compared and tested in the NMAS in this paper. Based on the simulation results, it can be concluded that the considered system can reach external consensus with control protocol without delay compensation (4) when the delay is within 2-step and below, as shown at Figure 2. However, as can be seen from the numerical simulation results in Figure 3, the NMAS with $\tau = 4$ produces an oscillating output under the traditional external consensus control consensus protocol (4). As the sampling time goes on, the oscillation of the system becomes serious. For example, when the sample time is 600, the output values of agent 1, agent 2, and agent 3 are 4.70, 20.05, and 10.96, respectively, far exceeding our expectation of 5. Hence, to compare and verify the effectiveness of the event-triggered-based and delay-compensated control protocol (16) proposed in this paper, a 5-step delay is considered.

The predictive control algorithm and event-triggering mechanism are applied to the NMAS with $\tau = 5$, and the simulation results are shown in Figure 4. It is observed that the considered system can be stable and reach external consensus with both control consensus protocols (16). Due to the addition of the time delay compensation mechanism, an NMAS with communication delay can produce a stable output, thus proving the feasibility and effectiveness of the proposed algorithm in this paper. Moreover, the triggering situation of each agent within a period is listed in Table 1 and shown in Figure 5. It shows that the time-based control protocol in [8] has a total of 20000 triggering times in the simulations. In contrast, the event-triggered-based control protocol (16) can actively and effectively reduce the triggering times.

It is readily seen that network-induced delay can be compensated, and uninterrupted data transmission can be avoided by the proposed control protocol effectively. At the same time, the external consensus problem can be solved.

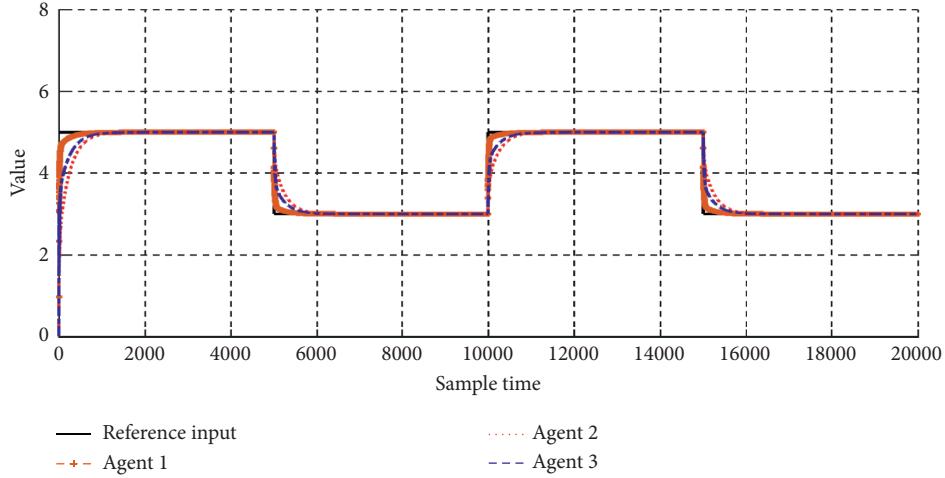
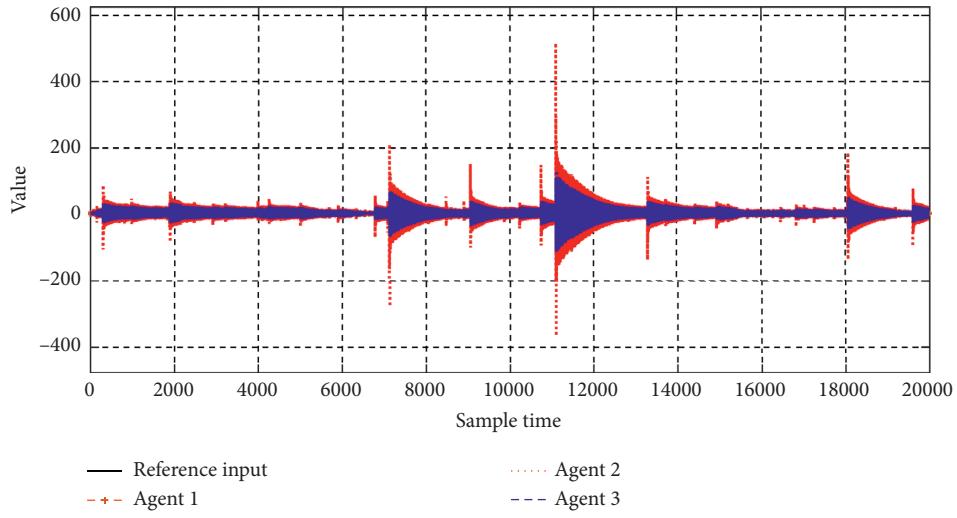
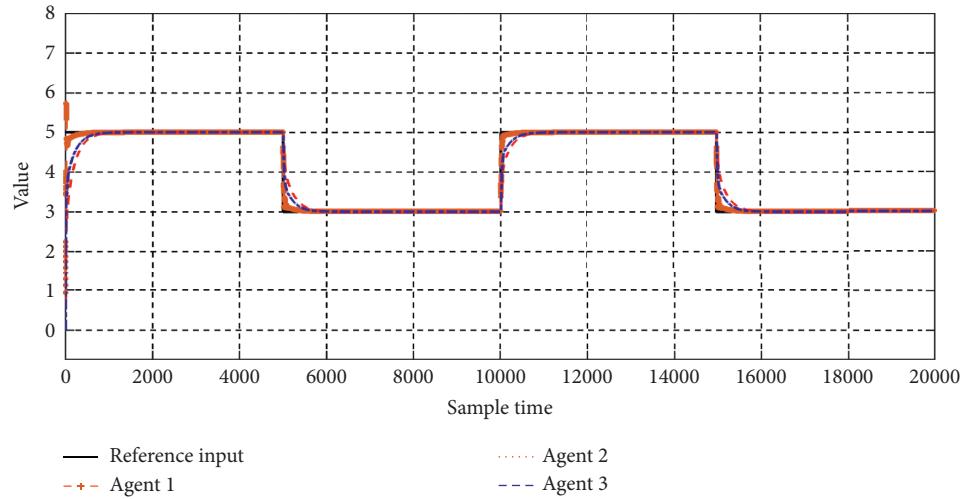
FIGURE 2: Result with $\tau = 2$ -step and protocol (4).FIGURE 3: Result with $\tau = 4$ -step and protocol (4).FIGURE 4: Result with $\tau = 5$ -step and protocol (16).

TABLE 1: Trigger situation of each agent.

Input	Triggering times with control protocol in [8]	Triggering times with control protocol (17)
u_1	20000	532
u_2	20000	973
u_3	20000	2864

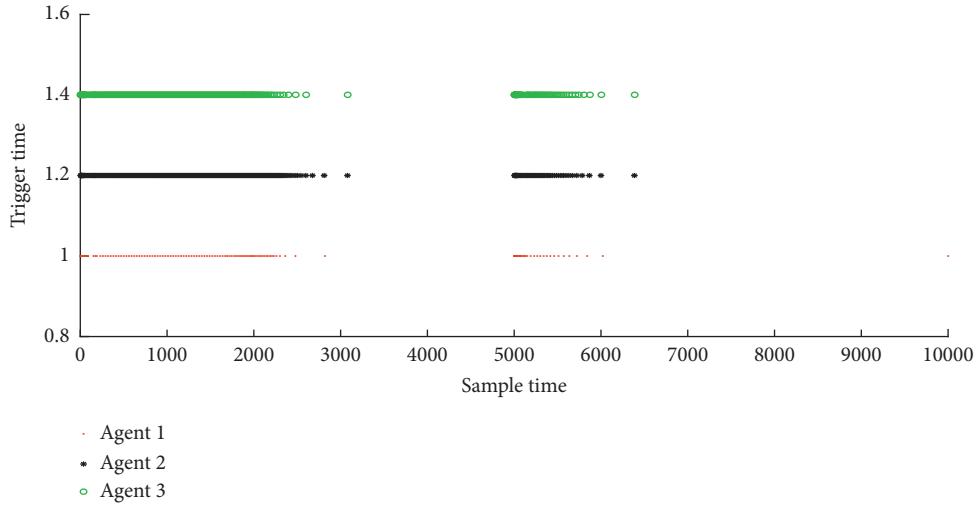


FIGURE 5: The trigger time of each agent.

6. Conclusions

In this paper, the external consensus problem of an NMAS with nonlinear dynamics and network-induced delay is investigated. An RBF-ARX modelling approach is used to approximate the nonlinear discrete-time dynamics and construct the global nonlinear model, called the RBF-ARX model. An RBF-ARX-model-based prediction strategy is applied to predict the output sequence of agents to compensate the negative effect due to network-induced delay. By using the delay-compensated output of the neighbouring agents, an event-triggered mechanism with delay compensation is designed to reduce the times of the agent's control update and communication between others. Consequently, an event-triggered-based and RBF-ARX-based delay-compensated external consensus control protocol is proposed. Both the external consensus and stability of the underlying system are discussed, and the validity of the proposed control protocol is demonstrated through simulation results. In future work, NMASs with communication time-varying delays will be considered. Also, we will replace the maximum eigenvalue by other local information of networks as far as possible. Additionally, we will improve the prediction strategy so that it can adapt to a variety of complex situations.

Data Availability

The data used to support the findings of this study have not been made available.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61573379 and in part by the 111 Project under Grant B17040.

References

- [1] M. Hasanuzzaman Shawon, S. M. Muyeen, A. Ghosh, S. M. Islam, and M. S. Baptista, "Multi-agent systems in ICT enabled smart grid: a status update on technology framework and applications," *IEEE Access*, vol. 7, pp. 97959–97973, 2019.
- [2] B. Teixeira, G. Santos, T. Pinto, Z. Vale, and J. M. Corchado, "Application ontology for multi-agent and web-services' co-simulation in power and energy systems," *IEEE Access*, vol. 8, pp. 81129–81141, 2020.
- [3] P. Leitão, S. Karnouskos, L. Ribeiro, J. Lee, T. Strasser, and A. W. Colombo, "Smart agents in industrial cyber-physical systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1086–1101, 2016.
- [4] Y. Sun, Z. Ji, Q. Qi, and H. Ma, "Bipartite consensus of multi-agent systems with intermittent interaction," *IEEE Access*, vol. 7, pp. 130300–130311, 2019.
- [5] R. M. Hadi and M. M. Bagher, "Adaptive output stationary average consensus for heterogeneous unknown linear multi-agent systems," *IET Control Theory and Applications*, vol. 12, no. 7, pp. 847–856, 2018.

- [6] B. M. Nejad, S. A. Attia, and J. Raisch, "Max-consensus in a max-plus algebraic setting: the case of fixed communication topologies," in *Proceedings of 2009 XXII International Symposium on Information, Communication and Automation Technologies*, pp. 1–7, Sarajevo, Bosnia and Herzegovina, October 2009.
- [7] J. Cortés, "Distributed algorithms for reaching consensus on general functions," *Automatica*, vol. 44, no. 3, pp. 726–737, 2008.
- [8] N. A. M. Subha and G.-P. Liu, "Design and practical implementation of external consensus protocol for networked multiagent systems with communication delays," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 619–631, 2015.
- [9] S. Li, J. Wang, X. Luo, and X. Guan, "A new framework of consensus protocol design for complex multi-agent systems," *Systems & Control Letters*, vol. 60, no. 1, pp. 19–26, 2011.
- [10] S. Yang and J.-X. Xu, "Improvements on "A new framework of consensus protocol design for complex multi-agent systems," *Systems & Control Letters*, vol. 61, no. 9, pp. 945–949, 2012.
- [11] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [12] M.-M. Duan, C.-L. Liu, and F. Liu, "Event-triggered consensus seeking of heterogeneous first-order agents with input delay," *IEEE Access*, vol. 5, pp. 5215–5223, 2017.
- [13] X. Wang, S. Li, and P. Shi, "Distributed finite-time containment control for double-integrator multiagent systems," *IEEE Transactions on Cybernetics*, vol. 44, no. 9, pp. 1518–1528, 2014.
- [14] X. Wang, S. Li, X. Yu, and J. Yang, "Distributed active anti-disturbance consensus for leader-follower higher-order multi-agent systems with mismatched disturbances," *IEEE Transactions on Automatic Control*, vol. 62, no. 11, pp. 5795–5801, 2017.
- [15] J. Wei and H. Fang, "Multi-agent consensus with time-varying delays and switching topologies," *Journal of Systems Engineering and Electronics*, vol. 25, no. 3, pp. 489–495, 2014.
- [16] Z. Ji, H. Lin, S. Cao, Q. Qi, and H. Ma, "The complexity in complete graphic characterizations of multiagent controllability," *IEEE Transactions on Cybernetics*, vol. 51, no. 1, pp. 64–76, 2021.
- [17] J. Qu, Z. Ji, and Y. Shi, "The graphical conditions for controllability of multiagent systems under equitable partition," *IEEE Transactions on Cybernetics*, pp. 1–12, in press, 2020.
- [18] 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 Transactions on Industrial Electronics*, vol. 63, no. 2, pp. 1249–1257, 2016.
- [19] Y. B. Huang, Y. He, J. An, and M. Wu, "Polynomial-type lyapunov-krasovskii functional and Jacobi-bessel inequality: further results on stability analysis of time-delay systems," *IEEE Transactions on Automatic Control*, 2020.
- [20] F. Long, C. Zhang, L. Jiang, Y. He, and M. Wu, "Stability analysis of systems with time-varying delay via improved lyapunov-krasovskii functionals," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, pp. 1–10, 2019.
- [21] C.-K. Zhang, F. Long, Y. He, W. Yao, L. Jiang, and M. Wu, "A relaxed quadratic function negative-determination lemma and its application to time-delay systems," *Automatica*, vol. 113, Article ID 108764, 2020.
- [22] F. Xiao, T. Chen, and H. Gao, "Synchronous hybrid event-and time-driven consensus in multiagent networks with time delays," *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1165–1174, 2016.
- [23] X. Xu, L. Liu, and G. Feng, "Consensus of discrete-time linear multiagent systems with communication, input and output delays," *IEEE Transactions on Automatic Control*, vol. 63, no. 2, pp. 492–497, 2018.
- [24] G.-P. Liu, Y. Xia, J. Chen, D. Rees, and W. Hu, "Networked predictive control of systems with random network delays in both forward and feedback channels," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 3, pp. 1282–1297, 2007.
- [25] G. P. Liu, "Consensus and stability analysis of networked multiagent predictive control systems," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, p. 1114, 2016.
- [26] G.-P. Liu, "Predictive control of networked multiagent systems via cloud computing," *IEEE Transactions on Cybernetics*, vol. 47, no. 8, pp. 1852–1859, 2017.
- [27] X. Shi, C.-C. Lim, P. Shi, and S. Xu, "Adaptive neural dynamic surface control for nonstrict-feedback systems with output dead zone," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5200–5213, 2018.
- [28] D. Xu, J. Liu, X.-G. Yan, and W. Yan, "A novel adaptive neural network constrained control for a multi-area interconnected power system with hybrid energy storage," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 8, pp. 6625–6634, 2018.
- [29] X. Liu, Y. Zhou, Z. Wang, and X. Chen, "A bp neural network-based communication blind signal detection method with cyber-physical-social systems," *IEEE Access*, vol. 6, pp. 43920–43935, 2018.
- [30] H. Peng, T. Ozaki, V. Hagan-Ozaki, and Y. Toyoda, "A parameter optimization method for radial basis function type models," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 432–438, 2003.
- [31] H. Peng, T. Ozaki, Y. Toyoda et al., "Rbf-ark model-based nonlinear system modeling and predictive control with application to a nox decomposition process," *Control Engineering Practice*, vol. 12, no. 2, pp. 191–203, 2004.
- [32] H. Peng, J. Wu, G. Inoussa, Q. Deng, and K. Nakano, "Nonlinear system modeling and predictive control using the rbf nets-based quasi-linear arx model," *Control Engineering Practice*, vol. 17, no. 1, pp. 59–66, 2009.
- [33] F. Zhou, H. Peng, Y. Qin, X. Zeng, X. Tian, and W. Xu, "A rbf-ark model-based robust mpc for tracking control without steady state knowledge," *Journal of Process Control*, vol. 51, pp. 42–54, 2017.
- [34] X. Yin, D. Yue, and S. Hu, "Distributed event-triggered control of discrete-time heterogeneous multi-agent systems," *Journal of the Franklin Institute*, vol. 350, no. 3, pp. 651–669, 2013.
- [35] W. Hu, L. Liu, and G. Feng, "Consensus of linear multiagent systems by distributed event-triggered strategy," *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 148–157, 2017.
- [36] L. Ding, Q.-L. Han, X. Ge, and X.-M. Zhang, "An overview of recent advances in event-triggered consensus of multiagent systems," *IEEE Transactions on Cybernetics*, vol. 48, no. 4, pp. 1110–1123, 2018.
- [37] D. V. Dimarogonas and K. H. Johansson, "Event-triggered control for multi-agent systems," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference*, pp. 7131–7136, Piscataway, NJ, USA, October 2009.

- [38] Z.-G. Wu, Y. Xu, R. Lu, Y. Wu, and T. Huang, "Event-triggered control for consensus of multiagent systems with fixed/switching topologies," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 10, pp. 1736–1746, 2018.
- [39] Y. S. Sun, Z. J. Ji, and K. E. Liu, "Event-based consensus for general linear multiagent systems under switching topologies," *Complexity*, vol. 2020, Article ID 5972749, 14 pages, 2020.
- [40] Q. Jia and W. K. S. Tang, "Event-triggered protocol for the consensus of multi-agent systems with state-dependent nonlinear coupling," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 2, pp. 723–732, 2018.