Predictive Control of Networked Multi-agent Systems via Cloud Computing

Guo-Ping Liu, *Fellow, IEEE*

Abstract -- The paper studies the design and analysis of networked multi-agent predictive control systems via cloud computing. A cloud predictive control scheme for networked multi-agent systems is proposed to achieve consensus and stability simultaneously and to compensate for network delays actively. The design of the cloud predictive controller for networked multi-agent systems is detailed. The analysis of the cloud predictive control scheme gives the necessary and sufficient conditions of stability and consensus of closed-loop networked multi-agent control systems. The proposed scheme is verified to characterise the dynamical behaviour and control performance of networked multi-agent systems through simulations. The outcome provides a foundation for the development of cooperative and coordinative control of networked multi-agent systems and its applications.

***Keywords*:** Multi-agent systems, networked control systems, cloud predictive control, consensus and stability.

I. INTRODUCTION

A multi-agent system (MAS) is a set of agents communicating each other, where each agent is an abstract or physical entity. In recent years, further development of communication technology, particularly Internet technology, has lead to a number of multi-agent systems that employ communication networks to exchange information mutually. This results in a new system named a networked multi-agent system (NMAS). This system is generally composed of many simple agents/sub-systems interacting via networks. The most important NMAS application is the Internet of things (IoT) [1], which is one of the hottest growth sectors in the global economy.

Various NMAS are widely used in the fields of sciences and engineering, such as smart grid, satellite communications, GPS, robot networks, biological networks, sensor networks, unmanned vehicles, power systems, etc [2-5]. Each agent in an NMAS has its own distributed knowledge, capabilities or skills when performing specific actions. However, it is unusual and even useless for an isolated agent to act individually despite of the common loosely-coupled network topology. All agents in NMAS are expected to be situated in a similar environment and they can communicate through a series of interaction protocols. Therefore, NMAS can be used to model many existing complex systems and its corresponding research can bring us new methods to deal with problems which can't be resolved by any individual agent. As for the advantages related to the usage of the NMAS technology, there are so many good properties compared with other current available methods, such as reliability, flexibility, robustness, reusability, extensibility and maintainability, etc.

NMAS synthesis involves the generation of a desired collective behavior by local interaction protocols among the agents. The main research on NMAS can be classified into two directions: the development of distributed estimation techniques for sensor networks, and the control of mobile autonomous agents using information obtained over networks. Various important contributions to both the directions have been made in past years. More specifically, control strategies of mobile robot formations have been studied by employing methods from control system theory to graph theory and a number of all possible transitions which the robots can have in formations have been presented in [6]. Based on the virtual structure approach, the formation control ideas for multiple spacecraft have been addressed [7]. Formation control and simultaneous tracking have been investigated for a group of autonomous agents evolving dynamically in a space which possesses a measurable vector field and the proposed methods can guarantee that agents' desired formation can be achieved and maintained by controlling their trajectories cooperatively [8]. A simply distributed algorithm achieving global stabilisation of formations has been proposed for relative sensing networks in arbitrary dimensions and the convergence properties have been analysed, based on the algorithm involved with scaling theory and distributed linear iterations [9]. An inverse optimal approach has been presented for distributed cooperative optimal control of multi-agent systems on directed graphs [10]. A consensus control problem of multi-agents with an active leader and variable interconnection topology has been addressed, and a neighbor-based local controller and a neighbor-based state-estimation rule have been given for each autonomous agent in order to follow such a leader [11]. An asynchronous consensus strategy for continuous-time MAS with variable delays and switching topology have been presented and a valid distributed consensus algorithm has been provided to overcome difficulties caused by unreliable communication channels [12].

NMAS analysis studies how global objectives are affected by network architectures and interactions between network components. The key issue of NMAS is the consensus problem that requires an agreement to be achieved with all agents. It implies that distributed control strategies need to be designed using local information so that all agents reach required agreements on certain quantities of interest. This topic has been addressed across various fields of engineering and science, and many interesting results on the consensus problem have been obtained. The consensus ability of NMAS is usually involved with each agent's isolated dynamics and its connection topology structure. Once the isolated agent dynamics are determined, the consensus ability of NMAS depends on its connection topology structure. In NMAS, it has been witnessed that various pioneering contributions are involved with different

This work was supported by the National Natural Science Foundation of China under Grant 61273104 and Grant 61333003.
Guo-Ping Liu is with the School of Engineering, University of South Wales, Pontypridd CF37 1DL, UK and also with CTGT Center in Harbin Institute of Technology, Harbin 150001, China. (e-mail: guoping.liu@southwales.ac.uk)

distributed strategies achieving consensus. Consensus criteria for communication networks with and without network delays have been introduced with convergence analysis for MAS with fixed and switching topologies [13, 14]. The average consensus problem for undirected NMAS with network delays has been discussed and sufficient conditions for its existence have been derived under bounded network delays [15]. The asymptotic average consensus problem for multi-agent systems with time-varying delay has been addressed in [16] and the consensus problem for second-order Markovian jump multi-agent systems with random network delay has been considered using stochastic switching topology in [17]. A complex dynamic network (CDN), which generally consists of a large number of interconnected dynamic nodes, has attracted tremendous attention recently [18]. Since the communication connection topology plays a crucial role in forming CDN behaviors, a variety of connection topologies have been examined to understand how the communication topology of networks affects the behavior of networks. Synchronisation of CND is one of the most important issues that influence the behaviour of networks and has extensively been studied for CDN with non-identical nodes [19]. The ideas and results in the synchronization problem of CDN have been applied to the consensus problem of NMAS [20].

The research on NMAS has reduced the expenses for establishment, operation and maintenance of the system tremendously based on its computational efficiency and speed. However, the research on NMAS is still confronted with many challenges and difficulties especially for design and analysis. For example, how to formulate or decompose the relevant tasks and objectives; how to design efficient and effective control protocols to compensate for communication constraints; how to guarantee the stability and achieve the consensus simultaneously. Those issues need to be solved for wide applications of NMAS, particularly for the Internet of Things. To address the above listed challenges and difficulties, this paper considers two objectives - stability and consensus of NMAS, proposes the cloud predictive control scheme to compensate for communication delays actively and efficiently, and derives necessary and sufficient conditions of stability and consensus of closed-loop NMAS. Also, simulations illustrate the dynamical behaviour and control performance of NMAS using the proposed cloud predictive control scheme.

II. NETWORKED MULTI-AGENT CONTROL SYSTEMS VIA CLOUD COMPUTING

In NMAS, there are multiple controllers rather than a single controller and also there exist interactions among the agents that the individual agent controllers must consider. Each individual agent controller adopts a control strategy, *e.g.*, networked predictive control [21]. It now considers not only dynamics and objectives but also communication constraints. Each agent controller solves a control problem based on its own information and shared information with the other agent controllers to improve the overall performance. In NMAS, as the scale of the system increases, the captured real-time data and required real-time computing will grow in size dramatically. There exist many challenges, including capturing, storage, visualization, sharing, transfer, search and analysis of the data, and allocation and coordination of the computing tasks. It is hard to handle this kind of real-time big data and computing using traditional database management and processing tools.

With the development of network technology and computing technology, cloud computing has come into our daily life. Now, cloud computing has exceeded its original product concept and become a service. It provides a shared pool of dynamically scalable and virtualized resources, including data access, computation, software and storage services. The concept of cloud control systems has briefly been discussed as an extension of networked control systems in [22, 23], which states that the research on cloud control systems will make new contributions to control system theory and practical applications in the very near future. This paper introduces both cloud computing and predictive control into networked multi-agent control systems in a generic form to overcome their challenging issues, particularly real-time big data, communication delays, heavy computing, and coordination of multiple tasks, which exist in conventional NMAS. Due to the merits of cloud computing, a cloud based control strategy is proposed for NMAS in this paper. The architecture of networked multi-agent control systems via cloud computing is shown in Figure 1.

Using the cloud based control strategy, a control method of NMAS will be implemented via cloud computing. The captured real-time data of NMAS by the sensors will be sent out to a cloud computing system via networks, after the data being processed by following a networked control method, the control sequences will be generated and sent back to the actuators of individual agents through networks. Each agent is linked to a controller cloud node in the cloud computing system via networks. All the controller cloud nodes are linked and exchange information in the cloud computing system. Each controller cloud node has functions of the task management, data collection and computation, and keeps broadcasting a request over its domain at a certain frequency so that all the non-controller cloud nodes in its covering domain can receive this request. In each sampling period, some suitable non-controller cloud nodes will be chosen to carry out the various sub-tasks (*e.g.*, state estimation, parameter optimsation, agent cooperation, control prediction, *etc*.) assigned by the controller cloud node and return the results to the controller cloud node. In this way, the cloud based control strategy will provide a powerful tool for the control of NMAS, which could not be imagined before.

Since all agents in NMAS communicate with a cloud computing system via networks, there exist communication constraints that affect the modelling and design of NMAS. The key communication constraints are network delays, data dropouts and data security. They seriously affect the control performance of NMAS. The network delays are delay s_i between the sensors of the i-th agent and the cloud computing system, and delay a_i between the actuators of the i-th agent and the cloud computing system, $\forall i \in \mathbb{N}$, where $\mathbb{N} = \{1, 2, \ldots, N\}$ and a_i and s_i are integers. The compensation for network delays will be considered in the next section. The data dropouts and data security can be dealt by the strategies used in [21] and [24], respectively.

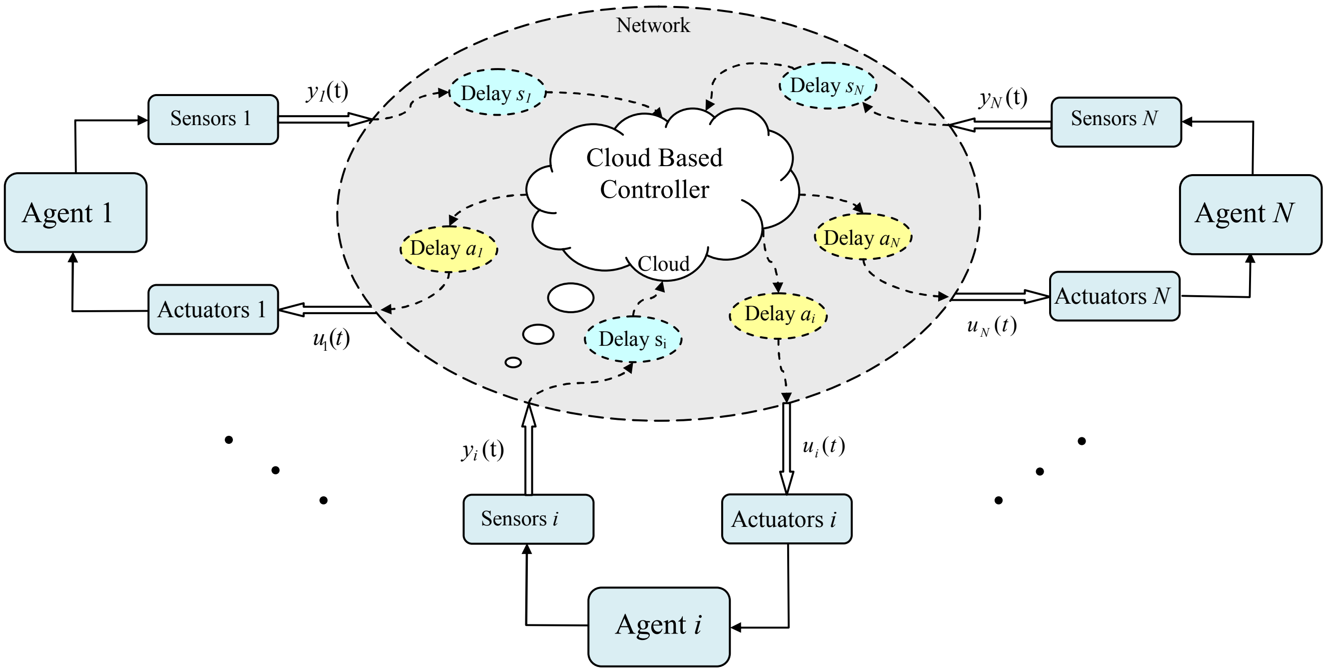

Figure 1. The networked multi-agent control system via cloud computing

III. DESIGN OF CLOUD PREDICTIVE CONTROLLERS FOR NETWORKED MULTI-AGENT SYSTEMS

Regardless of the type of networks, the performance of NMAS is always affected by communication constraints. These constraints are widely known to degrade the performance of a networked system. Therefore, to handle communication constraints in a closed-loop NMAS, an advanced control methodology is required. To deal with these constraints, a novel networked control structure needs to be developed. One of the key features of communication networks is that they can transmit a packet of data each time rather than a single data, which cannot be done in non-networked control systems. Making full use of this network feature, a networked predictive control (NPC) strategy [21] will be introduced for NMAS, which can overcome the effects caused by random network delays and data dropouts. This strategy will consist of a control prediction generator and a network constraint compensator. The control prediction generator, based on the dynamical model and a performance function to be optimised, will produce a control prediction sequence using available information up to time t, which contains several step ahead control predictions from time t onward. This control prediction sequence will be packed together and transmitted to the controlled agent side via networks. The network constraint compensator on the controlled agent side will choose the latest control prediction for time t from all available control prediction sequences in terms of the type of communication constraints and apply it to the actuator of the controlled agent. In this way, the communication constraints, particularly network delays and data dropouts, will be compensated actively and the networked predictive control strategy will provide the same or asymptotically same control performance as that of NMAS without communication constraints.

To simplify the presentation, the following assumptions are made: 1) there exist network delay s_i between the cloud controller and the i-th agent sensors, and network delay a_i between the cloud controller and the i-th agent actuators; 2) the network delays between nodes in the cloud computing system are much smaller than the networked delays and are ignored; 3) network delays s_i and a_i are known integers, which are multiples of the sampling rate of the agents.

To illustrate how the cloud predictive control scheme is designed, analysed and performed easily, the linear non-identical multi-agents are considered below. Actually, this scheme can be extended to more general NMAS, e.g., nonlinear NMAS with uncertainties and disturbances.

$$\begin{aligned} x_i(t+1) &= A_i x_i(t) + B_i u_i(t) \\ y_i(t) &= C_i x_i(t) \end{aligned} \tag{1}$$

$\forall i \in \mathbb{N}$, where $x_i \in \Re^{n_i}$, $y_i \in \Re^{l}$ and $u_i \in \Re^{m_i}$ are the state, output and input vectors of the i-th agent, respectively, and $A_i \in \Re^{n_i \times n_i}$, $B_i \in \Re^{n_i \times m_i}$ and $C_i \in \Re^{l \times n_i}$ are the matrices of the i-th agent.

It is assumed that all the agents are observerable but their states are immeasurable. Then, based on the output $y_i\,(t\text{-}s_i)$ and control input $u_i\,(t\text{-}s_i)$, a state observer for the i-th agent is designed as follows:

$$\begin{aligned} \hat{x}_i(t\text{-}s_i+1|t\text{-}s_i) &= A_i\hat{x}_i(t\text{-}s_i|t\text{-}s_i\text{-}1) + B_i u_i(t\text{-}s_i) + F_i\left(y_i(t\text{-}s_i) - \hat{y}_i(t\text{-}s_i|t\text{-}s_i\text{-}1)\right) \\ \hat{y}_i(t\text{-}s_i|t\text{-}s_i\text{-}1) &= C_i\hat{x}_i(t\text{-}s_i|t\text{-}s_i\text{-}1) \end{aligned} \tag{2}$$

where $\hat{x}_i(t-k\,|\,t-j)\in\Re^{n_i}$ $(k<j)$ represents the state prediction of the i-th agent for time t-k based on the available information up to time t-j, $\hat{y}_i(.|.)\in\Re^{l_i}$ is the output prediction, and $F_i\in\Re^{n_i\times l_i}$ is the observer gain matrix.

To predict the states of the i-th agent using the available information up to time t-s_i, the following state estimations for time from t-s_i+2 to t+a_i can be utilised.

$$\hat{x}_i(t-s_i+k|t-s_i) = A_i\hat{x}_i(t-s_i+k-1|t-s_i) + B_iu_i(t-s_i+k-1) \quad (3)$$

$$\hat{y}_i(t-s_i+k|t-s_i) = C_i\hat{x}_i(t-s_i+k|t-s_i) \quad (4)$$

for $k = 2,3,\dots,s_i+a_i$.

It is assumed that the desired reference input is denoted by a step signal vector r_0 and is only applied to one of the agents, e.g., the first agent with $a_1 \geqslant a_i$, $\forall i\in\mathbb{N}-\{1\}$. To track this desired reference input, a set of dynamical variables are introduced below.

$$z_1(t+1+a_1) = z_1(t+a_1) + \hat{y}_1(t+a_1|t-s_1) - r_0 \quad (5)$$

$$z_i(t+1+a_i) = z_i(t+a_i) + \hat{y}_i(t+a_i|t-s_i) - \hat{y}_1(t+a_i|t-s_1) \quad (6)$$

The effect of the dynamic variables in (5) and (6) is equivalent to the integration action in the conventional control systems, which can eliminate the state-steady tracking error. To compensate for the network delays s_i and a_i, $\forall i\in\mathbb{N}$ actively, a predictive control protocol for NMAS is presented as follows:

$$\hat{u}_i(t+a_i|t-s_i)=G_iz_i(t+a_i) + H_i\sum_{j=1}^{N}c_{ij}(\hat{y}_j(t+a_i|t-s_j)-\hat{y}_i(t+a_i|t-s_i)) \quad (7)$$

where

$$c_{ij}=\begin{cases}1, & \text{if } a_i\le a_j\\ 0, & \text{if } a_i>a_j\end{cases} \quad (8)$$

$G_i\in\Re^{m_i\times m_i}$ and $H_i\in\Re^{m_i\times l_i}$ are the gain matrices to be designed. The above implies that the predictive control protocol utilises the predictions of the outputs based on the information available upto time t-s_i, $\forall i\in\mathbb{N}$ to estimate the future control actions at time t+a_i, $\forall i\in\mathbb{N}$. Actually, the proposed predictive control protocol consists of two parts. One is for agent 1 to track the desired reference and for the other agents to track the output of agent 1, which is represented by the first item on the right hand side in (7). The other is the coordination between the agents, which is represented by the second item on the right hand side in (7).

Then, the predictive control input of the i-th agent is designed to be

$$u_i(t+a_i)=\hat{u}_i(t+a_i|t-s_i) \quad (9)$$

So, the control input of the i-th agent is given by

$$u_i(t)=\hat{u}_i(t|t-s_i-a_i) \quad (10)$$

Thus, the cloud predictive control scheme is proposed as follows:

a) The output data $y_i(t)$, $\forall i\in\mathbb{N}$, of all the agents from the sensors are sent to networks at each sampling time t.
b) Based on the received output data upto $y_i(t$-$s_i)$, $\forall i\in\mathbb{N}$ from the networks, the cloud computing system calculates the predictions $\hat{x}_i(t+a_i\,|\,t-s_i)$, $\hat{y}_i(t+a_i\,|\,t-s_i)$, $\hat{u}_i(t+a_i\,|\,t-s_i)$, $\forall i\in\mathbb{N}$, of the states, outputs and control inputs of the agents using (3), (4) and (7), respectively and dynamical variables $z_i(t$+$a_i)$, $\forall i\in\mathbb{N}$ using (5) and (6).
c) The control input predictions $u_i(t$+$a_i)$, $\forall i\in\mathbb{N}$ given by (9) are sent from the cloud computing system to the actuators of each agent via networks.
d) The actuators of all the agents receive the control input $u_i(t)$, $\forall i\in\mathbb{N}$ given by (10) from networks at each sampling time t.

The cloud predictive control scheme for NMAS is illustrated in Figure 2. It shows how the cloud predictive controller generates the agent control inputs $u_i(t$+$a_i)$, $\forall i\in\mathbb{N}$ after calculation via a cloud computing system, based on the agent outputs $y_i(t$-$s_i)$, $\forall i\in\mathbb{N}$.

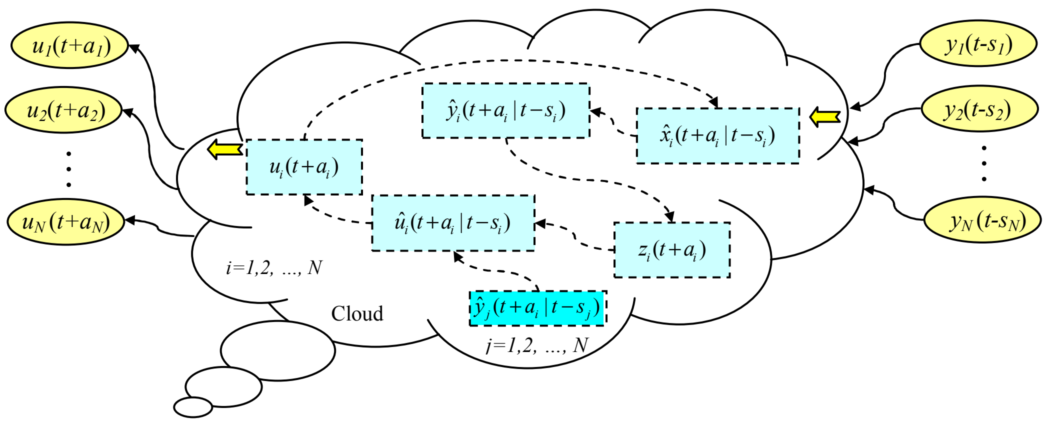

Fig 2. The cloud predictive controller for networked multi-agent systems

IV. STABILITY AND CONSENSUS ANALYSIS OF CLOUD PREDICTIVE CONTROL SYSTEMS

In practice, multi-agent control systems need to consider not only the consensus and but also stability. Both the consensus and stability are very important issues in multi-agent control systems. But the stability issue is often ignored in current research work of multi-agent control systems. Here, both consensus and stability of closed-loop NMAS with the cloud predictive control scheme are simultaneously analysed.

Definition 1: Networked multi-agent system (1) with the cloud predictive control scheme is input-output stable and achieves the output consensus if the following criteria are satisfied: 1) $\lim_{t\to\infty}\|y_i(t)-y_j(t)\|=0, \forall i,j\in\mathbb{N}$ and 2) $\lim_{t\to\infty}\|y_i(t)\|<\infty$ if $\|r_0\|<\infty$, for $t\geq 0, \forall i\in\mathbb{N}$, where r_0 is the reference input.

Condition 1) in Definition 1 implies all the agents achieves output consensus and Condition 2) means each individual agent is input-output stable.

Shifting t by s_i steps forward in observer (2) gives

$$\begin{aligned}\hat{x}_i(t+1|t) &= A_i\hat{x}_i(t|t-1)+B_iu_i(t)+F_i(y_i(t)-\hat{y}_i(t|t-1))\\ \hat{y}_i(t|t-1) &= C_i\hat{x}_i(t|t-1)\end{aligned} \tag{11}$$

Define the state estimation error as $e_i(t)=x_i(t)-\hat{x}_i(t|t-1)$. The following equation can be obtained from (1) and (11):

$$e_i(t+1)=(A_i-F_iC_i)e_i(t) \tag{12}$$

Let the total network delay of the i-th agent be $\tau_i=a_i+s_i$ and the mixed network delay of the i-th and j-th agents be $\tau_{ij}=a_i+s_j$. Calculating the state prediction using (3) recursively results in the following τ_i-th step ahead state prediction:

$$\begin{aligned}\hat{x}_i(t+a_i|t-s_i) &= A_i^{\tau_i-1}\hat{x}_i(t-s_i+1|t-s_i)+\sum_{k=2}^{\tau_i}A_i^{\tau_i-k}B_iu_i(t+k-s_i-1)\\ &= A_i^{\tau_i-1}(A_i-F_iC_i)\hat{x}_i(t-s_i|t-s_i-1)+A_i^{\tau_i-1}F_iC_ix_i(t-s_i)\\ &\quad+\sum_{k=1}^{\tau_i}A_i^{\tau_i-k}B_iu_i(t+k-s_i-1)\end{aligned} \tag{13}$$

Let t be replaced by $t-a_i$ in the above equation. Then

$$\begin{aligned}\hat{x}_i(t|t-\tau_i) &= A_i^{\tau_i-1}(A_i-F_iC_i)\hat{x}_i(t-\tau_i|t-\tau_i-1)+A_i^{\tau_i-1}F_iC_ix_i(t-\tau_i)\\ &\quad+\sum_{k=1}^{\tau_i}A_i^{\tau_i-k}B_iu_i(t+k-\tau_i-1)\end{aligned} \tag{14}$$

Utilising (1) recursively leads to

$$x_i(t)=A_i^{\tau_i}x_i(t-\tau_i)+\sum_{k=1}^{\tau_i}A_i^{\tau_i-k}B_iu_i(t+k-\tau_i-1) \tag{15}$$

which means

$$\sum_{k=1}^{\tau_i}A_i^{\tau_i-k}B_iu_i(t+k-\tau_i-1)=x_i(t)-A_i^{\tau_i}x_i(t-\tau_i) \tag{16}$$

Replacing the third term on the right hand side of (14) by (16) yields

$$\begin{aligned}\hat{x}_i(t|t-\tau_i) &= A_i^{\tau_i-1}(A_i-F_iC_i)\hat{x}_i(t-\tau_i|t-\tau_i-1)\\ &\quad+A_i^{\tau_i-1}F_iC_ix_i(t-\tau_i)+x_i(t)-A_i^{\tau_i}x_i(t-\tau_i)\\ &= x_i(t)+A_i^{\tau_i-1}(A_i-F_iC_i)\hat{x}_i(t-\tau_i|t-\tau_i-1)\\ &\quad-A_i^{\tau_i-1}(A_i-F_iC_i)x_i(t-\tau_i)\\ &= x_i(t)-A_i^{\tau_i-1}(A_i-F_iC_i)e_i(t-\tau_i)\\ &= x_i(t)-A_i^{\tau_i-1}e_i(t-\tau_i+1)\end{aligned} \tag{17}$$

which uses (12). Employing the above state prediction, the cloud predictive control protocol (10) can be rewritten as

$$\begin{aligned}u_i(t) &= \hat{u}_i(t|t-s_i-a_i)=\hat{u}_i(t|t-\tau_i)\\ &= G_iz_i(t)+H_i\sum_{j=1}^{N}c_{ij}(C_j\hat{x}_j(t|t-\tau_{ij})-C_i\hat{x}_i(t|t-\tau_i))\\ &= G_iz_i(t)+H_i\sum_{j=1}^{N}c_{ij}(C_jx_j(t)-C_ix_i(t))\\ &\quad+H_i\sum_{j=1}^{N}c_{ij}(C_iA_i^{\tau_i-1}e_i(t-\tau_i+1)-C_jA_j^{\tau_{ij}-1}e_j(t-\tau_{ij}+1))\end{aligned} \tag{18}$$

Define $\Delta x_i(t)=x_i(t)-x_i(t-1)$ and $\Delta z_i(t)=z_i(t)-z_i(t-1)$. From (1) and (18), it is straightforward to obtain the following state increment of the i-th agent:

$$\begin{aligned}\Delta x_i(t+1) &= A_i\Delta x_i(t)+B_iG_i\Delta z_i(t)\\ &\quad+B_iH_i\sum_{j=1}^{N}c_{ij}(C_j\Delta x_j(t)-C_i\Delta x_i(t))\\ &\quad+B_iH_i\sum_{j=1}^{N}c_{ij}(C_iA_i^{\tau_i-1}\Delta e_i(t-\tau_i+1)-C_jA_j^{\tau_{ij}-1}\Delta e_j(t-\tau_{ij}+1))\\ &= (A_i-B_id_iH_i)C_i)\Delta x_i(t)+B_iG_i\Delta z_i(t)\\ &\quad+B_iH_i\sum_{j=1}^{N}c_{ij}C_j\Delta x_j(t)\\ &\quad+B_iH_i\sum_{j=1}^{N}c_{ij}(C_iA_i^{\tau_i-1}\Delta e_i(t-\tau_i+1)-C_jA_j^{\tau_{ij}-1}\Delta e_j(t-\tau_{ij}+1))\end{aligned} \tag{19}$$

where $d_i=c_{i1}+c_{i2}+\ldots+c_{iN}$. Thus, all the state increments of NMAS can be expressed in the following compact form:

$$\Delta X(t+1)=(A_d-B_h+B_a)\Delta X(t)+B_g\Delta Z(t)+V(t) \tag{20}$$

where

$$\Delta X(t)=\begin{bmatrix}\Delta x_1^T(t) & \Delta x_2^T(t) & \cdots & \Delta x_N^T(t)\end{bmatrix}^T$$

$$\Delta Z(t)=\begin{bmatrix}\Delta z_1^T(t) & \Delta z_2^T(t) & \cdots & \Delta z_N^T(t)\end{bmatrix}^T$$

$$A_d=\mathrm{diag}\{A_1,\ A_2,\cdots,\ A_N\}$$

$$B_h=\mathrm{diag}\{B_1d_1H_1C_1,\ B_2d_2H_2C_2,\cdots,B_Nd_NH_NC_N\}$$

$$B_a=\begin{bmatrix}c_{11}B_1H_1C_1 & c_{12}B_1H_1C_2 & \cdots & c_{1N}B_1H_1C_N\\ c_{21}B_2H_2C_1 & c_{22}B_2H_2C_2 & \cdots & c_{21}B_2H_2C_N\\ \vdots & \vdots & \ddots & \vdots\\ c_{N1}B_NH_NC_1 & c_{N1}B_NH_NC_2 & \cdots & c_{NN}B_NH_NC_N\end{bmatrix}$$

$$B_g=\mathrm{diag}\{B_1G_1,\ B_2G_2,\cdots,\ B_NG_N\}$$

$$V(t)=\begin{bmatrix}v_1^T(t), v_2^T(t),\cdots,v_N^T(t)\end{bmatrix}^T$$

$$v_i(t)=B_iH_i\sum_{j=1}^{N}c_{ij}C_jA_j^{\tau_{ij}-1}(e_j(t-\tau_{ij})-e_j(t-\tau_{ij}+1))$$
$$+d_iB_iH_iC_iA_i^{\tau_i-1}(e_i(t-\tau_i+1)-e_i(t-\tau_i)),\forall i\in\mathbb{N}$$

From (5), (6) and (17), $\Delta z_i(t+1)$ can be derived by

$$\Delta z_1(t+1)=\Delta z_1(t)+C_1\Delta x_1(t)-C_1A_1^{\tau_1-1}\Delta e_1(t-\tau_1+1) \quad (21)$$

$$\begin{aligned}\Delta z_i(t+1)&=\Delta z_i(t)+\Delta\hat{y}_i(t|t-\tau_i)-\Delta\hat{y}_1(t|t-\tau_{i1})\\&=\Delta z_i(t)+C_i\Delta x_i(t)-C_1\Delta x_1(t)\\&\quad+C_1A_1^{\tau_{i1}-1}\Delta e_1(t-\tau_{i1}+1)-C_iA_i^{\tau_i-1}\Delta e_i(t-\tau_i+1)\end{aligned} \quad (22)$$

$\forall i\in\mathbb{N}-\{1\}$ .Then, the compact forms for (21) and (22) can be described by

$$\Delta Z(t+1)=(C_d-J_N\otimes C_1)\Delta X(t)+\Delta Z(t)+W(t) \quad (23)$$

where

$$C_d=\text{diag}\{C_1,C_2,\cdots,C_N\}$$

$$J_N=\left[[0,1,\cdots,1]^T\quad 0_{n\times(n-1)}\right]$$

$$W(t)=\left[w_1^T(t),w_2^T(t),\cdots,w_N^T(t)\right]^T$$

$$w_1(t)=C_1A_1^{\tau_1-1}(e_1(t-\tau_1)-e_1(t-\tau_1+1))$$

$$w_i(t)=C_1A_1^{\tau_{i1}-1}(e_1(t-\tau_{i1}+1)-e_1(t-\tau_{i1}))$$
$$-C_iA_i^{\tau_i-1}(e_i(t-\tau_i+1)-e_i(t-\tau_i)),\quad\forall i\in\mathbb{N}-\{1\}$$

and ⊗ represents the Kronecker product of matrices.

From (12), it is clear that

$$\varepsilon(t+1)=A_e\varepsilon(t) \quad (24)$$

where

$$\varepsilon(t)=\left[e_1^T(t-\tau_1+1),e_1^T(t-\tau_1),e_2^T(t-\tau_2+1),e_2^T(t-\tau_2),\cdots,e_N^T(t-\tau_N)\right]^T$$

$$A_e=\text{diag}\{A_1-F_1C_1,A_1-F_1C_1,A_2-F_2C_2,A_2-F_2C_2,$$
$$\cdots,A_N-F_NC_N,A_N-F_NC_N\}$$

Thus, combining (20), (23) and (24), the closed-loop NMAS with the cloud predictive control scheme can be expressed in the following compact form:

$$\begin{bmatrix}\delta(t+1)\\\varepsilon(t+1)\end{bmatrix}=\begin{bmatrix}A_{xz}&R_e\\\mathbf{0}&A_e\end{bmatrix}\begin{bmatrix}\delta(t)\\\varepsilon(t)\end{bmatrix} \quad (25)$$

where

$$A_{xz}=\begin{bmatrix}A_d-B_h+B_a&B_g\\C_d-J_N\otimes C_1&\mathbf{I}\end{bmatrix} \quad (26)$$

$\delta(t)=\left[\Delta X^T(t)\quad\Delta Z^T(t)\right]^T$, $R_e\varepsilon(t)=\left[V^T(t)\quad W^T(t)\right]$, ***0*** and ***I*** denote a zero matrix and identity matrix with an appropriate dimension, respectively.

Theorem 1: Networked multi-agent system (1) with the cloud predictive control protocol (7) is stable and achieves consensus if and only if all the matrices A_{xz} and A_i -F_iC_i, $\forall i\in\mathbb{N}$, are Schur stable.

Proof: It is clear from the above that the closed-loop NMAS with the cloud predictive control scheme is equivalent to (25). It means that the necessary and sufficient stability conditions of the closed-loop networked multi-agent control system are that all the matrices A_{xz} and A_e are Schur stable, which implies that matrices A_{xz} and A_i -F_iC_i , $\forall i\in\mathbb{N}$, are Schur stable.

If system (25) is stable, it means $\Delta\delta(t)\to 0$ and $\Delta\varepsilon(t)\to 0$, as $t\to\infty$, which implies that $\Delta x_i(t)\to 0$, $\Delta z_i(t)\to 0$ and $e_i(t)\to 0$ as $t\to\infty$, $\forall i\in\mathbb{N}$.

For $e_i(t)\to 0$ as $t\to\infty$, it can be resulted from (12) and (4) that $\hat{x}_i(t|t-\tau_i)\to x_i(t)$ and $\hat{y}_i(t|t-\tau_i)\to y_i(t)$ as $t\to\infty$. Rewriting (5) ad (6) can lead to

$$\Delta z_1(t+1)=\hat{y}_1(t|t-\tau_1)-r_0$$
$$\Delta z_i(t+1)=\hat{y}_i(t|t-\tau_i)-\hat{y}_1(t|t-\tau_{i1}),\forall i\in\mathbb{N}-\{1\}$$

For $\Delta z_i(t)\to 0$ as $t\to\infty$, $\forall i\in\mathbb{N}$, it implies from the above that $\hat{y}_1(t|t-\tau_1)\to r_0$ as $t\to\infty$ and $\hat{y}_i(t|t-\tau_i)\to$ $\hat{y}_i(t|t-\tau_i)\to\hat{y}_1(t|t-\tau_{i1})\to y_1(t)$ as $t\to\infty$, $\forall i\in\mathbb{N}-\{1\}$.

Clearly, it can be concluded from the above analysis that $y_1(t)\to r_0$ as $t\to\infty$ and $y_i(t)\to y_1(t)$ as $t\to\infty$, $\forall i\in\mathbb{N}-\{1\}$. It means that the two conditions of Definition 1 are satisfied. Therefore, the closed-loop NMAS with the cloud predictive control scheme is not only stable but also achieves the consensus.

Remark 1: The above theorem shows that the cloud predictive control scheme can achive the consensus of and gurranttee the stability of the closed-loop networked multi-agent control systems simutanously. This is a significant achievement in the design and analysis of networked multi-agent control systems.

IV. AN EXAMPLE

A multi-agent system with three agents is considered as an example to illustrate the performance of the cloud predictive control scheme of NMAS. The matrices of the three agents with different dynamics are given as follows:

$$A_1=\begin{bmatrix}1.7&-1.3\\1.6&-1.8\end{bmatrix},B_1=\begin{bmatrix}1.0\\2.0\end{bmatrix},C_1=\begin{bmatrix}1.0\\0.3\end{bmatrix}^T$$
$$A_2=\begin{bmatrix}1.8&-1.4\\1.8&-1.9\end{bmatrix},B_2=\begin{bmatrix}1.7\\3.4\end{bmatrix},C_2=\begin{bmatrix}0.7\\0.2\end{bmatrix}^T$$
$$A_3=\begin{bmatrix}1.4&-1.1\\1.3&-1.5\end{bmatrix},B_3=\begin{bmatrix}0.8\\1.6\end{bmatrix},C_3=\begin{bmatrix}1.1\\0.4\end{bmatrix}^T$$

Actually, agent 1 is unstable, agent 2 critically stable and agent 3 stable. The three agents are controlled with a cloud controller via a network. The network delays between the networked agent and the cloud computing system are set to be $a_1=3$, $a_2=2$, $a_3=3$, $s_1=2$, $s_2=4$, $s_3=1$. It can be obtained from (8) that $c_{11}=1$, $c_{12}=0$, $c_{13}=1$, $c_{21}=1$, $c_{22}=1$, $c_{23}=1$, $c_{31}=1$, $c_{32}=0$, $c_{33}=1$. In this example, the initial conditions of all the agent states, control inputs and observer states are zero. The desired reference input r_0 for the first agent is 1 for t∈[0, 100) and t∈[200, 300) and is 0 for t∈[100, 200).

The gains G_i and H_i in the control protocol (7) are designed by the eigenstructure assignment method [25], which gives one of possible solutions that make matrix A_{xz} be stable, *i.e.,*

$$G_1 = -0.16,\ G_2 = -0.18,\ G_3 = -0.14$$
$$H_1 = -0.12,\ H_2 = -0.10,\ H_3 = -0.14$$

To have a good convergence rate for observer states, the poles of the observers should be located in a desired area which is normally near to the coordinate origin for discrete-time systems. Therefore, the observer gain matrices of the three agents are designed to be

$$F_1 = \begin{bmatrix} -0.4483 \\ -1.1724 \end{bmatrix},\ F_2 = \begin{bmatrix} -0.6803 \\ -1.6191 \end{bmatrix},\ F_3 = \begin{bmatrix} -0.3908 \\ -0.9254 \end{bmatrix}$$

to assign the desired poles of the three observers to 0.3 and 0.4 using the pole placement function called 'place' in MATLAB, which give desired transient responses for the observers.

Two cases are considered in this example. One is there are no network delays and the other is there are network delays.

Case 1: Without network delays

For this case, the network delays in the networked three-agent control system are zero (*i.e.,* $a_i = s_i = 0$, for $i=1,2,3$) and the cloud control protocols (7) become

$$u_i(t) = G_i z_i(t) + H_i \sum_{j=1}^{3} c_{ij}(y_j(t) - y_i(t)),\ \text{for } i = 1, 2, 3$$

where $z_1(t+1) = z_1(t) + y_1(t) - r_0$ and $z_i(t+1) = z_i(t) + y_i(t) - y_1(t)$, for $i=2,\ 3$. The outputs $y_i(t)$ of the three closed-loop cloud control agents are shown in Figure 3. Clearly, the three-agents are stable and achieve consensus very well.

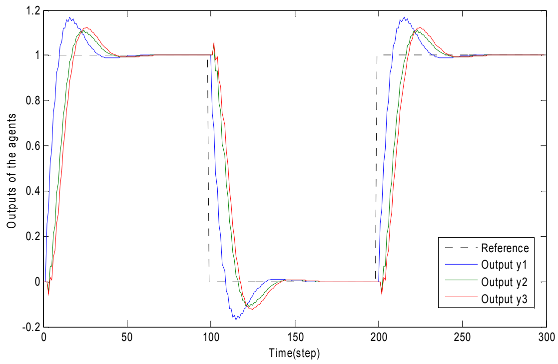

Figure 3 The outputs of the closed-loop cloud control agents

Case 2: With network delays

For this case, there exist network delays in the networked three-agent control system, and the cloud predictive control protocols (7) is utilised, *i.e.,*

$$u_i(t) = G_i z_i(t) + H_i \sum_{j=1}^{3} c_{ij}(\hat{y}_j(t|t-a_i-s_j) - \hat{y}_i(t|t-a_i-s_i)),\ \text{for } i = 1, 2, 3$$

where $z_i(t+1) = z_i(t) + \hat{y}_i(t|t-a_i-s_i) - \hat{y}_1(t|t-a_i-s_1)$, for $i=2,\ 3$ and $z_1(t+1) = z_1(t) + \hat{y}_1(t|t-a_1-s_1) - r_0$. The outputs $y_i(t)$ of the three closed-loop cloud predictive control agents are shown in Figure 4. It can be noted from the simulation results that the stability and consensus performance of the three closed-loop cloud predictive control agents is very similar to the one of Case 1.

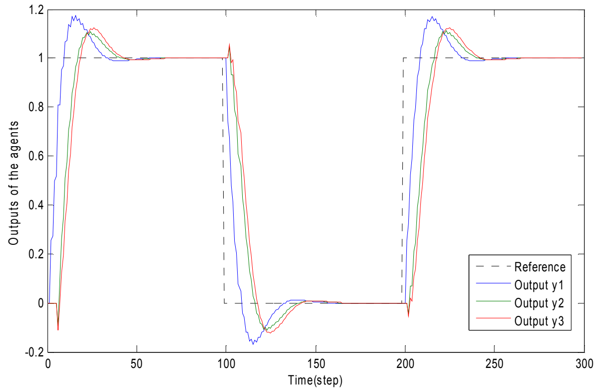

Figure 4 The outputs of the closed-loop cloud predictive control agents

To compare the performance of Case 2 and Case 1 precisely, the output errors between Case 2 and Case 1 are given in Figure 5. It is clear that there exist the large output errors for 0⩽t⩽50 because it takes some time for the state observers of the three agents to converge. After those state observers converge for t>50, the output errors between Case 2 and Case 1 are nearly zero. It implies that the performance of the networked multi-agent cloud predictive control system with network delays is almost the same as the one of the system without network delays. This shows the cloud predictive control scheme proposed in this paper actively compensates for network delays well.

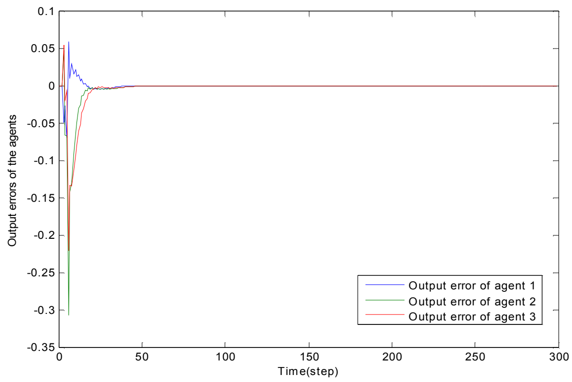

Figure 5 The output errors between Case 2 and Case 1

To illustrate disturbance rejection of the cloud predictive control scheme, different random disturbances $d_i(t) \in \Re^{2\times1}$, for $i=1,2,3$, within amplitude ±0.005 were added to the states of all the individual agents. This means the state equations are described by $x_i(t+1) = A_i x_i(t) + B_i u_i(t) + d_i(t)$, for $i=1,2,3$. The output responses of the three closed-loop cloud predictive control agents are given in Fig. 6. It shows that the proposed cloud predictive control scheme is of disturbance rejection capability.

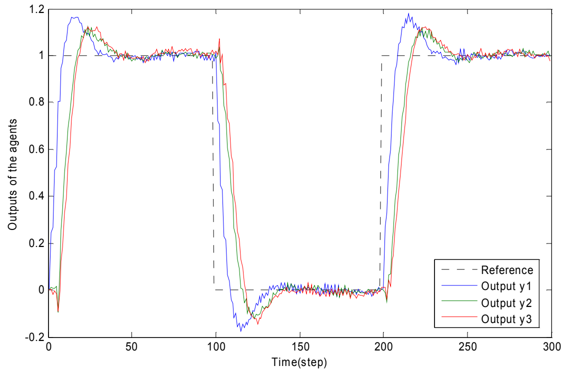

Figure 6 The outputs of the closed-loop cloud predictive control agents with disturbances

V. CONCLUSIONS

In this paper, a cloud predictive control scheme for networked multi-agent systems via cloud computing has been proposed to achieve both consensus and stability simultaneously and to compensate for communication delays actively. The design and analysis of networked multi-agent cloud predictive control systems have been studied. It has detailed what the cloud predictive control architecture looks like, how to design the cloud predictive controllers and how to implement the controller via a cloud computing systems. The necessary and sufficient conditions of stability and consensus of the closed-loop networked multi-agent cloud predictive control system have been derived. A simulated example has successfully demonstrated the stability, consensus and control performance of the proposed cloud predictive control scheme for networked multi-agent systems.

The cloud predictive control is a new idea and has not been studied before. There exist many potential applications of networked multi-agent cloud predictive control systems in practice, for example, electronic power control of the production and distribution of electricity in smart grid, formation control of GPS satellites. Although the paper has provided the details of the proposed cloud predictive control scheme and the simulation example demonstrates how it performs, it is still at the theoretical research stage and there is much work to do for general applications.

REFERENCES

[1] Y. Xia, M. Fu and G.P. Liu, *Analysis and Synthesis of Networked Control Systems.* Heidelberg, Germany: Springer-Verlag, 2011.

[2] L. Busoniu, R. Babuska and B.D. Schutter, "A comprehensive survey of multi-agent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics. Part C: Applications and Reviews*, vol. 38, no. 2, pp. 156–172, Feb. 2008.

[3] Y. Xu, Z. Li, "Distributed optimal resource management based on the consensus algorithm in a microgrid," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 2584 - 2592, Feb. 2015.

[4] W. Li, "Notion of control-law module and modular framework of cooperative transportation using multiple nonholonomic robotic agents with physical rigid-formation-motion constraints," *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1242 - 1248, May 2016.

[5] Z. Sunberg, S. Chakravorty, R. S. Erwin, "Information space receding horizon control for multisensor tasking problems," *IEEE Transactions on Cybernetics*, vol. 46, no. 6, pp. 1325 - 1336, June 2016.

[6] J.P. Desai, J.P. Ostrowski and V. Kumar, "Modeling and control of formation of nonholonomic mobile robust." *IEEE Trans. on Robotics and Automation*, vol. 17, no. 6, pp. 905–908, June 2001.

[7] W. Ren, R.W. Beard, "Formation feedback control for multiple spacecraft via virtual structures," *IEE Proc. Control Theory and Applications*, vol. 151, no. 3, pp. 357–368, Mar. 2004.

[8] M. Porfiri, D.G. Roberson and D.J. Stilwell, "Tracking and formation control of multiple autonomous agents: a two-level consensus approach," *Automatica*, vol. 43, no. 8, pp. 1318 – 1328, Aug. 2007.

[9] J. Cortes, "Global and robust formation-shape stabilization of relative sensing networks," *Automatica*, vol. 45, no. 12, pp. 2754-2762, Dec. 2009.

[10] W.Y. Jing, W.W. Yan and W. X. Jiang, "Consensus in markovian jump second-order multi-agent systems with random communication delay," *IET Control Theory & Applications*, vol. 8, no. 16, pp. 1666–1675, 2014.

[11] Y.G. Hong, J.P. Hu and L.X. Gao, "Tracking control for multi-agent consensus with an active leader and variable topology," *Automatica*, vol. 42, no. 7, pp. 1177‑1182, Jul. 2006.

[12] F. Xiao and L. Wang, "Asynchronous consensus in continuous-time multiagent systems with switching topology and time-varying delays," *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1804 – 1816, Aug. 2008.

[13] R. Olfati-Saber and R.M. Murray, "Consensus problems in networks with switching topology and time-delays," *IEEE Transactions on Automatic Control*, 49(9): 1520 – 1533, 2004.

[14] H. J. Savino, C. R. P. dos Santos, F. O. Souza, L. C. A. Pimenta, M. de Oliveira, R. M. Palhares, "Conditions for consensus of multi-agent systems with time-delays and uncertain switching topology," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 2, pp. 1258 – 1267, Feb. 2016.

[15] P.A. Bliman and G.F. Trecate, "Average consensus problems in networks of agents with delayed communications," *Automatica*, vol. 44, no.8, pp. 1985 – 1995, Aug. 2008.

[16] C. N. Hadjicostis and T. Charalambous, "Average consensus in the presence of delays in directed graph topologies," *IEEE Trans. on Automatic Control*, vol. 59, no. 3, pp. 763–768, Mar. 2014.

[17] H. Zhang, T. Feng, G.H. Yang and H. Liang, "Distributed cooperative optimal control for multiagent systems on directed graphs: an inverse optimal approach," *IEEE Trans. Cybernetics*, vol. 45, no. 7, pp. 1315 – 1326, Jul. 2015.

[18] A.L. Barab´asi, R. Albert, et al., "Power-law distribution of the world wide web," *Science*, vol. 287, pp. 2115, Mar. 2000.

[19] J. Zhao, D. J. Hill and T. Liu, "Synchronization of dynamical networks with nonidentical nodes: criteria and control," *IEEE Tran. Circuits and Syst. I*, vol. 58, no. 3, pp. 584 – 594, Mar. 2011.

[20] W.S. Zhong, G.P. Liu and C. Thomas, "Global bounded consensus of multi-agent systems with non-identical nodes and time delays," *IEEE Transactions on Systems, Man, and Cybernetics: Part B*, vol. 42, no. 5, pp. 1480-1488, May 2012.

[21] G.P. Liu, Y. Xia, D. Rees and W.S. Hu, "Design and stability criteria of networked predictive control systems with random network delay in the feedback channel," *IEEE Transactions on Systems, Man and Cybernetics Part C*, vol. 37, no. 2, pp. 173-184, Feb. 2007.

[22] Y. Xia, "Cloud control systems," *IEEE/CAA Journal of Automatica Sinica,* vol. 2, no. 2, pp. 134-142, Feb. 2015.

[23] Y. Xia, Y. Qin, D. Zhai and S. Chai, "Further results on cloud control systems," *Science China Information Sciences,* vol. 59, no.7, pp. 1-5, Jul. 2016.

[24] Z.H. Pang, and G.P. Liu, "Design and practical implementation of secure networked predictive control systems," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 5, pp.1334-1342, May 2012.

[25] G.P. Liu and R.J. Patton, *Eigenstructure Assignment for Control System Design.* New York, NY, USA: Wiley, 1998

Guo-Ping Liu (M'97–SM'99–F'11) received the B.Eng. and M.Eng. degrees in automation from the Central South University, Changsha, China, in 1982 and 1985, respectively, and the Ph.D. degree in control engineering from the University of Manchester, Manchester, U.K., in 1992.

He has been a professor with the University of South Wales, Cardiff, U.K., since 2004, a Hundred-Talent Program visiting professor with the Chinese Academy of Sciences, Beijing, China, since 2001, and a Changjiang Scholar visiting professor with the Harbin Institute of Technology, Harbin, China, since 2008. He is a chair of control engineering with the University of South Wales. His current research interests include networked multi-agent control systems, nonlinear system identification and control, advanced control of industrial systems, and multiobjective optimization and control. He has authored/co-authored over 200 journal papers and 8 books on control systems.

Prof. Liu is an editor-in-chief of the International Journal of Automation and Computing and IET Fellow.