Design and Practical Implementation of External Consensus Protocol for Networked Multi-agent System with Communication Delays

Nurul Adilla Mohd Subha and Guo-Ping Liu, *Fellow, IEEE*

***Abstract*—This paper discusses the design and practical implementation of solving an external consensus problem for inhomogeneous networked multi-agent system (NMAS) with constant network delay in the output feedback. Based on the recursive equation, a novel prediction strategy is proposed using the transfer function form to overcome the effects of the network delay. By considering a single-input single-output (SISO) linear system for each agent, the proposed strategy is simulated, demonstrated and validated through a test-rig application. Concurrently, the stability of the control scheme and the criterion to determine the appropriate coupling gains value for the proposed consensus protocol are also being studied.**

***Index Terms*— consensus, networked multi-agent system, recursive prediction control, network delay, stability, coupling.**

I. INTRODUCTION

THE problem of consensus in multi-agent system through network communication is gaining attention in research domain as indicated by high amount of research output around the world. Consensus problem in networked multi-agent system (NMAS) can be described as a group of subsystems or agents working together under a shared network enabling to converge to a common consensus value upon request. The consensus value may represent any physical quantities such as angle, temperature, velocity, level and etc. There are a few significant findings in earlier research works such as in [1, 2, 3, 4] that have provided solid underpinning for this field and formed a basis for this research work. In those papers, theoretical frameworks of the consensus and its compact analysis have been developed to solve the consensus problem in multi-agent system on various aspects of the NMAS design. However, the studies are only considering and analyzing the average consensus problem with homogeneous NMAS in the context of single-integrator agents.

In average consensus protocol, consensus value is calculated based on the initial value of every agent in NMAS. To solve the average consensus problem, various methods of analysis, implementation and consideration have been developed [5]-[12]. Besides average consensus, max-min consensus and consensus function are also depending on the initial value information from NMAS's agents. Detailed discussion on max-min consensus and consensus function can be found in [13, 14, 15]. Rigorous numerical examples and simulation results have been illustrated to prove the effectiveness of initial value based protocol design. However, this type of consensus protocols can only be applied and limited to NMAS with non-zero initial value agents.

Consensus protocol that is independent from the initial value of NMAS's agents is called external consensus protocol. The consensus value in external consensus protocol is set by the external reference input given to one or more agents in NMAS. This type of consensus protocol is described in [16, 17, 18, 19]. In [17], an improvement work of [16] on the whole NMAS stability has been discussed and proved by demonstrating the proposed algorithm with two continuous inhomogeneous systems using Proportional-Integral (PI) controller. However, the stability derived is rigid and completely depends on the characteristics of the external reference input.

In the past few years, many efforts have been devoted to expand NMAS research area towards higher reliability in practical implementation. In an attempt to simulate near to real NMAS application scenario, system model structure with higher degree of complexity than single-integrator is studied. Typically, the system model is developed from identification method or linear formula based on available system parameters. In [20, 21, 22, 23], control researchers have put a considerable interest on linear NMAS with *n*-th order system. In Wang [20], a strong and a weak-output-feedback of consensus protocol are developed for a linear multi-agent system depending on the

Manuscript received October , 2013. Manuscript revised April 24, 2014. Accepted for publication June 22, 2014. This work was supported in part by the National Natural Science Foundation of China under Grants 61273104 and 61333003.

Nurul Adilla Mohd Subha is with the School of Engineering, University of South Wales, CF37 1DL, UK (e-mail: nurul.mohdsubha@southwales.ac.uk) on leave from the Department of Control and Mechatronics Engineering, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Skudai, Malaysia (email: adilla@fke.utm.my).

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

presence of noise. In [21, 22], the consensus of homogeneous linear multi-agent system for static and dynamic output error feedback have been proposed for a fixed and switching topology cases. Constrained and robust consensus problems are solved in [23] using a linear quadratic regulator and a predictive controller. However, the effects of network delay are not considered in [16]-[23]. Network delay is an unavoidable weakness in NMAS which normally exists in a network based communication. This is due to a limited bandwidth of communication channel and the finite transmission speed. The communication delay can degrade the stability and the performance of the control system significantly.

With this background, this paper addresses the external consensus problem for inhomogeneous NMAS with the presence of a uniform network delay. An external reference input is given only to one agent in NMAS. In order to compensate the network delay, a novel prediction algorithm for NMAS is presented based on a predictive controller proposed in [24] and inspired by the results in [25]. The objective of this paper is to describe the experimental and simulation findings in solving the proposed external consensus problem for second-order inhomogeneous NMAS with constant network delay via prediction algorithm. Further to that, stability analysis and a criterion in choosing the appropriate coupling gains between NMAS agents are also presented. In the previous literature, most of the research outcomes have been demonstrated solely by means of simulation [26, 27, 28, 29]. This paper however has produced test-rig experimental results along with the simulation on the proposed method to elevate level of practicality of the developed theoretical solution. Through the test-rig experiments, a clear practical performance for a developed NMAS theory is illustrated.

II. Consensus of NMAS in a Fixed Communication Topology

The interaction and communication between agents in NMAS can be modeled by an undirected graph $G=(V,E,A)$ for order *n* with the set of nodes or agents $V=\{v_1,v_2,...,v_n\}$ and edges $E\subseteq V\times V$. An edge from *i* to *j* is denoted by $e=(v_i,v_j)$ indicate that agent *j* can receive information from agent *i* and vice versa. In undirected graph, an edge from *i* to *j* and *j* to *i* has no exact direction which has positive unweighted adjacency matrix $a_{ij}=a_{ji}=1$ for all *i, j*. No self-loop is allowed, hence $a_{ii}=a_{jj}=0$. The set of neighbor agent *i* is denoted by $N_i=\{j\in V:(j,i)\in E\}$. The Laplacian matrix *L* with respect to undirected graph *G* can be simply obtained as

$$L=[l_{ij}]_{n\times n} \qquad \text{where} \qquad l_{ij}=\begin{cases}|N_i|, & i=j\\ -1, & N_i\\ 0, & otherwise\end{cases}$$

However, in this work, the Laplacian matrix *L* has non-zero elements because every agent is interconnected to one another. Obviously, all the row-sums of *L* are zero. Therefore, *L* always has a zero eigenvalue $\lambda_1=0$ and second smallest eigenvalue of *L* is $\lambda_2>0$ if and only if *G* is connected and has a spanning tree [30]. Thus, the eigenvalues of *L* can be ordered as $0=\lambda_1\le\lambda_2\le\lambda_3\le...\le\lambda_n$.

Consider inhomogeneous NMAS composed of *n* single-input single-output (SISO) agents where the dynamic of each agent is described in the polynomial transfer function form

$$A_i(z^{-1})y_i(k+1)=B_i(z^{-1})u_i(k) \tag{1}$$

where $y_i(k)$ and $u_i(k)$ are the output and control input of agent i respectively for $i=1,2,...,n$ and n is the number of agents in NMAS. The polynomial of $A_i(z^{-1})$ and $B_i(z^{-1})$ for model agent *i* are represented as

$$P_i(z^{-1})=\frac{B_i(z^{-1})}{A_i(z^{-1})}=\frac{b_{i0}+b_{i1}z^{-1}+..+b_{im_{bi}}z^{-m_{bi}}}{1+a_{i1}z^{-1}+..+a_{in_{ai}}z^{-n_{ai}}} \tag{2}$$

where n_{ai} and m_{bi} are polynomial order of $A_i(z^{-1})$ and $B_i(z^{-1})$ respectively with $n_{ai}\ge m_{bi}$. The transfer function of virtual controller for each NMAS agents is designed as

$$G_i(z^{-1})=\frac{D_i(z^{-1})}{C_i(z^{-1})}=\frac{d_{i0}+d_{i1}z^{-1}+..+d_{in_{di}}z^{-n_{di}}}{1+c_{i1}z^{-1}+..+c_{in_{ci}}z^{-n_{ci}}} \tag{3}$$

with n_{ci} and n_{di} are polynomial order of $C_i(z^{-1})$ and $D_i(z^{-1})$ respectively where $n_{ci}\ge n_{di}$. The virtual controller in (3) is designed without considering the network delay in order to achieve the desired performance of NMAS. Classical or advanced control design methods can be used for this purpose.

For the simplicity of the consensus analysis, the following assumptions can reasonably be made:

i. Each agent *i* can receive information from agent $j\in i\cup N_i$.
ii. Network delay τ is assumed to be a constant with known positive integer which represents the upper bound of the measured time-varying network delay [31].

The external consensus protocol for NMAS with uniform output network delay τ but without network delay compensation can be described as follows:

$$u_i(k)=\begin{cases}G_i(z^{-1})(R(k)-(y_i(k)+K_i\sum_{j=2}^{n}(y_i(k-\tau)-y_j(k-\tau)))), & for \quad i=1\\ -G_i(z^{-1})(K_i\sum_{\substack{j=1\\ j\neq i}}^{n}(y_i(k-\tau)-y_j(k-\tau))), & for \quad i=2,3,..,n\end{cases} \tag{4}$$

where $R(k)\in\Re$ is the external reference input, $y_i(k-\tau)\in\Re$ is the output of agent i for i =1,2,3,…,*n* at time $k-\tau$, $y_j(k-\tau)\in\Re$ is the output of neighboring agent *i* N_i at time $k-\tau$, $u_i(k)\in\Re$ is the control input of agent *i*, and $G_i(z^{-1})$ is the virtual controller of agent *i*. Following the consensus protocol in (4), $\tau>0$ is a communication delay for all outputs information exchanged between agents. Only agent *i*=1 is connected to an external reference $R(k)$ and $R(k)$ = 0 for i =2,3,…,n. The coupling gain K_i is a positive constant used as tunable parameter that strengthen or loosen coupling between the individual agents and to adjust the convergence speed (settling time) of linear NMAS [32-34]. Equivalently, (4) can be simplified as

$$u_i(k) = G_i(z^{-1})U_{Gi} \tag{5}$$

$$U_{Gi} = \begin{cases} R(k) - y_{ei}(k-\tau), & for \quad i=1 \\ -y_{ei}(k-\tau), & for \quad i=2,3,..,n \end{cases} \tag{6}$$

$$y_{ei}(k-\tau) = \begin{cases} y_i(k) + K_i \sum_{j=2}^{n} (y_i(k-\tau) - y_j(k-\tau)), & for \quad i=1 \\ K_i \sum_{\substack{j=1 \\ j\neq i}}^{n} (y_i(k-\tau) - y_j(k-\tau)), & for \quad i=2,3,..,n \end{cases} \tag{7}$$

where U_{Gi} is a signal that feeds to the corresponding controller $G_i(z^{-1})$. Due to the existence of network delay in (4) to (7), NMAS is unable to achieve external consensus and produces oscillating output. For that reason, the recursive prediction controller (RPC) is proposed to predict the output of NMAS agents in order to solve the external consensus problem for NMAS with constant output network delay.

III. Design of Networked Multi-agent Predictive Control

In this section, linear inhomogeneous NMAS with constant output network delay is considered. To overcome the effects of the output network delay, the RPC is designed by positioning the output predictor at feedback channel for every agent. Since the constant output network delay is constant and known, the prediction sequence is computed within agent i and then transmitted to the other agents. Based on the agent's model $P_i(z^{-1})$ and controller's model $G_i(z^{-1})$, output predictors are designed to generate the output predictions sequence recursively from time $k-\tau+1$ to k for agent $i, j = 1,2,3,..,n$ where $i \neq j, j \in N_i$. The one-step ahead prediction sequence of agent's i output at time $k-\tau$ is constructed as follows:

$$y_i(k-\tau+1 \mid k-\tau) = -\sum_{f=1}^{n_{ai}} a_{if} y_i(k-f-\tau+1) + b_{i0} u_i(k-\tau \mid k-\tau) + \sum_{f=1}^{m_{bi}} b_{if} u_i(k-f-\tau) \tag{8}$$

$$u_i(k-\tau \mid k-\tau) = -\sum_{f=1}^{n_{ci}} c_{if} u_i(k-f-\tau) + D_i(z^{-1}) r_i(k-\tau) - \sum_{f=0}^{n_{di}} d_{if} y_{ei}(k-f-\tau) \tag{9}$$

where $u_i(k-\tau \mid k-\tau) = u_i(k-\tau)$. Based on (8), one-step ahead prediction of $y_{ei}(k-\tau+1 \mid k-\tau)$ is computed as follows:

$$y_{ei}(k-\tau+1 \mid k-\tau) = \begin{cases} y_i(k-\tau+1 \mid k-\tau) + K_i \sum_{j=2}^{n} (y_i(k-\tau+1 \mid k-\tau) - y_j(k-\tau+1 \mid k-\tau)), & for \quad i=1 \\ K_i \sum_{\substack{j=1 \\ j\neq i}}^{n} (y_i(k-\tau+1 \mid k-\tau) - y_j(k-\tau+1 \mid k-\tau)), & for \quad i=2,3,..,n \end{cases} \tag{10}$$

where $y_j(k-\tau+1 \mid k-\tau)$ is obtained from the received output prediction sequence at time $k-\tau$ from neighbor's agent i. The received output prediction sequence for length $\tau+1$ can be represented as follows:

$$\begin{bmatrix} y_j(k-\tau \mid k-\tau) \\ y_j(k-\tau+1 \mid k-\tau) \\ \vdots \\ \vdots \\ y_j(k \mid k-\tau) \end{bmatrix} \tag{11}$$

Thus, one-step ahead control input prediction sequence of agent i at time $k-\tau$ can be obtained as

$$u_i(k-\tau+1 \mid k-\tau) = -c_{i1} u_i(k-\tau \mid k-\tau) - \sum_{f=2}^{n_{ci}} c_{if} u_i(k-f-\tau+1) + D_i(z^{-1}) r_i(k-\tau+1) - d_{i0} y_{ei}(k-\tau+1 \mid k-\tau) - \sum_{f=1}^{n_{di}} d_{if} y_{ei}(k-f-\tau+1) \tag{12}$$

Since tracking signal $r_i(k) = R(k)$ for i=1 and $r_i(k) = 0$ for $i = 2,3,...,n$, (12) can be expanded as

$$\hat{u}_i(k-\tau+1 \mid k-\tau) = -c_{i1} u_i(k-\tau \mid k-\tau) - \sum_{f=2}^{n_{ci}} c_{if} u_i(k-f-\tau+1) + D_i(z^{-1}) R(k-\tau+1) - d_{i0} y_{ei}(k-\tau+1 \mid k-\tau) - \sum_{f=1}^{n_{di}} d_{if} y_{ei}(k-f-\tau+1), \quad for \quad i=1 \tag{13}$$

$$\hat{u}_i(k-\tau+1 \mid k-\tau) = -c_{i1} u_i(k-\tau \mid k-\tau) - \sum_{f=2}^{n_{ci}} c_{if} u_i(k-f-\tau+1) - d_{i0} y_{ei}(k-\tau+1 \mid k-\tau) - \sum_{f=1}^{n_{di}} d_{if} y_{ei}(k-f-\tau+1), \quad for \quad i=2,3,...,n$$

Corresponding to computation in (8) and (13), the agent's i output and its control input predictions from time $k-\tau+1$ to k are generated for $p \in [1, \tau]$ which results in

$$y_i(k-\tau+p\mid k-\tau)=-\sum_{f=1}^{\min\{n_{ai},p-1\}}a_{if}y_i(k-f-\tau+p\mid k-\tau) \quad (14)$$
$$-\sum_{f=p}^{n_{ai}}a_{if}y_i(k-f-\tau+p)+\sum_{f=0}^{\min\{m_{bi},p-2\}}b_{if}\,\hat{u}_i(k-f-1-\tau+p\mid k-\tau)$$
$$+\sum_{f=p-1}^{m_{bi}}b_{if}u_i(k-f-1-\tau+p)$$

$$\hat{u}_i(k-\tau+p\mid k-\tau)= \quad (15)$$
$$\begin{cases} -\sum_{f=1}^{\min\{n_{ci},p-1\}}c_{if}\hat{u}_i(k-f-\tau+p\mid k-\tau)-\sum_{f=p}^{n_{ci}}c_{if}u_i(k-f-\tau+p) \\ +D_i(z^{-1})R(k-\tau+p)-\sum_{f=0}^{\min\{n_{di},p-1\}}d_{if}y_{ei}(k-f-\tau+p\mid k-\tau) \\ -\sum_{f=p}^{n_{di}}d_{if}y_{ei}(k-f-\tau+p), \quad for \quad i=1 \\ -\sum_{f=1}^{\min\{n_{ci},p-1\}}c_{if}\hat{u}_i(k-f-\tau+p\mid k-\tau)-\sum_{f=p}^{n_{ci}}c_{if}u_i(k-f-\tau+p)- \\ -\sum_{f=0}^{\min\{n_{di},p-1\}}d_{if}y_{ei}(k-f-\tau+p\mid k-\tau) \\ -\sum_{f=p}^{n_{di}}d_{if}y_{ei}(k-f-\tau+p), \quad for \quad i=2,3,...n \end{cases}$$

From (14)-(15), it can be seen that all equations have output sequence and control input sequence in two separated parts. The first part is prediction sequence while the second part is current signal sequence that available at time $k-\tau$. Based on above equations, in order to compensate the constant output network delay τ between agents, the output prediction of agent *i* at time *k* is selected to be $y_i(k|k-\tau)$ which can be simplified as $\bar{y}_i(k)=y_i(k|k-\tau)$. This information is sent through network and become $\bar{y}_j(k)$ as this information is received by other agent(s). At time *k*, the output controller for agent *i* =1, 2, 3,…,*n* with delay compensation can be obtained as

$$u_i(k)=G_i(z^{-1})\breve{U}_{Gi} \quad (16)$$

$$\breve{U}_{Gi}=\begin{cases} R(k)-y_{ei}(k), & for \quad i=1 \\ -y_{ei}(k), & for \quad i=2,3,..,n \end{cases} \quad (17)$$

$$y_{ei}(k)=\begin{cases} y_i(k)+K_i\sum_{j=2}^{n}(y_i(k)-\bar{y}_j(k)), & for \quad i=1 \\ K_i\sum_{\substack{j=1\\ j\neq i}}^{n}(y_i(k)-\bar{y}_j(k)), & for \quad i=2,3,...,n \end{cases} \quad (18)$$

where a signal that feeds to the corresponding controller with delay compensation is denotes as $\breve{U}_{Gi}$. From (16) to (18), it is evident that RPC requires predicted output value $\bar{y}_j(k)$ at time *k* for agent *j*. However, this is not the case for agent *i* since network delay is only induced during information transmission from agent *j* to agent *i*, thus actual output data $y_i(k)$ is available at agent *i*. For NMAS (3) with a network communication delay τ, external consensus protocol (16) is said to solve the consensus problem if $\lim_{k\to\infty}\|y_i(k)-y_j(k)=0\|$ for $i,j\in V$ is achieved.

IV. NMASs Stability Analysis with RPC

This section considers the stability of the whole NMAS with the proposed RPC for constant output network delay. The analysis is simplified in matrix form for (14) to (15) with assumption that the reference input of agent 1 is zero. For this case, there is the following theorem.

Theorem 1: For RPC with constant output network delay τ, the closed-loop NMAS is stable if and only if all eigenvalues of the matrix (19) below are within the unit circle where $\Lambda(\tau)\in\Re^{2n(\tau+nm+1+n_{di})\times 2n(\tau+nm+1+n_{di})}$.

$$\Lambda(\tau)=\begin{bmatrix} A_T+B_TT & B_T(C_T+W) \\ T & (C_T+W) \end{bmatrix} \quad (19)$$

where $A_T=diag(A_1,A_2,...,A_n)$, $B_T=diag(B_1,B_2,...,B_n)$, $C_T=diag(C_1,C_2,...,C_n)$ and A_i,B_i, and C_i $\in\Re^{(\tau+nm+1+n_{di})\times(\tau+nm+1+n_{di})}$ are the reformulation of polynomials (2) and (3) in matrix form with appropriate dimension for $i=1,2,...,n$. The definition of matrix $T,W\in\Re^{n(\tau+nm+1+n_{di})\times n(\tau+nm+1+n_{di})}$ can be found at the end of this section.

Proof: Firstly, (14) and (15) are reformulated in matrix form for agent *i* and can be rewritten as

$$\hat{Y}_i=\hat{A}_i\hat{Y}_i+\tilde{A}_iY_i+\hat{B}_i\hat{U}_i+\tilde{B}_iU_i \quad (20)$$

$$\hat{U}_i=\hat{C}_i\hat{U}_i+\tilde{C}_iU_i+\hat{D}_i\hat{Y}_{ei}+\tilde{D}_iY_{ei}=\begin{cases} \hat{C}_i\hat{U}_i+\tilde{C}_iU_i+\hat{D}_i(\hat{Y}_i+K_i\sum_{\substack{j=2\\ j\in N_i}}^{n}(\hat{Y}_i-\hat{Y}_j)) \\ +\tilde{D}_i(Y_i+K_i\sum_{\substack{j=2\\ j\in N_i}}^{n}(Y_i-Y_j)), \quad for \quad i=1 \\ \hat{C}_i\hat{U}_i+\tilde{C}_iU_i+\hat{D}_iK_i\sum_{\substack{j=1\\ j\in N_i}}^{n}(\hat{Y}_i-\hat{Y}_j) \\ +\tilde{D}_iK_i\sum_{\substack{j=1\\ j\in N_i}}^{n}(Y_i-Y_j), \quad for \quad i=2,3,...,n \end{cases} \quad (21)$$

where

$$\hat{U}_i=\begin{bmatrix} \hat{u}_i(k-1\mid k-\tau) \\ \hat{u}_i(k-2\mid k-\tau) \\ \vdots \\ \vdots \\ \vdots \\ \hat{u}_i(k-\tau+1\mid k-\tau) \end{bmatrix} \quad U_i=\begin{bmatrix} u_i(k) \\ u_i(k-1) \\ \vdots \\ u_i(k-\tau) \\ \vdots \\ u_i(k-\tau-nm) \end{bmatrix} \quad Y_i=\begin{bmatrix} y_i(k) \\ y_i(k-1) \\ \vdots \\ y_i(k-\tau) \\ \vdots \\ y_i(k-\tau-nm) \end{bmatrix}$$

$$\hat{Y}_i=\begin{bmatrix} y_i(k\,|\,k-\tau) \\ y_i(k-1\,|\,k-\tau) \\ \vdots \\ \vdots \\ \vdots \\ y_i(k-\tau+1\,|\,k-\tau) \end{bmatrix} \quad \hat{C}_i=\begin{bmatrix} 0 & -c_{i1} & \cdots & -c_{in_{ci}} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \cdots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \ddots & \ddots & \cdots & \ddots & 0 \\ \vdots & \cdots & \cdots & \ddots & \ddots & \cdots & -c_{in_{ci}} \\ \vdots & \cdots & \cdots & \cdots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \ddots & -c_{i1} \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix}$$

$$\tilde{C}_i=\left[\begin{array}{ccc} 0_{(\tau-1)\times\tau} & \begin{array}{cccc} -c_{i(\tau-1)} & \cdots & \cdots & \cdots \\ \vdots & \ddots & \cdots & \cdots \\ -c_{in_{ci}} & \cdots & \ddots & \cdots \\ \vdots & \ddots & \cdots & \ddots \\ \vdots & \cdots & \ddots & \cdots \\ -c_{i1} & \cdots & \cdots & -c_{in_{ci}} \end{array} & 0_{(\tau-1)\times(nm+1-n_{ci})} \end{array}\right]$$

$$\hat{D}_i=\begin{bmatrix} 0 & -d_{i0} & \cdots & -d_{in_{di}} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \cdots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \ddots & \ddots & \cdots & \ddots & 0 \\ \vdots & \cdots & \cdots & \ddots & \ddots & \cdots & -d_{in_{di}} \\ \vdots & \cdots & \cdots & \cdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & -d_{i0} \end{bmatrix}$$

$$\tilde{D}_i=\left[\begin{array}{ccc} 0_{(\tau-1)\times\tau} & \begin{array}{cccc} -d_{i(\tau-1)} & \cdots & \cdots & \cdots \\ \vdots & \ddots & \cdots & \cdots \\ -d_{in_{di}} & \cdots & \ddots & \cdots \\ \vdots & \ddots & \cdots & \ddots \\ \vdots & \cdots & \ddots & \cdots \\ -d_{i1} & \cdots & \cdots & -d_{in_{di}} \end{array} & 0_{(\tau-1)\times(nm+1-n_{di})} \end{array}\right]$$

$$\hat{A}_i=\begin{bmatrix} 0 & -a_{i1} & \cdots & -a_{in_{ai}} & 0 & \cdots & 0 \\ \vdots & \cdots & \ddots & \cdots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \cdots & \ddots & \cdots & \ddots & 0 \\ \vdots & \cdots & \cdots & \cdots & \ddots & \cdots & -a_{in_{ai}} \\ \vdots & \cdots & \cdots & \cdots & \cdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & -a_{i1} \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix}$$

$$\tilde{A}_i=\left[\begin{array}{ccc} 0_{\tau\times\tau} & \begin{array}{ccccc} -a_{i\tau} & 0 & \cdots & \cdots & \cdots \\ \vdots & \ddots & \cdots & \ddots & \ddots \\ \vdots & \cdots & \ddots & \cdots & \ddots \\ \vdots & \cdots & \cdots & \ddots & \cdots \\ -a_{in_{ai}} & \cdots & \cdots & \cdots & -a_{i(\tau+nm+1)} \\ \vdots & \ddots & \cdots & \cdots & \cdots \\ \vdots & \cdots & \ddots & \cdots & \cdots \\ \vdots & \cdots & \cdots & \ddots & \cdots \\ -a_{i1} & \cdots & \cdots & \cdots & -a_{in_{ai}} \end{array} & 0_{\tau\times(nm+1-n_{ai})} \end{array}\right]$$

$$\hat{B}_i=\begin{bmatrix} b_{i0} & \cdots & b_{im_{bi}} & 0 & \cdots & 0 \\ \vdots & \ddots & \cdots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \ddots & \cdots & \ddots & 0 \\ \vdots & \cdots & \cdots & \ddots & \cdots & b_{im_{bi}} \\ \vdots & \cdots & \cdots & \cdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & b_{i0} \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix}$$

$$\tilde{B}_i=\left[\begin{array}{ccc} 0_{\tau\times\tau} & \begin{array}{cccccc} b_{i(\tau-1)} & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & \cdots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \ddots & \cdots & \ddots & \vdots \\ b_{m_{bi}} & \cdots & \cdots & \ddots & \cdots & \vdots \\ \vdots & \ddots & \cdots & \cdots & \ddots & \vdots \\ \vdots & \cdots & \ddots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \ddots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \ddots & 0 \\ b_0 & \cdots & \cdots & \cdots & \cdots & b_{m_{bi}} \end{array} & 0_{\tau\times(nm-m_{bi})} \end{array}\right]$$

0_* denotes a zero matrix, $\overline{n}_i$ represents the maximum value of n_{di} and n_{ai}, while $\overline{m}_i$ represents the maximum value of n_{ci} and m_{bi}. Let nm is the maximum order of $\overline{n}_i$ and $\overline{m}_i$.

Secondly, based on matrix in (20)-(21), for i=1,2,..,n the whole closed-loop NMAS can be simplified as follows:

$$\hat{U}_T=\hat{C}_T\hat{U}_T+\tilde{C}_TU_T+\hat{D}_T(I_1^R+L_K)\hat{Y}_T+\tilde{D}_T(I_1^R+L_K)Y_T \tag{22}$$

$$\hat{Y}_T=\hat{A}_T\hat{Y}_T+\tilde{A}_TY_T+\hat{B}_T\hat{U}_T+\tilde{B}_TU_T \tag{23}$$

where

$$\hat{U}_T=\begin{bmatrix}\hat{U}_1\\ \hat{U}_2\\ \vdots\\ \hat{U}_n\end{bmatrix} \quad U_T=\begin{bmatrix}U_1\\ U_2\\ \vdots\\ U_n\end{bmatrix} \quad \hat{Y}_T=\begin{bmatrix}\hat{Y}_1\\ \hat{Y}_2\\ \vdots\\ \hat{Y}_n\end{bmatrix} \quad Y_T=\begin{bmatrix}Y_1\\ Y_2\\ \vdots\\ Y_n\end{bmatrix}$$

$$I_1^R=\begin{bmatrix}1 & 0 & \cdots & 0\\ 0 & 0 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & \cdots & 0 & 0\end{bmatrix} \quad L_K=\begin{bmatrix}K_1N_1 & -K_1 & \cdots & -K_1\\ -K_2 & K_2N_2 & \cdots & \vdots\\ \vdots & \vdots & \ddots & -K_{n-1}\\ -K_n & \cdots & -K_n & K_nN_n\end{bmatrix}$$

$$\hat{C}_T=diag(\hat{C}_1,\hat{C}_2,...,\hat{C}_n) \qquad \tilde{C}_T=diag(\tilde{C}_1,\tilde{C}_2,...,\tilde{C}_n)$$
$$\hat{D}_T=diag(\hat{D}_1,\hat{D}_2,...,\hat{D}_n) \qquad \tilde{D}_T=diag(\tilde{D}_1,\tilde{D}_2,...,\tilde{D}_n)$$
$$\hat{A}_T=diag(\hat{A}_1,\hat{A}_2,...,\hat{A}_n) \qquad \tilde{A}_T=diag(\tilde{A}_1,\tilde{A}_2,...,\tilde{A}_n)$$
$$\hat{B}_T=diag(\hat{B}_1,\hat{B}_2,...,\hat{B}_n) \qquad \tilde{B}_T=diag(\tilde{B}_1,\tilde{B}_2,...,\tilde{B}_n)$$

Matrix I_1^R is $n\times n$ diagonal matrix with the first diagonal entries is one. Matrix L_K represents multiplication of K_i and Laplacian matrix L defined in section II. Simplified (22) obtained

$$\hat{U}_T=(I-\hat{C}_T)^{-1}(\tilde{C}_TU_T+\hat{D}_T(I_1^R+L_K)\hat{Y}_T+\tilde{D}_T(I_1^R+L_K)Y_T) \tag{24}$$

The inverse of matrix $I-\hat{C}_T$ is exists because it has non-zero diagonal line of upper triangular matrix. Combining (23) and (24) gives

$$\hat{Y}_T = (I-\hat{A}_T-\hat{B}_T(I-\hat{C}_T)^{-1}(\hat{D}_T I_1^R+\hat{D}_T L_K))^{-1} \\ ((\tilde{A}_T+\hat{B}_T(I-\hat{C}_T)^{-1}(\tilde{D}_T I_1^R+\tilde{D}_T L_K))Y_T+(\tilde{B}_T+\hat{B}_T(I-\hat{C}_T)^{-1}\tilde{C}_T)U_T) \tag{25}$$

As $(I-\hat{A}_T-\hat{B}_T(I-\hat{C}_T)^{-1}(\hat{D}_T I_1^R+\hat{D}_T L_K))$ is an upper triangular matrix with non-zero diagonal line, its inverse exists. Also, (25) can be denoted as

$$\hat{Y}_T = HY_T + FU_T \tag{26}$$

where

$$H = (I-\hat{A}_T-\hat{B}_T(I-\hat{C}_T)^{-1}(\hat{D}_T I_1^R+\hat{D}_T L_K))^{-1}(\tilde{A}_T+\hat{B}_T(I-\hat{C}_T)^{-1}(\tilde{D}_T I_1^R+\tilde{D}_T L_K)) \\ F = (I-\hat{A}_T-\hat{B}_T(I-\hat{C}_T)^{-1}(\hat{D}_T I_1^R+\hat{D}_T L_K))^{-1}(\tilde{B}_T+\hat{B}_T(I-\hat{C}_T)^{-1}\tilde{C}_T) \tag{27}$$

To compensate the network communication delay τ, delay compensator (DC) or selector is designed to choose the latest value of the prediction sequence. To do this, we introduce

$$CR = \begin{bmatrix} 1 & 0_{1\times(\tau-1)} \end{bmatrix} \in \Re^{1\times\tau} \qquad g = 0_{1\times\tau} \in \Re^{1\times\tau}$$

$$DC = \begin{bmatrix} CR & g & \cdots & g \\ g & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & g \\ g & \cdots & g & CR \end{bmatrix} \in \Re^{n\times n\tau} \tag{28}$$

Matrix g is a zero matrix with one row and multiple columns. The number of columns is determined by the value of τ.

The output of NMAS agents with network delay compensation denotes as $\bar{Y}_T$ can be expressed as follows:

$$\bar{Y}_T = \bar{H}Y_T + \bar{F}U_T = DC\hat{Y}_T \tag{29}$$

$$\bar{H} = \begin{bmatrix} \bar{H}_{11} & \bar{H}_{12} & \cdots & \bar{H}_{1n} \\ \bar{H}_{21} & \bar{H}_{22} & \cdots & \bar{H}_{2n} \\ \vdots & \cdots & \ddots & \vdots \\ \bar{H}_{n1} & \cdots & \cdots & \bar{H}_{nn} \end{bmatrix} \in \Re^{n\times n(\tau+nm+1)}$$

$$\bar{F} = \begin{bmatrix} \bar{F}_{11} & \bar{F}_{12} & \cdots & \bar{F}_{1n} \\ \bar{F}_{21} & \bar{F}_{22} & \cdots & \bar{F}_{2n} \\ \vdots & \cdots & \ddots & \vdots \\ \bar{F}_{n1} & \cdots & \cdots & \bar{F}_{nn} \end{bmatrix} \in \Re^{n\times n(\tau+nm+1)}$$

where

$$\bar{Y}_T = \begin{bmatrix} \bar{y}_1(k) & \bar{y}_2(k) & \cdots & \bar{y}_{n-1}(k) & \bar{y}_n(k) \end{bmatrix}^T \in \Re^{n\times 1}$$

$$\bar{H}_{ii} = \begin{bmatrix} h_{ii,1} & h_{ii,2} & \cdots & h_{ii,(\tau+nm+1)} \end{bmatrix} \\ \bar{H}_{ij} = \begin{bmatrix} h_{ij,1} & h_{ij,2} & \cdots & h_{ij,(\tau+nm+1)} \end{bmatrix} \\ \bar{F}_{ii} = \begin{bmatrix} f_{ii,1} & f_{ii,2} & \cdots & f_{ii,(\tau+nm+1)} \end{bmatrix} \\ \bar{F}_{ij} = \begin{bmatrix} f_{ij,1} & f_{ij,2} & \cdots & f_{ij,(\tau+nm+1)} \end{bmatrix} \tag{30}$$

Let

$$\begin{bmatrix} soe_i(k) \\ \vdots \\ soe_i(k-n_{di}) \end{bmatrix} = \begin{bmatrix} \sum_{\substack{i\neq j \\ j\in N_i}}^{n} K_i(y_i(k)-\bar{y}_j(k)) \\ \vdots \\ \sum_{\substack{i\neq j \\ j\in N_i}}^{n} K_i(y_i(k-n_{di})-\bar{y}_j(k-n_{di})) \end{bmatrix} \tag{31}$$

where soe_i is the sum of output error between output of agent i and received output with delay compensation from its neighboring agent(s).

Thirdly, substitute (31) in (16) yields

$$u_1(k) = -c_{11}u_1(k-1)-..-c_{1n_{c1}}u_1(k-n_{c1})-d_{10}y_1(k) \\ -d_{11}y_1(k-1)-..-d_{1n_{d1}}y_1(k-n_{d1})-d_{10}soe_1(k) \\ -d_{11}soe_1(k-1)-..-d_{1n_{d1}}soe_1(k-n_{d1}) \tag{32}$$

and for $i\neq 1$, (33)

$$u_i(k) = -c_{i1}u_i(k-1)-..-c_{in_{ci}}u_i(k-n_{ci})-d_{i0}soe_i(k) \\ -d_{i1}soe_i(k-1)-..-d_{in_{di}}soe_i(k-n_{di})$$

For i=1,2,..n, simplify (31) based on (29) in matrix form as below:

$$\begin{bmatrix} soe_1(k) \\ \vdots \\ soe_n(k) \end{bmatrix} = (I_{yi}-I_{k0}\bar{H})Y_T - I_{k0}\bar{F}U_T = \tilde{H}Y_T + \tilde{F}U_T \\ \vdots \\ \begin{bmatrix} soe_1(k-n_{d1}) \\ \vdots \\ soe_n(k-n_{dn}) \end{bmatrix} = \tilde{H}Y_{T(k-n_{di})} + \tilde{F}U_{T(k-n_{di})} \tag{34}$$

where I_{k0} is $n\times n$ matrix with diagonal entries are zero and other entries are K_i.

$$I_{k0} = \begin{bmatrix} 0 & K_1 & \cdots & K_1 \\ K_2 & \ddots & \ddots & \vdots \\ \vdots & \cdots & \ddots & K_{n-1} \\ K_n & \cdots & K_n & 0 \end{bmatrix}$$

$$I_{yi} = \begin{bmatrix} K_1N_1 & 0_{1\times(\tau+nm)} & & & 0_{1\times(n-1)(\tau+nm+1)} \\ 0_{1\times(2-1)(\tau+nm+1)} & K_2N_2 & 0_{1\times(\tau+nm)} & & \vdots \\ \vdots & & \ddots & \ddots & 0_{1\times(n-(n-1))(\tau+nm+1)} \\ 0_{1\times(n-1)(\tau+nm+1)} & & & K_nN_n & 0_{1\times(\tau+nm)} \end{bmatrix}$$

$$Y_{T(k-n_{di})} = [y_1(k-n_{di}) \quad y_1(k-n_{di}-1) \quad \cdots \quad y_1(k-\tau-nm-n_{di}) \\ y_2(k-n_{di}) \quad \ldots \quad y_n(k-\tau-nm-n_{di}+1) \quad y_n(k-\tau-nm-n_{di})]^T$$

$$U_{T(k-n_{di})} = [u_1(k-n_{di}) \quad u_1(k-n_{di}-1) \quad \cdots \quad u_1(k-\tau-nm-n_{di}) \\ u_2(k-n_{di}) \quad \ldots \quad u_n(k-\tau-nm-n_{di}+1) \quad u_n(k-\tau-nm-n_{di})]^T$$

Combined (32)-(34) in matrix form gives

$$U_{m(k)} = C_{te}U_{(k-1)} - \bar{d}_{i0}(I_R+\tilde{H})Y_T - \bar{d}_{i0}\tilde{F}U_T - \bar{d}_{i1}(I_R+\tilde{H})Y_{T(k-1)} \\ -\bar{d}_{i1}\tilde{F}U_{T(k-1)} - \ldots - \bar{d}_{in_{di}}(I_R+\tilde{H})Y_{T(k-n_{di})} - \bar{d}_{in_{di}}\tilde{F}U_{T(k-n_{di})} \tag{35}$$

where

$$\bar{d}_{i0} = diag(d_{10}, d_{20}, \ldots, d_{n0})$$

$$\bar{d}_{i1} = diag(d_{11}, d_{21}, \ldots, d_{n1})$$

$$\bar{d}_{in_{di}} = diag(d_{1n_{di}}, d_{2n_{di}}, \ldots, d_{nn_{di}})$$

$$U_{m(k)} = \begin{bmatrix} u_1(k) & u_2(k) & \cdots & u_n(k) \end{bmatrix}^T$$

$$U_{(k-1)} = \begin{bmatrix} U_{1(k-1)} & U_{2(k-1)} & \cdots & U_{n(k-1)} \end{bmatrix}^T$$

$$U_{i(k-1)} = \begin{bmatrix} u_i(k-1) & u_i(k-2) & \cdots & u_i(k-\tau-nm-n_{di}-1) \end{bmatrix}^T$$

$$I_R = \begin{bmatrix} I_1 & 0_{1\times(n-1)(\tau+nm+1)} \\ 0_{(n-1)\times n(\tau+nm+1)} & \end{bmatrix} \qquad I_1 = \begin{bmatrix} 1 & 0_{1\times(\tau+nm)} \end{bmatrix}$$

$$C_{te} = \begin{bmatrix} \breve{C}_1 & \breve{C}_2 & \cdots & \breve{C}_n \end{bmatrix}^T$$

$$\breve{C}_i = \begin{bmatrix} 0_{1\times((i-1)(\tau+nm+n_{di}+1))} & -c_{i1} & -c_{i2} & \cdots & -c_{n_{ci}} & 0_{1\times(\tau+nm+n_{di}+1-n_{ci})} & 0_{1\times((n-i)(\tau+nm+n_{di}+1))} \end{bmatrix}$$

Let

$$\sigma = I_R + \tilde{H} \qquad (36)$$
$$\alpha = \tilde{F}$$

$$U_{m(k)} = C_{te}U_{(k-1)} - \bar{d}_{i0}\sigma Y_T - \bar{d}_{i1}\sigma Y_{T(k-1)} - \ldots - \bar{d}_{in_{di}}\sigma Y_{T(k-n_{di})} - \bar{d}_{i0}\alpha U_T - \bar{d}_{i1}\alpha U_{T(k-1)} - \ldots - \bar{d}_{in_{di}}\alpha U_{T(k-n_{di})} \qquad (37)$$

$$U_{m(k)} = C_{te}U_{(k-1)} + \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_n \end{bmatrix} Y_{(k)} + \begin{bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_n \end{bmatrix} U_{(k-1)} \qquad (38)$$

where

$$\begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_n \end{bmatrix} = \begin{bmatrix} (I_1 + K_1(\bar{N}_1 - \sum_{\substack{j=2 \\ j\neq i}}^{n} \bar{H}_{j1}))D_1 & (-K_1\sum_{\substack{j=2 \\ j\neq i}}^{n} \bar{H}_{j2})D_1 & \cdots\ \cdots & (-K_1\sum_{\substack{j=2 \\ j\neq i}}^{n} \bar{H}_{jn})D_1 \\ (-K_2\sum_{\substack{j=1 \\ j\neq i}}^{n} \bar{H}_{j1})D_2 & (K_2(\bar{N}_2 - \sum_{\substack{j=1 \\ j\neq i}}^{n} \bar{H}_{j2}))D_2 & \cdots\ \cdots & (-K_2\sum_{\substack{j=1 \\ j\neq i}}^{n} \bar{H}_{jn})D_2 \\ \vdots & \vdots & \cdots\ \ddots & \vdots \\ (-K_n\sum_{\substack{j=1 \\ j\neq i}}^{n-1} \bar{H}_{j1})D_n & (-K_n\sum_{\substack{j=1 \\ j\neq i}}^{n-1} \bar{H}_{j2})D_n & \cdots\ \cdots & (K_n(\bar{N}_n - \sum_{\substack{j=1 \\ j\neq i}}^{n-1} \bar{H}_{jn}))D_n \end{bmatrix} \qquad (39)$$

$$\begin{bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_n \end{bmatrix} = \begin{bmatrix} (-K_1\sum_{\substack{j=2 \\ j\neq i}}^{n} \bar{F}_{j1})D_1 & (-K_1\sum_{\substack{j=2 \\ j\neq i}}^{n} \bar{F}_{j2})D_1 & \cdots\ \cdots & (-K_1\sum_{\substack{j=2 \\ j\neq i}}^{n} \bar{F}_{jn})D_1 \\ (-K_2\sum_{\substack{j=1 \\ j\neq i}}^{n} \bar{F}_{j1})D_2 & (-K_2\sum_{\substack{j=1 \\ j\neq i}}^{n} \bar{F}_{j2})D_2 & \cdots\ \cdots & (-K_2\sum_{\substack{j=1 \\ j\neq i}}^{n} \bar{F}_{jn})D_2 \\ \vdots & \vdots & \cdots\ \ddots & \vdots \\ (-K_n\sum_{\substack{j=1 \\ j\neq i}}^{n-1} \bar{F}_{j1})D_n & (-K_n\sum_{\substack{j=1 \\ j\neq i}}^{n-1} \bar{F}_{j2})D_n & \cdots\ \cdots & (-K_n\sum_{\substack{j=1 \\ j\neq i}}^{n-1} \bar{F}_{jn})D_n \end{bmatrix} \qquad (40)$$

and for $i = 1,2,...,n$,

$$\bar{N}_i = \begin{bmatrix} N_i & 0_{1\times(\tau+nm)} \end{bmatrix} \qquad Y_{(k)} = \begin{bmatrix} Y_{1(k)} & Y_{2(k)} & \cdots & Y_{n(k)} \end{bmatrix}^T$$

$$D_i = \begin{bmatrix} -d_{i0} & -d_{i1} & \cdots & -d_{in_{di}} & 0 & \cdots & 0 \\ 0 & -d_{i0} & \cdots & \cdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -d_{i0} & \cdots & \cdots & -d_{in_{di}} \end{bmatrix} \in R^{(\tau+nm+1)\times(\tau+nm+1+n_{di})}$$

$$Y_{i(k)} = \begin{bmatrix} y_i(k) & y_i(k-1) & \cdots & y_i(k-\tau-nm-n_{di}) \end{bmatrix}^T$$

Based on (38)-(40), control input of every agent i at time k can be expressed as follows:

$$u_{i(k)} = \breve{C}_i U_{(k-1)} + \begin{bmatrix} S_{i1} & S_{i2} & \ldots & S_{in} \end{bmatrix} \begin{bmatrix} Y_{1(k)} \\ Y_{2(k)} \\ \vdots \\ Y_{n(k)} \end{bmatrix} + \begin{bmatrix} Q_{i1} & Q_{i2} & \ldots & Q_{in} \end{bmatrix} \begin{bmatrix} U_{1(k-1)} \\ U_{2(k-1)} \\ \vdots \\ U_{n(k-1)} \end{bmatrix} \qquad (41)$$

For $i = 1,2,...,n$, matrix $S_{i1}, S_{i2},..., S_{in}$ and $Q_{i1}, Q_{i2},..., Q_{in}$ are $S_{i1}, S_{i2},..., S_{in} \in \Re^{1\times(\tau+nm+1+n_{di})}$. In compact form, control input of NMAS agents can be formed as,

$$U_{(k)} = TY_{(k)} + (C_T + W)U_{(k-1)} \qquad (42)$$

where

$$U_{(k)} = \begin{bmatrix} U_{1(k)} & U_{2(k)} & \cdots & U_{n(k)} \end{bmatrix}^T$$

$$U_{i(k)} = \begin{bmatrix} u_i(k) & u_i(k-1) & \cdots & u_i(k-\tau-nm-n_{di}) \end{bmatrix}^T$$

$$T = \begin{bmatrix} S_{11} & S_{12} & \cdots\ \cdots\ \cdots\ \cdots & S_{1n} \\ 0_{(\tau+nm+n_{di})\times(\tau+nm+1+n_{di})} & 0_{(\tau+nm+n_{di})\times(\tau+nm+1+n_{di})} & \cdots\ \cdots\ \cdots\ \cdots & 0_{(\tau+nm+n_{di})\times(\tau+nm+1+n_{di})} \\ S_{21} & S_{22} & \cdots\ \cdots\ \cdots\ \cdots & S_{2n} \\ 0_{(\tau+nm+n_{di})\times(\tau+nm+1+n_{di})} & 0_{(\tau+nm+n_{di})\times(\tau+nm+1+n_{di})} & \cdots\ \cdots\ \cdots\ \cdots & 0_{(\tau+nm+n_{di})\times(\tau+nm+1+n_{di})} \\ \vdots & \vdots & \cdots\ \cdots\ \cdots\ \cdots & \vdots \\ S_{n1} & S_{n2} & \cdots\ \cdots\ \cdots\ \cdots & S_{nn} \\ 0_{(\tau+nm+n_{di})\times(\tau+nm+1+n_{di})} & 0_{(\tau+nm+n_{di})\times(\tau+nm+1+n_{di})} & \cdots\ \cdots\ \cdots\ \cdots & 0_{(\tau+nm+n_{di})\times(\tau+nm+1+n_{di})} \end{bmatrix}$$

$$W = \begin{bmatrix} Q_{11} & Q_{12} & \cdots\ \cdots\ \cdots\ \cdots & Q_{1n} \\ 0_{(\tau+nm+n_{di})\times(\tau+nm+1+n_{di})} & 0_{(\tau+nm+n_{di})\times(\tau+nm+1+n_{di})} & \cdots\ \cdots\ \cdots\ \cdots & 0_{(\tau+nm+n_{di})\times(\tau+nm+1+n_{di})} \\ Q_{21} & Q_{22} & \cdots\ \cdots\ \cdots\ \cdots & Q_{2n} \\ 0_{(\tau+nm+n_{di})\times(\tau+nm+1+n_{di})} & 0_{(\tau+nm+n_{di})\times(\tau+nm+1+n_{di})} & \cdots\ \cdots\ \cdots\ \cdots & 0_{(\tau+nm+n_{di})\times(\tau+nm+1+n_{di})} \\ \vdots & \vdots & \cdots\ \cdots\ \cdots\ \cdots & \vdots \\ Q_{n1} & Q_{n2} & \cdots\ \cdots\ \cdots\ \cdots & Q_{nn} \\ 0_{(\tau+nm+n_{di})\times(\tau+nm+1+n_{di})} & 0_{(\tau+nm+n_{di})\times(\tau+nm+1+n_{di})} & \cdots\ \cdots\ \cdots\ \cdots & 0_{(\tau+nm+n_{di})\times(\tau+nm+1+n_{di})} \end{bmatrix}$$

$$C_T = diag(C_1, C_2, ..., C_n)$$

$$C_i = \begin{bmatrix} -c_{i1} & -c_{i2} & \cdots & -c_{in_{ci}} & 0 & \cdots & 0 & 0 \\ & & I_{(\tau+nm+n_{di})\times(\tau+nm+n_{di})} & & & & & 0 \end{bmatrix}$$

Fourthly, the whole NMAS can be described in the following form:

$$Y_{(k+1)} = A_T Y_{(k)} + B_T U_{(k)}$$
$$Y_{(k+1)} = (A_T + B_T T)Y_{(k)} + B_T(C_T + W)U_{(k-1)} \qquad (43)$$

where

$$A_T = diag(A_1, A_2, ..., A_n) \qquad B_T = diag(B_1, B_2, ..., B_n)$$

$$A_i = \begin{bmatrix} -a_{i1} & -a_{i2} & \cdots & -a_{in_{ai}} & 0 & \cdots & 0 & 0 \\ & & I_{(\tau+nm+n_{di})\times(\tau+nm+n_{di})} & & & & & 0 \end{bmatrix}$$

$$B_i = \begin{bmatrix} b_{i0} & b_{i1} & \cdots & b_{im_{bi}} & 0 & \cdots & 0 \\ & & 0_{(\tau+nm+n_{di})\times(\tau+nm+1+n_{di})} & & & & \end{bmatrix}$$

As a result, combining (41) and (42) yields the following closed-loop system equation for NMAS which can be expressed as:

$$X_{k+1} = \Lambda(\tau)X_k \qquad (44)$$

where

$$X_k = [y_1(k) \ \ y_1(k-1) \ \cdots \ y_1(k-\tau-nm-n_{di}) \ \ y_2(k) \ \cdots \ y_2(k-\tau-nm-n_{di}) \ \cdots \ y_n(k) \ \cdots \ y_n(k-\tau-nm-n_{di}) \ \ u_1(k-1) \ \cdots \ u_1(k-\tau-nm-n_{di}-1) \ \ u_2(k-1) \cdots \ u_2(k-\tau-nm-n_{di}-1) \ \cdots \ \cdots \ u_n(k-\tau-nm-n_{di}-1)]$$

The polynomial order for agent's model and virtual controller in (2) and (5) for i=1,2,..,n are designed to be similar. For RPC with constant output network delay, the closed-loop NMAS is stable if and only if all eigenvalues of the matrix (44) are within the unit circle.

V. Selection of Coupling Gain

In this section, theorem of determining appropriate coupling gains value for NMAS protocol (16) is introduced.

Theorem 2: The protocol defined in (16-18) solves the consensus problem if the following conditions are satisfied:

(i) There exist coupling gains K_i for i=1,2,..,n satisfying that $\lim_{z\to 1}(z-1)M(zI-P(I-GF)^{-1}GH)^{-1}(P(I-GF)^{-1}G_1R_{ext}(z))=0$, where $P=diag(P_1(z^{-1}),P_2(z^{-1}),...,P_n(z^{-1}))$, I represents $n\times n$ identity matrix, $R_{ext}(z)$ is a $n\times 1$ column vector with the first row equal to $R(z)$ and the rest row elements are zeroes, and G_1 is a $n\times n$ matrix with the first diagonal entries is $G_1(z^{-1})$ and the rest of the entries are zeroes. The definition of *M*, *GF* and *GH* can be found at the end of this section.

(ii) The second smallest eigenvalue λ_2 of L_K with designed K_i satisfies $\lambda_2 \le \left(\frac{n}{n-1}\right)\min_i l_{ii}$, where l_{ii} is the diagonal element of L_K defined in (22) [34].

(iii) The eigenvalues λ_i satisfy $0=\lambda_1<\lambda_2\le ...\le\lambda_n\le 2\bar{k}$ where $\bar{k}$ is the largest degree in the graph $\mathcal{G}$ [35].

Proof: Any discrete-time sequence in the following section is written in *z*-transform. From (16), control input $u_i(k)$ for i=1,2,..,n is converted to *z*-transform and expanded into one matrix form as follows:

$$\begin{bmatrix}U_1(z)\\U_2(z)\\\vdots\\U_n(z)\end{bmatrix}=\begin{bmatrix}G_1(z^{-1}) & 0 & \cdots & 0\\0 & G_2(z^{-1}) & \ddots & \vdots\\\vdots & \cdots & \ddots & 0\\0 & \cdots & 0 & G_n(z^{-1})\end{bmatrix}\left(\begin{bmatrix}1 & 0 & \cdots & 0\\0 & 0 & \ddots & \vdots\\\vdots & \ddots & \ddots & \vdots\\0 & \cdots & \cdots & 0\end{bmatrix}\begin{bmatrix}R(z)\\0\\\vdots\\0\end{bmatrix}-\begin{bmatrix}1 & 0 & \cdots & 0\\0 & 0 & \ddots & \vdots\\\vdots & \ddots & \ddots & \vdots\\0 & \cdots & \cdots & 0\end{bmatrix}\begin{bmatrix}Y_1(z)\\Y_2(z)\\\vdots\\Y_n(z)\end{bmatrix}\right)$$
$$+\begin{bmatrix}G_1(z^{-1})K_1 & 0 & \cdots & 0\\0 & G_2(z^{-1})K_2 & \ddots & \vdots\\\vdots & \cdots & \ddots & 0\\0 & \cdots & 0 & G_n(z^{-1})K_n\end{bmatrix}\left(\begin{bmatrix}-N_1 & 0 & \cdots & 0\\0 & -N_2 & \ddots & \vdots\\\vdots & \ddots & \ddots & \vdots\\0 & \cdots & \cdots & -N_n\end{bmatrix}\begin{bmatrix}Y_1(z)\\Y_2(z)\\\vdots\\Y_n(z)\end{bmatrix}+\begin{bmatrix}0 & 1 & \cdots & 1\\1 & 0 & \ddots & \vdots\\\vdots & \ddots & \ddots & 1\\1 & \cdots & 1 & 0\end{bmatrix}\begin{bmatrix}\bar{Y}_1(z)\\\bar{Y}_2(z)\\\vdots\\\bar{Y}_n(z)\end{bmatrix}\right) \quad (45)$$

Next, rewrite (29) and (30) in discrete-time signal and transform it to a function of *z*-transform with variable *z* yields

$$\begin{bmatrix}\bar{Y}_1(z)\\\bar{Y}_2(z)\\\vdots\\\bar{Y}_n(z)\end{bmatrix}=\begin{bmatrix}H_{11p}(z^{-1}) & H_{12p}(z^{-1}) & \cdots & H_{1np}(z^{-1})\\H_{21p}(z^{-1}) & H_{22p}(z^{-1}) & \ddots & \vdots\\\vdots & \ddots & \ddots & \vdots\\H_{n1p}(z^{-1}) & \cdots & \cdots & H_{nnp}(z^{-1})\end{bmatrix}\begin{bmatrix}Y_1(z)\\Y_2(z)\\\vdots\\Y_n(z)\end{bmatrix}+\begin{bmatrix}F_{11p}(z^{-1}) & F_{12p}(z^{-1}) & \cdots & F_{1np}(z^{-1})\\F_{21p}(z^{-1}) & F_{22p}(z^{-1}) & \ddots & \vdots\\\vdots & \ddots & \ddots & \vdots\\F_{n1p}(z^{-1}) & \cdots & \cdots & F_{nnp}(z^{-1})\end{bmatrix}\begin{bmatrix}U_1(z)\\U_2(z)\\\vdots\\U_n(z)\end{bmatrix} \quad (46)$$

Substitute (46) into (45) and simplify the equation yields

$$U_T(z)=(I-GF)^{-1}(G_1R(z)+GHY_T(z)) \quad (47)$$

where G_1 is $n\times n$ matrix with first diagonal entries equal to $G_1(z^{-1})$, $R(z)$ is $n\times 1$ matrix with first row equal to input $R_{ext}(z)$. Elements in matrix GF and GH contains multiplication of $G_i(z^{-1})$, K_i with $\sum_{j=1, j\ne i}^{n}F_{jip}(z^{-1})$ and $\sum_{j=1, j\ne i}^{n}H_{jip}(z^{-1})$ respectively.

From (1), output of agent *i* for *i*=1,2,..,*n* can be expanded in matrix form as:

$$y_T(k+1)=\begin{bmatrix}y_1(k+1)\\y_2(k+1)\\\vdots\\y_n(k+1)\end{bmatrix}=\begin{bmatrix}P_1(z^{-1}) & 0 & \cdots & 0\\0 & P_2(z^{-1}) & \ddots & \vdots\\\vdots & \cdots & \ddots & 0\\0 & \cdots & 0 & P_n(z^{-1})\end{bmatrix}\begin{bmatrix}u_1(k)\\u_2(k)\\\vdots\\u_n(k)\end{bmatrix} \quad (48)$$

$$y_T(k+1)=Pu_T(k)$$

Substitute (47) into (48) in *z*-transform yields

$$Y_T(z)=(zI-P(I-GF)^{-1}GH)^{-1}(P(I-GF)^{-1}G_1R(z)) \quad (49)$$

Let $e(k)=[e_2(k)\ \ e_3(k)\ \ \cdots\ \ e_n(k)]^T$, $e(k)=My_T(k)$, $e_i(k)=y_i(k)-y_1(k), i\in V$ and $M=[-1_{n-1}\ \ I_{n-1}]$ where $1_{n-1}\in R^{(n-1)\times 1}$ is the column vector with ones, and I_{n-1} is identity matrix with (*n*-1)-dimension. The protocol (16) solves the consensus problem if and only if $\lim_{k\to\infty}\|e(k)\|=0$ is true. According to the Final Value theorem for z-transform, $\lim_{k\to\infty}\|e(k)\|=\lim_{z\to 1}(z-1)E(z)$. Thus, $E(z)$ can be written as

$$E(z)=MY_T(z)=M(zI-P(I-GF)^{-1}GH)^{-1}(P(I-GF)^{-1}G_1R(z))^T \quad (50)$$

Applying $\lim_{z\to 1}(z-1)E(z)=0$ into (50), the appropriate optimal value of K_i is determined which satisfy $\lambda_2\le\left(\frac{n}{n-1}\right)\min_i l_{ii}$, where l_{ii} is the diagonal element in L_K [32] and all eigenvalues satisfy $0=\lambda_1<\lambda_2\le ...\le\lambda_n\le 2\bar{k}$ where $\bar{k}$ is the largest degree in the graph $\mathcal{G}$ [33].

VI. Simulation and Experiment

A. Numerical Simulation

Consider NMAS with n =2 indexed by 1 and 2 respectively. The mathematical model of the two water level control systems in the simulation are formulated based on the real laboratory-scale test rig. The test rigs are bench-mounting CE117 process trainer developed by the TecQuipment and a bench-top coupled-tank developed by Quanser Consulting Inc. In this work, the test rigs are purposely setup having a simple water level control process with variable water pump speed. Water level will be controlled through a voltage supplied to the water pump. Water level on CE117 is acquired via a vertical capacitive water level transmitter while on the coupled-tanks, water level is measured through a pressure-sensitive sensor located at the bottom of the water tank. Water tanks on both test rigs are also scaled for a visual approximate reading of the water level.

Since there is no prior information available regarding the test rigs actual parameters, the mathematical model in (51) and (52) has been developed using black box system identification which exploits test rigs measured input and output data sampled at 0.10s. The dynamics of agent *i* (*i* = 1, 2) are described by (2) and can be written as follows:

$$P_1(z^{-1}) = \frac{0.001703z^{-1} + 0.005419z^{-2}}{1 - 0.9718z^{-1} - 0.025z^{-2}} \quad (51)$$

$$P_2(z^{-1}) = \frac{0.002352z^{-1} + 0.001673z^{-2}}{1 - 1.646z^{-1} + 0.6573z^{-2}} \quad (52)$$

To ensure model accuracy, several sets of data are collected meticulously on an identified optimum test rig operating mode. Then, using the Matlab System Identification toolbox, each data set is divided into two small sets of data for the toolbox's identification and validation purposes. Approximated derived model based on ARX model structure is has been selected based on the best fit percentage. Before the model is implemented in the simulation, a comparison between the simulated model output and the actual test rigs output has been carried out. This is to ensure that the estimated model is able to replicate the test rig characteristic as close as possible.

The transfer function of virtual controller for each agent has been obtained by employing the Proportional-Integral (PI) controller. The designed controllers are tuned using basic Ziegler-Nichols method prior to RPC design to ensure that the closed-loop agent's system without network delay is stable and they are

$$G_1(z^{-1}) = \frac{1.35 - 1.346z^{-1}}{1 - z^{-1}} \qquad G_2(z^{-1}) = \frac{1 - 0.96z^{-1}}{1 - z^{-1}} \quad (53)$$

The coupling gains K_1 and K_2 which have been designed based on Theorem 2 are calculated to be 4.6164 and 4.6164 respectively. The connection of undirected graph with the communication delay τ between agents for NMAS with *n* =2 is shown in Fig. 1.

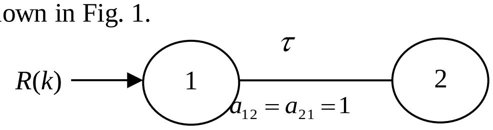

Fig. 1. Communication topology of order 2 (*n* =2)

The stability analysis has been derived in section IV for $\tau = 6,7,8,9$ step-delays using models in (51-53) and the designed coupling gains. The result shows that the eigenvalues for matrix (44) are within the unit circle for the specified delay range. Therefore, based on the above analysis, it can be concluded that NMAS is stable for $\tau = 6,7,8,9$ step-delays and has achieved the consensus. With the same model used in the analysis, the results obtained are verified using simulation.

B. *Consensus Output* (*Simulation*)

In this section, a study on the simulation for NMAS with $n = 2$ is presented to demonstrate the effectiveness of the designed method developed in this paper. The simulation diagram is shown in Fig. 2.

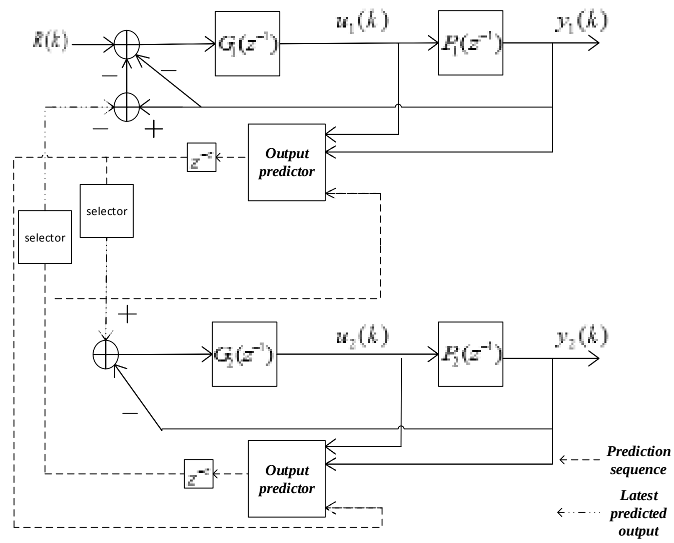

Fig. 2. Block diagram of simulation

The closed-loop NMAS consists of two inhomogeneous agent models $P_i(z^{-1})$, two virtual controllers transfer function $G_i(z^{-1})$, and two output predictors. The network delay is represented by $z^{-\tau}$ block in Simulink Matlab. The network delay is measured to be within the range of 1-step to 3-step. The simulation has been carried out on 4000 samples at an interval 0.10 s sampling period. A comparison is made between the PI controller and the RPC for four different step delay starting from 6-step delay (based on simulation and experimental results with PI controller in (53), delay equal or lower than 6-step will not affect the output of NMAS for agent's model in (51-52)). Hence, the tested maximum output network delay is 9-step delay, which is the upper bound of 3-steps range mentioned earlier.

Based on the simulation and experimental results with the PI controller, external consensus can only be achieved when the network delay is within 6-step and below. Thus, to compare and to show the effectiveness of the proposed prediction strategy (RPC) over the PI controller, 6-step and above has been considered. The range of 6-step to 9-step delay times are chosen because the maximum measured delay within the network is 3-step.

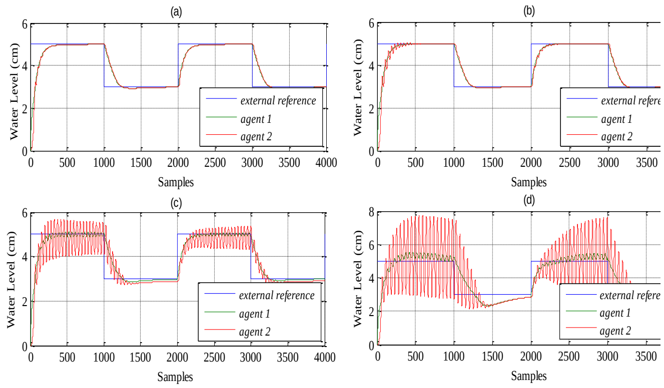

Fig. 3. External consensus for NMAS using PI controller (no delay compensation) with (a) 6-step delay ; (b) 7-step delay ;(c) 8-step delay; (d) 9-step delay

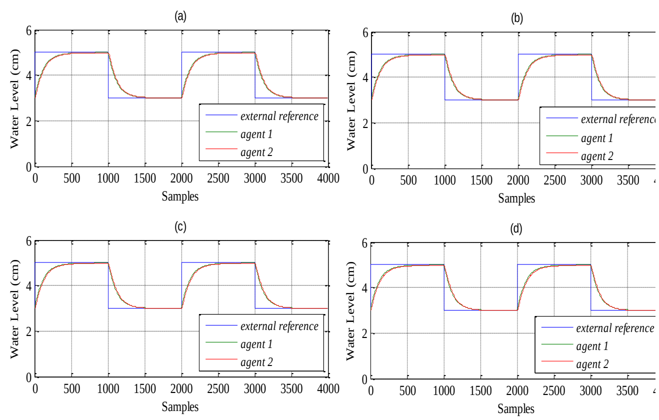

Fig. 4. External consensus for NMAS using RPC with (a) 6-step delay ; (b) 7-step delay ;(c) 8-step delay; (d) 9-step delay

Based on Fig. 3 and 4, it is observed that there is an obvious difference in output performance between PI controller and RPC particularly at 8-step and 9-step delay. NMAS with PI controller has the tendency to oscillate when the network delay is increased while NMAS with RPC continue to produce a stable output. The estimated prediction error (*ei*) between actual and predicted output of agents i = 1, 2 are shown in Fig. 5 at 9-step delay. These results indicate that the proposed RPC has effective delay compensating ability and able to produce accurate agents output measurement. To verify this result, a practical experiment was conducted.

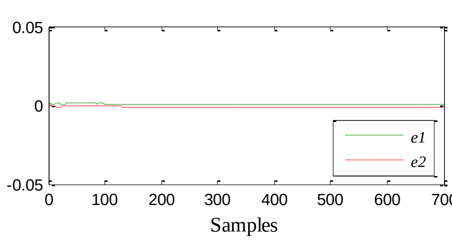

Fig. 5. Estimated prediction error at 9 step-delay (simulation)

C. *Experimental Setup for NMAS test rigs*

To validate the simulation result, NMAS with two different test rigs was built as illustrated in Fig. 6.

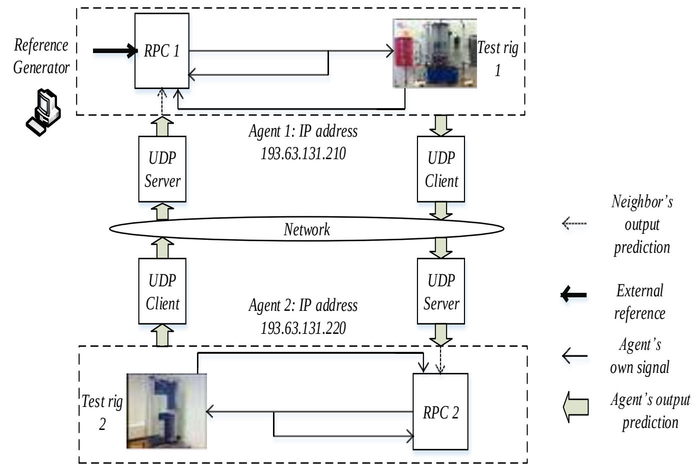

Fig. 6. Networked Multi-agents System (NMAS) composed of two different test rigs through network

In this setup, agent 1 and agent 2 are represented by Test rig 1 and Test rig 2 respectively. Test rig 1 is the CE117 process trainer and Test rig 2 is the coupled-tank systems which have been described earlier in the section VI. Test rig 1 consists two main parts: the Experiment Module and the Control Module. The Experiment Module has two separate flow circuits which are Process Flow Circuit and Heater Flow Circuit. Only Process Flow Circuit is utilised in this work. The Control Module provides access to all of the actuators and transmitters circuit contained in the Experimental Module. It also provides an interface between Test rig 1 to the PC through A/D and D/A port channels. Test rig 2 has simpler setup which consists of two water tanks (upper and lower) and one inlet water pump. It does not have any separate control module. For this experiment, only the upper tank is used. Test rig 2 is connected to the PC through A/D and D/A port channels. Water level on both test rigs is controlled through a voltage to the water pump which is supplied from the PC.

A/D and D/A port channels on Test rig 1 and Test rig 2 are then directly connected to networked controller board. The networked controller board has twelve A/D input channels and six D/A channels which can be selected depending on the user preference. These channels were connected to the designed S-function block in Matlab Simulink to allow signal interfacing from/to PC to/from test rigs through specified intranet IP address.

Test rig 1 has been observed having slower dynamic response compared to Test rig 2 due to its apparatus design. Therefore, Test rig 1 was selected to be connected directly to the external reference input. However, both systems were set to a similar water level control range. Such setting is made so that both test rigs will be able to achieve any selected external reference value within this range. Both RPC blocks consist of output predictor and virtual controller transfer function that have been described in (34) can also be seen in the same figure. The network communication was accomplished through intranet connection using User Datagram Protocol/ Internet Protocol (UDP/IP) of the laboratory at the University of South Wales. Two intranet IP addresses 193.63.131.210 and 193.63.131.220 were set on two networked controller board for Test rig 1 and Test rig 2 respectively.

Throughout the experimentation process, a few significant challenges have been identified. In order to maintain the ideal and optimum test rigs performance, regular checking and servicing of the sensors and pumps on the test rig is required. This is important to ensure that the measured water level is accurate and the water flow delivered from the pump is consistent. Another challenge is on the characteristic of UDP network hardware which offers high speed but with unreliable data transfer. To rectify this problem, repetitive data collection is performed by resetting the network controller board frequently in order to ensure that the information or data exchanged has occurred.

D. Practical Experiments

Based on experimental setup in Fig. 6, 4000 samples of agent's output was collected from both Test rig 1 and Test rig 2 with PI controller and RPC. The experiment results are shown in Fig. 7 and 8.

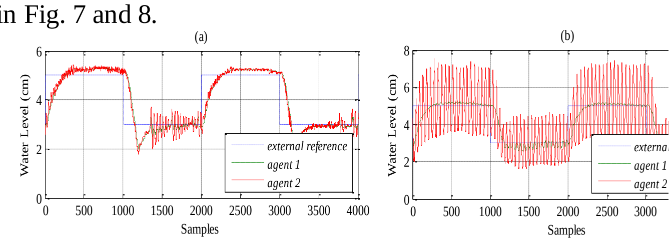

Fig. 7. Practical experiment of NMAS using PI controller (no delay compensation) with (a) 6-step delay ; (b) 7-step delay

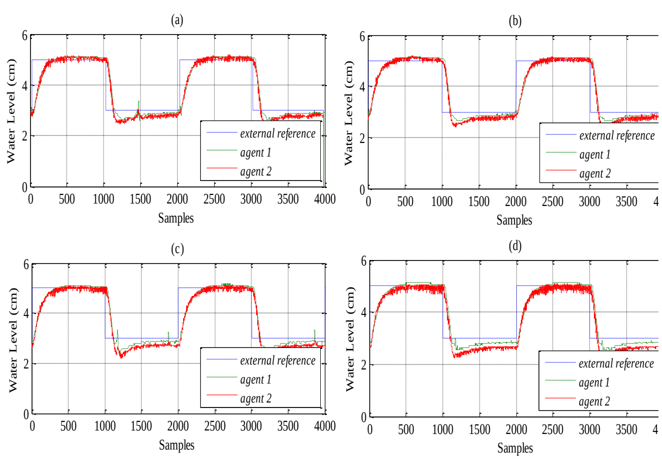

Fig. 8. Practical experiment of NMAS using RPC with (a) 6-step delay ; (b) 7-step delay ;(c) 8-step delay; (d) 9-step delay

From Fig. 7, it has been found that NMAS with the PI controller is unable to achieve its external consensus value at 7-step delay if it is executed without delay compensation. Due to this, the experiment has not been able to be repeated for 8-step and 9-step delay. In contrast, the proposed RPC has enabled to compensate the network delay actively and solved the external consensus problem in the experiment. The proposed RPC has also enabled to replicate the simulation result successfully. Through the test rigs experiments, it proves that the RPC is not only theoretically visible, but also practically reliable in test-rig applications. In addition, the estimated prediction error for each of NMAS agents is nearly equal to zero as shown in Fig. 9.

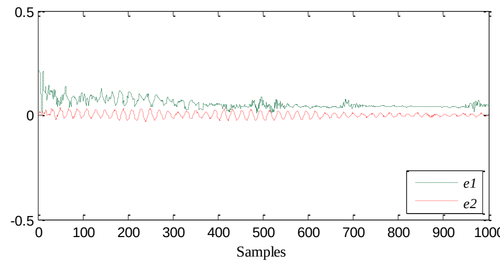

Fig. 9. Estimated prediction error at 9 step-delay (experiment)

Although it is hard to obtain the agent's model to closely represent the real test rig, the results of the practical experiments in this research are comparable to those of the real-time simulations. These results also prove that even though simulations can be relied on to estimate the possible outcome of a designed controller, an actual test rigs experiment will provide much accurate and better understanding of the controller capabilities.

VII. CONCLUSION

The design and stability analysis of external consensus protocol for inhomogeneous NMAS with constant network delay have been discussed. The RPC has been employed to compensate the output feedback network delay for NMAS's agents. The stability criterion has been obtained to ensure that NMAS is stable for the solved external consensus problem. In addition, a criterion on the selection of appropriate coupling gains between NMAS agents has been presented. In order to demonstrate the performance of the proposed protocol, numerical simulations and practical experiments have been conducted for NMAS which consist of two water level control test rigs with dissimilar dynamics and characteristics through intranet connection. It has been proven that the RPC has the ability to actively compensate the network delay for the proposed external consensus protocol.

REFERENCES

[1] R. Olfati Saber and R. M. Murray, "Consensus Protocols for Networks of Dynamic Agents", *Proceedings of American Control Conference*, pp 951-956, June, 2003.

[2] R. Olfati Saber and R. M. Murray, "Agreement Problems in Networks with Directed Graphs and Switching Topology",*Proceedings 42nd IEEE Conference on Decision and Control*,Vol. 4 , pp 4126 - 4132, December, 2003.

[3] 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, September, 2004.

[4] R. Olfati Saber, J.Alex Fax and R. M. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems",*Proceedings of IEEE*,Vol. 95 , No. 1, pp 215-233, January, 2007.

[5] G. Ferrari-Trecate, L. Galbusera, M. P. E. Marciandi and R. Scattolini, "A Model Predictive Control Scheme for Consensus in Multi-Agent Systems with Single-Integrator Dynamics and Input Constraints", *46th IEEE Conference on Decision and Control,* pp 1492-1497, December, 2007.

[6] J. Wu and Y. Shi, "Average Consensus in Multi-agent Systems with Time-varying Delays and Packet Losses", *American Control Conference*, pp 1579 - 1584, June, 2012.

[7] Kazunori Sakurama and Kazushi Nakano, "Average-consensus Problem for Networked Multi-agent Systems with Heterogenous Time-delays", *18th IFAC World Congress*, September 2011.

[8] A. Seuret, D. V. Dimarogonas and K. H. Johansson, "Consensus under Communication Delays", *Proceedings og the 47th IEEE Conference on Decision and Control,* December, 2008.

[9] C. Liu and Y. Tian, "Consensus of Multi-agent Systems with Diverse Communication Delays", *Proceedings of the 26th Chinese Control Conference,* July, 2007.

[10] P. A. Bliman, A. Nedic and A. Ozdaglar, "Rate of Convergence for Consensus with Delays", *Proceedings of the 47th IEEE Conference on Decision and Control,* December, 2008.

[11] Toru Namerikawa and Chika Yoshioka, "Consensus Control of Observer-based Multi-Agent System with Communication Delay", *SICE Annual Conference 2008,* August, 2008.

[12] G. Ferrari-Trecate, L. Galbusera, M. P. E. Marciandi and R. Scattolini, "Model Predictive Control Schemes for Consensus in Multi-Agent Systems with Integrator Dynamics and Time-varying Communication", *IEEE Transactions on Automatic Control,* Vol. 54, No. 11, pp 2560-2572, November, 2009.

[13] B. M. Nejad, S. A. Attia and J. Raisch, "Max-Consensus in a Max-Plus Algebraic Setting: The Case of Fixed Communication Topologies", *XXII International Symposium on Information, Communication and Automation Technologies,* pp 1-7, October, 2009.

[14] X. Wang and Y. Hong, "Distributed Finite-time χ-Consensus Algorithms for Multi-agent System with Variable Coupling Topology", *Journal of Systems Science and Complexity,* Vol. 23 , No. 2, pp 209 - 218, April, 2010.

[15] Jorge Cortes, "Distributed algorithms for reaching consensus on general functions",*Automatica,* Vol. 44, pp 726-737, 2008.

[16] 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 , pp 19 - 26, 2011.

[17] 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 , pp 945 - 949, 2012.

[18] Gabriel Rodrigues de Campos, Lara Brinon-Arranz, Alexandre Seuret and Silviu-Iulian Niculescu, "On the Consensus of Heterogenous Multi-agent Systems: a Decoupling Approach", *3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems,* 2012.

[19] E. S. Kazerooni and K. Khorasani, "Team Consensus for a Network of Unmanned Vehicles in Presence of Actuator Faults", *IEEE Transactions on Control Systems Technology,* Vol. 18, No. 5, September, 2010.

[20] X. Wang, L. Cheng, Z. Q. Chao, C. Zhou, M. Tan and Z. G. Hou, "Output-feedback Consensus Control of Linear Multi-agent Systems: A Fixed Topology", *International Journal of Innovative Computing, Information and Control,* Vol. 7, No. 5(A), May, 2011.

[21] J. Wang, D. Cheng and X. Hu, "Consensus of Multi-agent Linear Dynamic Systems", *Asian Journal of Control,* Vol. 10, No. 2, pp 144 - 155, March, 2008.

[22] J. Wang, Z. Liu and X. Hu, "Consensus of High Order Linear Multi-agent Systems Using Output Error Feedback", *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference,* December, 2009.

[23] J. Lee and J. S. Kim, "Discrete-time Consensus Problem using Optimal Control", *ICROS-SICE International Joint Conference,* August, 2009.

[24] S. C. Chai, J. X. Mu, G. P. Liu and D. Rees, "Design and Practical Implementation of Internet Based Predictive Control of a Servo System", *IEEE Transactions on Control Systems Technology*, Vol. 16, No. 1, pp 158-168, January, 2008.

[25] H. F. Scherer, E. Camponogara and A. Codas, "Transfer Function Modeling of Linear Dynamic Networks for Distributed MPC", *IEEE International Conference on Automation Science and Engineering,* pp 613 – 618, August, 2011.

[26] W. Ren, H. Chao, W. Bourgeous, N. Sorensen and Y. Chen, "Experimental Validation of Consensus Algorithm for Multivehicle Cooperative Control", *IEEE Transactions on Control Systems Technology*, Vol. 16, No. 4, July, 2008.

[27] Y. Cao, D. Stuart, W. Ren and Z. Meng, "Distributed Containment Control for Multiple Autonomous Vehicles With Double-Integrator Dynamics: Algorithms and Experiments", *IEEE Transactions on Control Systems Technology*, Vol. 19, No. 4, July 2011.

[28] Y. Zhu, X. Guan and X. Luo, "Finite-time consensus for multi-agent systems via nonlinear control protocols", *International Journal of Automation and Computing*, March 2013.

[29] H. Y. Yang, F. C. Wang and S. Y. Zhang, "Consensus of Second-order Multi-agent Systems with Nonsymmetric Interconnection and Heterogeneous Delays", November 2011.

[30] D. A. Spielman. (2009, September). The Laplacian: Spectral Graph Theory. [Online]. Available: http://www.cs.yale.edu/homes/spielman/561/2009/lect02-09.pdf

[31] C. Tan and G. P. Liu, "Consensus of Discrete-Time Linear Networked Multi-Agent Systems with Communication Delays", *IEEE Transactions on Automatic Control,* 2013.

[32] Z. Li, G. Wen, Z. Duan, and W. Ren. (2013, December). Designing Fully Distributed Consensus Protocols for Linear Multi-agent Systems with Directed Communication Graphs. *Arxiv.* [Online]. Available: http://arxiv.org/pdf/1312.7377.pdf.

[33] Z. Meng, Z. Li, A. V. Vasilakos, and S. Chen, "Delay-Induced Synchronization of Identical Linear Multiagent Systems", *IEEE transactions on Cybernetics,* Vol. 43, No. 2, April 2013.

[34] M. Fiedler, "Algebraic Connectivity of Graphs", *Czechoslovak Mathematical Journal,* Vol. 23, No. 98, 1973.

[35] T. Yamamoto and A. Namatame, "Networked Consensus Agents Dynamics through Matrix Eigenvalue," unpublished.

Nurul Adilla Mohd Subha received the B. Eng. in Control and Instrumentation Engineering and M. Eng. degrees in Mechatronics and Automatic Control from Universiti Teknologi Malaysia in 2007 and 2010, respectively. She is an academic staff at Department of Control and Mechatronics Engineering, Faculty of Electrical Engineering, University Teknologi Malaysia. Currently, she is on study leave at School of Engineering, University of South Wales for Ph.D degree with specialization on Consensus of Networked Multi-Agent System. Her research interest covers system identification, networked control, process control and multi-agent system.

Guo-Ping Liu received the BEng and MEng degrees from Central South University of Technology (now Central South University, China) in 1982 and 1985, respectively, and the PhD degree from UMIST (now University of Manchester, UK) in 1992. He is a professor of control engineering in the University of South Wales in the UK. He holds a visiting professorship at Harbin Institute of Technology in China. Prof Liu's research interests include networked control systems, multi-objective optimal control, nonlinear identification and control, and industrial advanced control applications. He is an author of 8 books, 2 edited books, over 200 journal papers and numerous conference papers. Prof. Liu is a fellow of IET. He serves as an editor-in-chief of the International Journal of Automation and Computing. IEEE Membership: M '97, SM '99, F' 11.