

# Distributed Collaborative Processing under Communication Delay over Wireless Sensor and Actuator Networks

L. Mo, B. Xu

**Lei Mo, Bugong Xu**

Key Laboratory of Autonomous Systems and Network Control,  
Ministry of Education  
College of Automation Science and Engineering  
South China University of Technology  
Guangzhou 510640, China  
E-mail: mo.lei@mail.scut.edu.cn, aubgxu@scut.edu.cn

## **Abstract:**

In wireless sensor and actuator networks (WSANs), the sensor nodes are involved in gathering information about the physical phenomenon, while the actuator nodes take decisions and then perform appropriate actions upon the environment. The collaborative operation of sensor and actuator nodes brings significant advantages over WSNs, including improved accuracy and timely actions upon the sensed phenomena. However, unreliable wireless communication and finding a proper control strategy cause challenges in designing such network control system. In order to accomplish effective sensing and acting tasks, efficient coordination mechanisms among different nodes are required. In this paper, the coordination and communication problems in WSANs are studied. First, we formulate the mathematical models for the WSANs system. Then, a predictor-controller algorithm based on distributed estimation is adopted to mitigate the effects of network-induced delay. Finally, we apply a collaborative processing mechanism to meet the desired system requirements and improve the overall control performance. This approach will group the sensor and actuator nodes to work in parallel so as to reduce the computation complexity and enhance the system reacting time. Simulation results demonstrate the effectiveness of our proposed method.

**Keywords:** Wireless sensor and actuator networks, Distributed estimation, Collaborative processing.

## 1 Introduction

Wireless sensor and actuator networks(WSANs) comprise groups of sensor and actuator nodes that are connected with wireless medium. It is an important extension of wireless sensor networks (WSNs), allowing actuator nodes within the network to make autonomous decisions and then perform appropriate actions in response to the sensor nodes measurements [1]. Thus, the novel network architecture performs not only 'read' operations, but also 'write' operations, which brings about unique and new challenges that need to be addressed [2]. In order to satisfy the requirements introduced by the coexistence of sensor and actuator nodes, multiple coordination levels among nodes are required to implement, which can be defined as: Sensor-Sensor(S-S), Sensor-Actuator(S-A) and Actuator-Actuator(A-A). The S-S coordination is similar to the scheme already used in wireless sensor networks applications. Thus, in this paper we mainly focus on the S-A and A-A coordination.

Due to the unreliable wireless communications, system noise and time-delay are the common phenomenons which will influence the overall system performance. To this end, it is quite necessary for the nodes to perform the estimation and compensation algorithms of the required information [3]. A queuing strategy is introduced both in controllers and actuator nodes in [4], and the time delay between controllers and actuator nodes is compensated by multi-step control increment given by the algorithm of general predictive control. The work given in [5]

presents a real-time architecture for automated WSAWs, the delay bound of S-A communication is maintained by the distributed mechanisms for S-A event reporting and self-aware coordination. In [6] a general reliability-centric framework for event reporting in WSAWs is proposed, which contains an fault-tolerant event data aggregation algorithm and a delay-aware data transmission protocol. In [7], the authors study the model-based predictive networked control systems that compensate random delays and data loss of the communication, and use a predictive control scheme to avoid performance loss. Our work is motivated by the above studies. The key difference is that we focus on the S-A delay, not the uniform network nodes delay. Moreover, we apply a predictor-controller algorithm based on the state estimation to mitigate the detrimental effects of the communication delay. In this context, the model of the WSAWs system needs to be analyzed in detail.

Finding a proper control strategy is still the core in designing the A-A coordination [8], This process involves which actuator node should be scheduled to execute a specific task and how to adjust its actuation to meet the desired system requirements. According to the way data is routed among different actuator nodes, control strategies can be categorized into the distributed control (DC) and centralized control (CC) scheme [9]. In the DC scheme, the control decision of a signal actuator node relies on the local information received from its neighbor nodes rather than global information [10]. Then it can achieve a superior performance in modularity, integrated diagnostics, quick and easy maintenance and low cost. In [11] a framework of optimizing a collaborative sensing and actuation system is built for environment control, the sensor is set in the actuator node and the control objection is to balance the energy saving against the spatial smoothness of the control signals. In [12], the authors propose two control schemes in WSAWs for building-environment control systems, a CC scheme in which control decisions are made based on global information, and a DC scheme that enables distributed actuator nodes to make decisions locally. In [8] a new distributed estimation and collaborative control scheme is proposed for industrial control systems with WSAWs, which can achieve robust control against inaccurate system parameters. In this paper, we focus on the problem of utilize distributed sensor measurements to design control strategies in order to elicit a desired response from the monitored environment. Our methodology incorporates a dynamic clustering schedule into the collaborative estimation and control framework, which can minimize the control error and improve the control quality.

The remainder of this paper is organized as follows: Section 2 models the WSAWs system. While Section 3 provides a delay compensation algorithm. Then a distributed collaborative proceeding method is designed in Section 4. At last, the results of simulations conducted to explore the performance of proposed algorithms are demonstrated in Section 5.

## 2 System Models

We consider the WSAWs system that are employed to the industrial instrumentation and control applications. The control objective is to adjust the system variables to meet our requirements. A set of static sensor and actuator nodes that are spread throughout the region of interest (ROI) to detect and track events and take necessary actions. Let  $x$  denote the system variable of our concern, such as temperature, brightness, humidity, sound, pressure, vibrations, etc. in different parts of the field. Let  $S^A$  represent the set of actuator nodes, with  $n_a = |S^A|$ . Let  $S^S$  represent the set of sensor nodes, with  $n_s = |S^S|$ . We make the following assumption of our network: (1) Sensor node is the time-driven device, input reception or output transmission is controlled by a sample time, while the actuator node is the event-driven device depends on the control techniques used; (2) Sensor and actuator nodes are aware of their geographical position; (3) The network is synchronized by means of one of the existing synchronization protocols [13];

(4) The randomly varying delays between S-A are bounded; (5) The system is observable and controllable.

Let  $S_i$  denote the  $i$ th sensor node, each node has the following model:

$$z_i(k) = h_i x_i(k) + \nu_i(k), i = 1, \dots, n_s \quad (1)$$

where  $h_i$  and  $\nu_i(k)$  are the observation item and measurement noise, respectively. Assume that  $\nu_i(k)$  is a zero-mean Gaussian white noise with  $E\{\nu_i(k)\} = 0$ ,  $E\{\nu_i(k)\nu_j^T(l)\} = r_i(k)\delta_{kl}\delta_{ij}$ , where  $\delta_{kl} = 1$  if  $k = l$ , and  $\delta_{kl} = 0$ , otherwise. Then the matrix form of Eq.(1) is:

$$Z(k) = HX(k) + \nu(k) \quad (2)$$

where  $Z(k) = [z_1(k), \dots, z_{n_s}(k)]^T$ ,  $H = \text{diag}[h_1, \dots, h_{n_s}]$ ,  $X(k) = [x_1(k), \dots, x_{n_s}(k)]^T$  and  $\nu(k) = [\nu_1(k), \dots, \nu_{n_s}(k)]^T$ .

Let  $A_j$  denote the  $j$ th actuator node,  $f_j$  denote its output which influences its ambient plant state,  $u_j$  denote the control signal that is used to adjust  $A_j$ 's actuation. The change of each actuator node's actuation is assumed linearly proportional to the control signal received by this node, which is modeled as:

$$f_j(k) = g u_j(k), j = 1, \dots, n_a \quad (3)$$

where  $g$  is the transfer function of  $A_j$ . Here, we consider a scenario with homogenous actuator nodes. Then the matrix form of Eq.(3) is:

$$F(k) = GU(k) \quad (4)$$

where  $F(k) = [f_1(k), \dots, f_{n_a}(k)]^T$ ,  $G = \text{diag}[g_1, \dots, g_{n_a}]$  and  $U(k) = [u_1(k), \dots, u_{n_a}(k)]^T$ .

Here, we used two sets to indicate the interaction between the sensor and actuator nodes: the associated sensor nodes of  $A_i, \forall i \in \{1, \dots, n_a\}$ :

$$S_{A_i} = \{S_j | d_{ij} \neq 0, j = 1, \dots, m_s\} \quad (5)$$

and the influenced actuator nodes of  $S_j, \forall j \in \{1, \dots, n_s\}$ :

$$S_{S_j} = \{A_i | d_{ij} \neq 0, i = 1, \dots, m_a\} \quad (6)$$

where the parameter  $d_{ij}$  represents the relation between the  $i$ th and the  $j$ th nodes:

$$d_{ij} = \begin{cases} 1, & \text{influenced} \\ 0, & \text{isolated} \end{cases} \quad (7)$$

Eq.(5) and Eq.(6) show that the sensor nodes in set  $S_{A_i}$  will transmit the sensing data to  $A_i$ , while the actuator nodes in set  $S_{S_j}$  will influence the plant state monitored by  $S_j$ .

In WSANs, the sensor and actuator nodes are usually linked with wireless medium, since the actuator nodes are connected with each other directly and much more powerful than the ordinary sensor nodes, communication delay between S-A become a general problem of such network control system [14]. The S-A delays do not only degrade the system performance, but can also destabilize the system [15]. The delay system at sample step  $k$  has the following dynamics:

$$Z_a(k) = Z(k - \Delta_k) \quad (8)$$

The finite non-negative integers  $\Delta_k$  represent the S-A delays at the  $k$ th step,  $Z_a(k)$  is the sensing data received by the actuator nodes with communication delays.

In the course of the practice, the variation of plant state  $x_i$  at time  $t$  is caused by the output transferred from the actuator nodes and its ambient environment [12], under that assumption we have:

$$\frac{dx_i}{dt} = \sum_{1 \leq k \leq n_s, k \neq i} \alpha_{ki}(x_k - x_i) + \sum_{1 \leq l \leq n_a} \beta_{li}(f_l - x_i) \quad (9)$$

where  $\alpha_{ki}$  and  $\beta_{li}$  are coefficients relating to the state-transfer efficiency. So, the plant state equation can be written in the matrix form as follows:

$$\frac{dX}{dt} = \Phi X(t) + \Psi F(t) \quad (10)$$

where  $\Phi \in R^{n_s \times n_s}$ ,  $\Psi \in R^{n_a \times n_a}$ , we assume that  $F$  is constant within each step, i.e.,  $F(t) = F(k), t \in [kT, (k+1)T)$ . Since  $\Phi$  and  $\Psi$  are coefficients or constants, then the dynamic system can be modeled as:

$$X(k+1) = AX(k) + BF(k) \quad (11)$$

where  $A = e^{\Phi T}$  and  $B = \Phi^{-1}(e^{\Phi T} - 1)\Psi$ .

### 3 Delay Compensation Algorithm

The main idea of the delay compensation algorithm is to utilize an observer to estimate the plant states and a multi-step predictor to compute predictive control inputs based on the past sensor measurements. The block diagram of the delay compensation algorithm is shown in Figure 1.

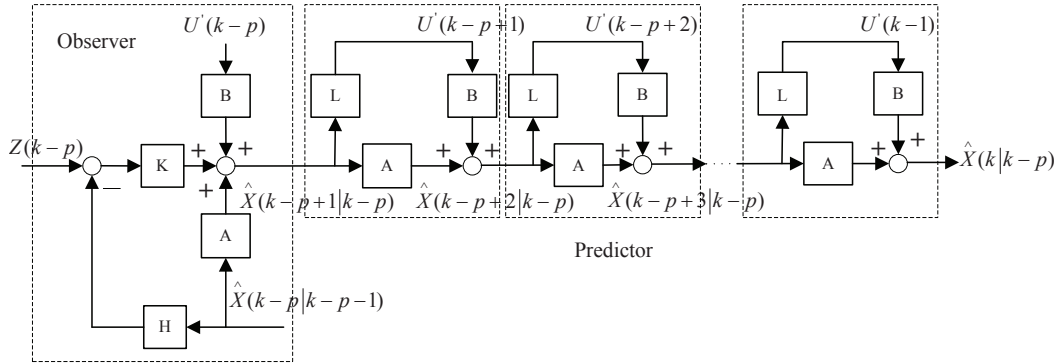


Figure 1: Block diagram of the delay compensation algorithm.

In order to keep the track of past measurements, received sensing data have to be stored in a  $p$  length FIFO (First-In-First-Out), denoted as  $Q$ , and  $p$  is the upper bound of  $\Delta_k$ . Thus, the S-A delay is transformed to a constant delay, which is much easier to control than the random delay systems [15]. The delay compensation algorithm is delineated below:

Observer model:

$$\hat{X}(k-p+1 | k-p) = A\hat{X}(k-p | k-p-1) + BU'(k-p) + K(k-p)(Z(k-p) - H\hat{X}(k-p | k-p-1)) \quad (12)$$

Predictor model:

$$\hat{X}(k | k-p) = A\hat{X}(k-1 | k-p) + BU'(k-1) \quad (13)$$

Control law:

$$U'(k) = L(k)\hat{X}(k | k - p) \quad (14)$$

In [16], the authors have proved that the resulting closed-loop equations can be expressed as:

$$\begin{bmatrix} X(k+1) \\ \hat{E}(k-p+1) \end{bmatrix} = \begin{bmatrix} A + BL(k) & * \\ 0 & A - K(k-p)H \end{bmatrix} \begin{bmatrix} X(k) \\ \hat{E}(k-p) \end{bmatrix} \quad (15)$$

where:

$$\hat{E}(k) = X(k) - \hat{X}(k | k - 1) \quad (16)$$

If  $K$  and  $L$  are constant, then Eq.(15) determines the stability of the delay compensator due to the separation of the controller and observer. Since the performance of the observer and predictor are highly dependent on the model certainty, then the dynamic model of the plant has to be very precise.

## 4 Distributed Collaborative Processing

### 4.1 Dynamic clustering schedule

In order to maximize the network lifetime and data throughput, and provide load balancing and fault tolerance [17], the clustering schedule should be established. In this paper, based on the characteristics of the current events, an event-triggered dynamic clustering schedule is designed for WSAWs. If there is no event occurs, the nodes follow a static sleep schedule. When an event occurs, the sensor nodes whose sensing range cover it will be activated, and transmit the sensing data to each coordinator, then the coordinator organizes its neighbor nodes into a working cluster to take a proper action till the error signal becomes zero. During this process, the coordinator will act as the cluster head and the neighbor nodes will be selected as the cluster members. Then the control decision is made by the cluster head according to the fusion data aggregate from the associated sensor node and cluster members. For sensor node's coordinator is the nearest actuator node, since the closer the actuator node to the sensor node is, the earlier the actuator node is informed, thus the quicker the actuator node reacts and the earlier action to be initiated. Here, neighbor nodes can be defined as the actuator nodes which are within the communication range of coordinator and the associated sensor nodes are activated. So, the energy constrained sensor node does not need to transmit its readings to multiple actuator nodes. Instead, the coordinator will receive this message and relay it to its neighbor nodes to come up with an appropriate actuation. The process of dynamic clustering schedule is shown in Figure 2. Here, we assume that the data route from source sensor node to terminal actuator node is in one hop,  $|S_{S_j}| = 1, j = 1, \dots, n_s$  and  $|S_{A_i}| = 1, i = 1, \dots, n_a$ .

### 4.2 Collaborative processing algorithm

Consider the control objective which is to meet the set points  $X^* = [x_1^*, \dots, x_{n_s}^*]^T$ . In order to balance the control requirements against the spatial smoothness of the control signals, we define the control objective of  $A_i$  as:

$$J_i(k+1) = \frac{\alpha}{2} \sum_{j \in N_{ai}, j \neq i} (e_i(k+1) - e_j(k+1))^2 + \frac{1-\alpha}{2} e_i^2(k+1) \quad (17)$$

where  $N_{ai}$  is the neighbor nodes set of  $A_i$ ,  $e_i(k+1) = x_i(k+1) - x_i^* = a_{ii}\hat{x}_i(k) + b_{ii}g_i u_i(k) - x_i^*$  and  $e_j(k+1) = x_j(k+1) - x_j^* = a_{jj}\hat{x}_j(k) + b_{jj}g_j u_j(k) - x_j^*$ , ( $j \in N_{ai}, j \neq i$ ).

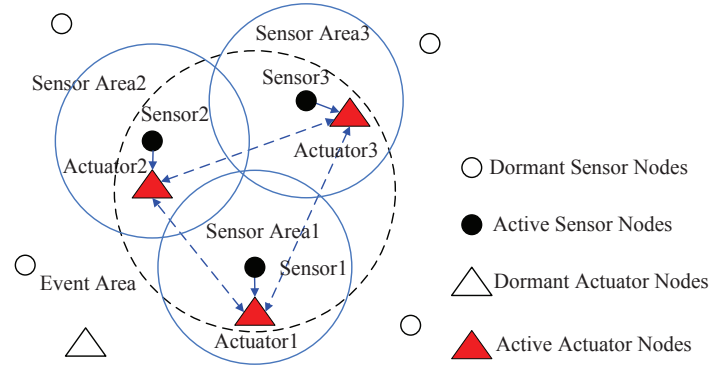


Figure 2: Dynamic clustering schedule.

In order to minimize  $J_i(k+1)$ , the gradient descending method can be used. The partial derivative of  $J_i(k+1)$  with respect to  $u_i(k)$  is calculated as:

$$\frac{\partial J_i(k+1)}{\partial u_i(k)} = [\alpha \sum_{j \in N_{ai}, j \neq i} (e_i(k) - e_j(k)) + (1 - \alpha)e_i(k)] b_{ii} g_i \quad (18)$$

For each  $A_i, i \in (1, \dots, n_a)$ , its control law  $u_i(k)$  is updated by:

$$u_i(k+1) = u_i(k) + \Delta u_i = u_i(k) - \varepsilon \frac{\partial J_i(k+1)}{\partial u_i(k)} \quad (19)$$

where  $\varepsilon$  is a positive step size called the learning step length. If  $\varepsilon$  is small, the convergence speed of the objective function  $J_i$  will be slow. If  $\varepsilon$  is too large, it often leads to unstable. So it is important how to choose the proper  $\varepsilon$ .

In order to investigate the stability of Eq.(19), we rewrite  $\Delta u_i$  as:

$$\Delta u_i = -\varepsilon \frac{\partial J_i(k+1)}{\partial u_i(k)} = -\varepsilon \frac{\partial J_i(k+1)}{\partial e_i(k+1)} \frac{\partial e_i(k+1)}{\partial u_i(k)} = -\varepsilon b_{ii} g_i \frac{\partial J_i(k+1)}{\partial e_i(k+1)} = -\lambda_i \frac{\partial J_i(k+1)}{\partial e_i(k+1)} \quad (20)$$

We define the learning error as:

$$\Delta e_i = -\lambda_i \frac{\partial J_i(k+1)}{\partial e_i(k+1)} \quad (21)$$

where

$$\frac{\partial J_i(k+1)}{\partial e_i(k+1)} = \alpha \sum_{j \in N_{ai}, j \neq i} (e_i(k+1) - e_j(k+1)) + (1 - \alpha)e_i(k+1) \quad (22)$$

Let the array  $[\partial J_1(k+1)/\partial e_1(k+1), \dots, \partial J_{n_a}(k+1)/\partial e_{n_a}(k+1)]^T$  to be zero, then it can be represented as:

$$DE(k+1) = 0 \quad (23)$$

Here,  $D$  is a  $n_a \times n_a$  positive definite matrix,  $E(k+1) = [e_1(k+1), \dots, e_{n_a}(k+1)]^T$  and the elements of  $D$  satisfy the following equation:

$$|d_{ii}| - \sum_{j=1, j \neq i}^{n_a} |d_{ij}| = 1 - \alpha \quad (24)$$

We define the residual error as:  $R(k) = E(k) - E^*$ , where  $E^*$  is the solution of  $DE^* = 0$ . From Eq.(21), we have:

$$\begin{aligned} R(k) &= R(k-1) - \lambda(DE(k-1)) = R(k-1) - \lambda(DE(k-1) - DE^*) = (I - \lambda D)R(k-1) \\ &= \prod_{i=1}^k (I - \lambda D)R(0) = Y \prod_{i=1}^k (I - \lambda \Lambda)Y^T R(0) = Y \prod_{i=1}^k (I - \varepsilon \Lambda')Y^T R(0) \end{aligned} \quad (25)$$

where  $D = Y\Lambda Y^T$ ,  $\Lambda = \text{diag}(\eta_1, \dots, \eta_{n_a})$ ,  $\lambda = \text{diag}(\lambda_1, \dots, \lambda_{n_a})$ , and  $\eta_1, \dots, \eta_{n_a}$  are the eigenvalues of  $D$ , so we can get:

$$\Lambda' = B G \Lambda = \text{diag}(b_{11}g_1\eta_1, \dots, b_{n_a n_a}g_{n_a}\eta_{n_a}) = \text{diag}(\sigma_1, \dots, \sigma_{n_a}) \quad (26)$$

If we select  $0 < \varepsilon < 2/\max(\sigma_i), 1 \leq i \leq n_a$ , then  $R(k) \rightarrow 0$  as  $k \rightarrow \infty$ .

Eq.(19) is a completely distributed collaborative processing method, there has no need a sink to help in the coordination of the sensor and actuator nodes. Instead, each actuator node combines itself and neighbor nodes' messages to access the control law and pursuit the optimal solutions step by step.

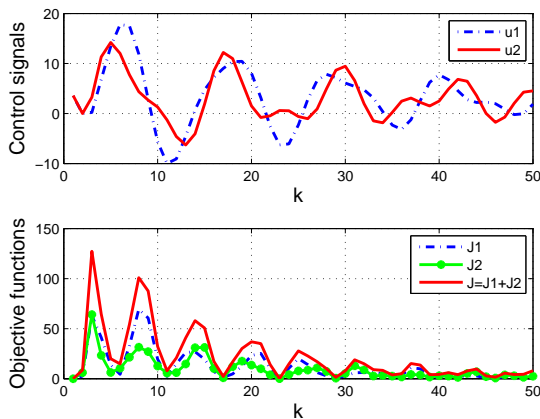
## 5 Numerical Examples

Let's consider a simple Humidity, Ventilation, Air Conditioning (HVAC) control system for temperature control with two sensor nodes ( $n_s = 2$ ) and two actuator nodes ( $n_a = 2$ ). The control arm is to meet the set points  $X^* = [16(^{\circ}C), 18(^{\circ}C)]$ . The system parameters are:

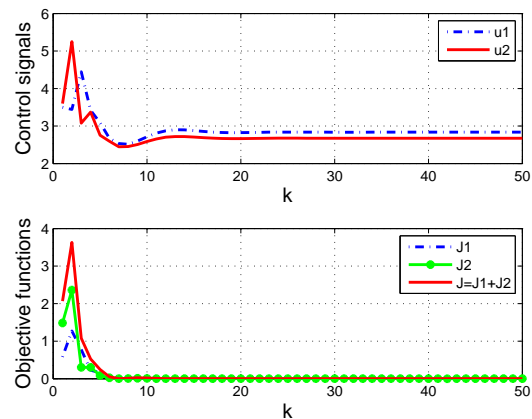
$$A = \begin{bmatrix} 0.9 & 0 \\ 0 & 0.9 \end{bmatrix}, B = \begin{bmatrix} 0.57 & 0 \\ 0 & 0.68 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, K = \begin{bmatrix} -0.68 & 0 \\ 0 & -0.57 \end{bmatrix}, L = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$$

where  $K$  and  $L$  satisfy the stability condition according to Eq.(15).

The effectively actuation of actuator nodes are highly depend on the precision of the sensing data. The longer time delay between sensing and acting is, the bigger estimation error introduced. The increasing control decision error does not only degrade the system performance, but also can destabilize the system, just as shown in Figure 3(a) and 3(c). Figure 3(b) and 3(d) clearly show that the compensated system are less oscillatory than those of the uncompensated system. The predictor-controller compensation algorithm provides a valid way to estimate the sensing data with latency, reduce the estimation bias and enhance the precision of feedback control.



(a)  $\varepsilon = 1.63, \alpha = 0.5, \Delta_k = 1, p = 0$



(b)  $\varepsilon = 1.63, \alpha = 0.5, \Delta_k = 1, p = 1$

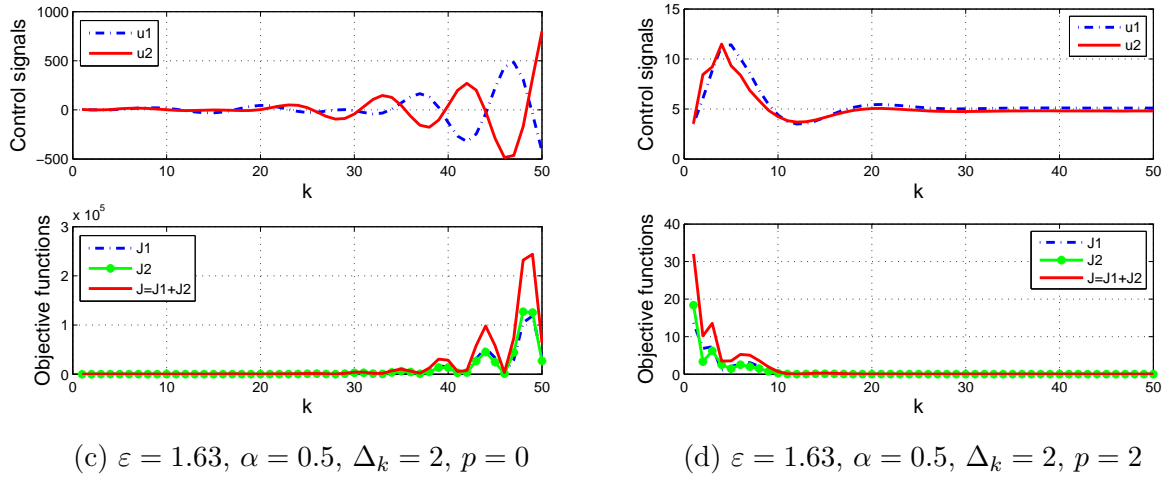
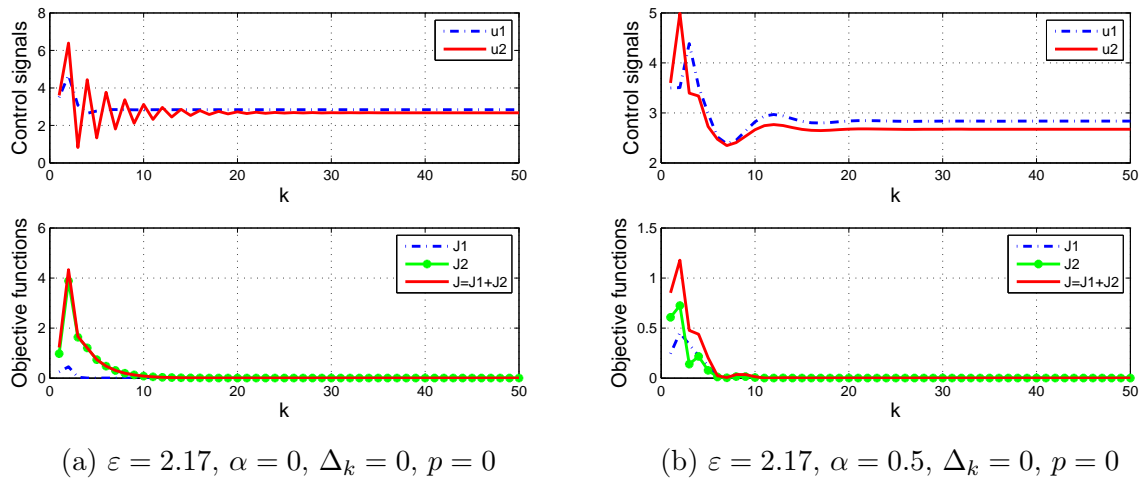
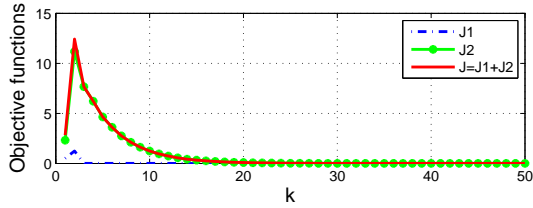
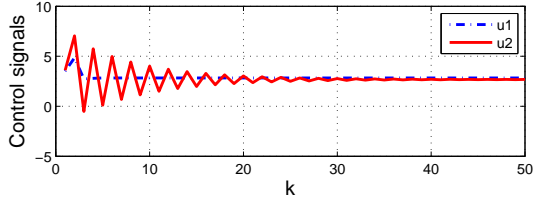


Figure 3: Dynamic responses of uncompensated ( $\Delta_k \neq 0, p = 0$ ) and compensated system ( $p = \Delta_k \neq 0$ ).

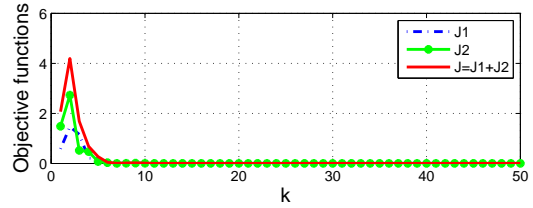
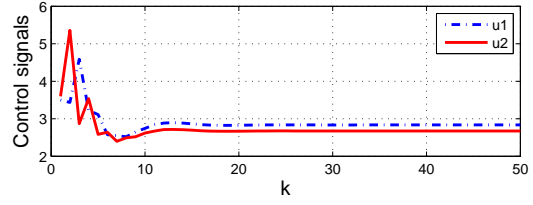
In Eq.(17),  $\alpha$  is a collaborative factor between 0 and 1. When  $\alpha = 0$ , the neighbor nodes' messages are not taken into consideration, but if we select  $\alpha \neq 0$ , the collaborative processing among different nodes are introduced. Moreover,  $\alpha$  can also performs as a smooth factor, it will reduce the control overshooting and stabilize the system from oscillating. The performance of compensated system with and without collaborative processing method are shown in Figure 4. It is obviously seen that the proposed method can greatly improve the system performance, which can smooth the actuator control signal and accelerate the system convergence speed.



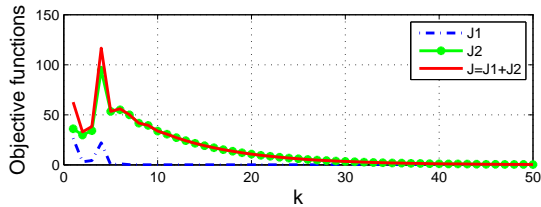
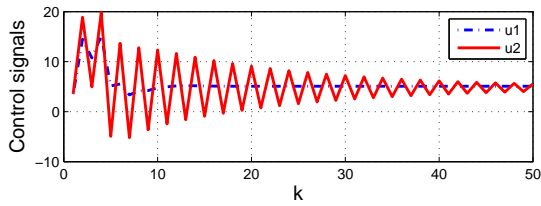




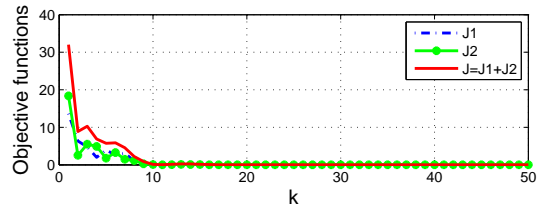
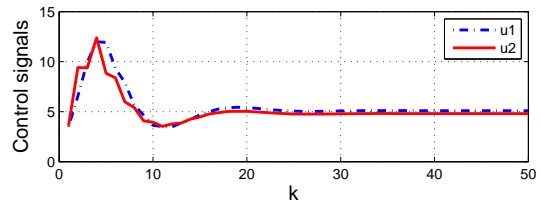
(c)  $\varepsilon = 1.74, \alpha = 0, \Delta_k = 1, p = 1$



(d)  $\varepsilon = 1.74, \alpha = 0.5, \Delta_k = 1, p = 1$

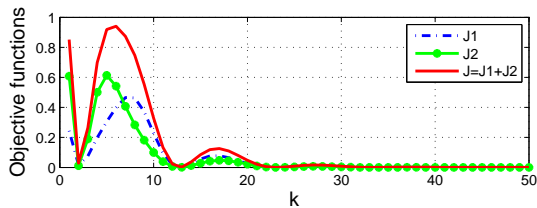
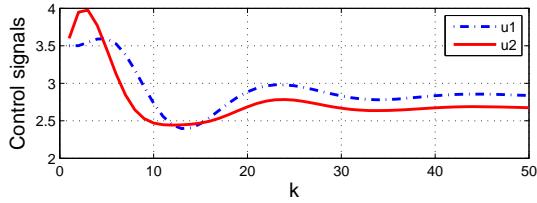


(e)  $\varepsilon = 1.96, \alpha = 0, \Delta_k = 2, p = 2$

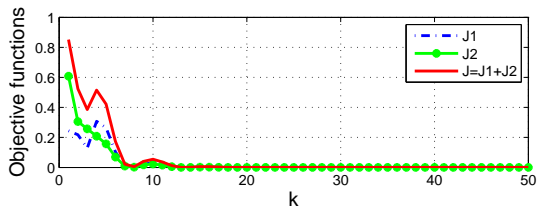
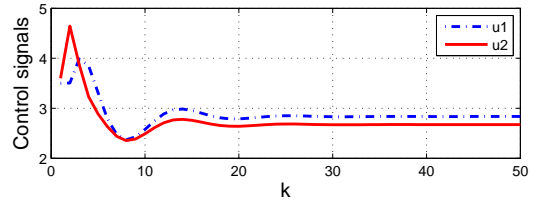


(f)  $\varepsilon = 1.96, \alpha = 0.5, \Delta_k = 2, p = 2$

Figure 4: Dynamic responses of real-time ( $\Delta_k = 0$ ) and compensated system ( $p = \Delta_k \neq 0$ ) under different  $\alpha$ .



(a)  $\varepsilon = 0.54, \alpha = 0.5, \Delta_k = 0, p = 0$



(b)  $\varepsilon = 1.63, \alpha = 0.5, \Delta_k = 0, p = 0$

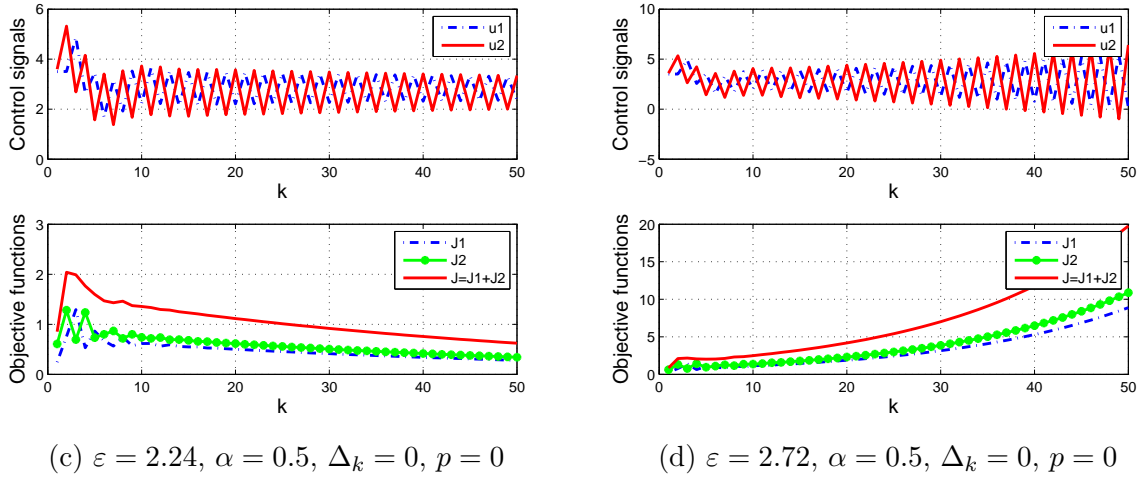


Figure 5: Dynamic responses of real-time system ( $\Delta_k = 0$ ) under different  $\varepsilon$ .

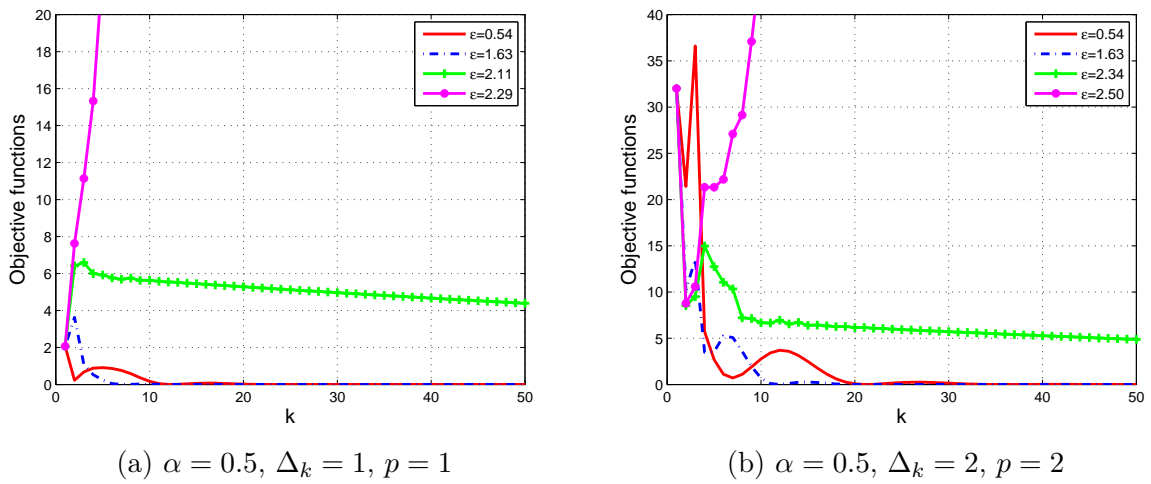


Figure 6: Dynamic responses of compensated system ( $p = \Delta_k \neq 0$ ) under different  $\varepsilon$ .

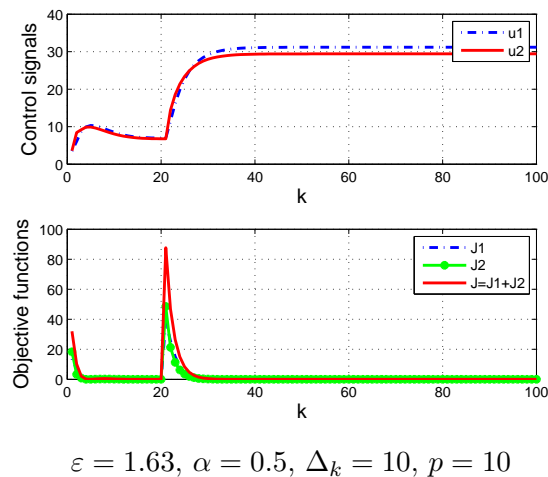


Figure 7: Dynamic responses of system with multi-step delay compensation.

The learning step length  $\varepsilon$  plays an important role in the gradient descending method. Figure 5 compares the responses of real-time system under different  $\varepsilon$  ( $\varepsilon = 0.54, 1.63, 2.24, 2.72$ ). We could observe that the bigger  $\varepsilon$  is, the faster convergence speed of  $J$  can be achieved. But if  $\varepsilon$  exceeds the critical value  $\varepsilon_c = 2.24$ , i.e.,  $\varepsilon = 2.72$ , the system become unstable. Figure 6 shows the dynamic responses of compensated system with collaborative processing method under different  $\varepsilon$ .  $\Delta_k$  between S-A are set at one-step and two-step, respectively. We can get that  $\varepsilon_c = 2.11$  and  $\varepsilon_c = 2.34$  are the critical step lengths, and the stable control can be achieved within those values.

Comparing Figure 6(a) and 6(b), we could see that, the bound of  $J$  is influenced by the delay step  $\Delta_k$ . The variance of  $J$  increased as the longer S-A latency. Figure 7 shows that when the system suffers a multi-step delay, such as  $\Delta_k = 10$ , the proposed compensation and control scheme is also useful.  $J$  will tend to be zero eventually and control signals both converge to their stable states.

## 6 Conclusions

In this paper, we focus on the communication and control problems in WSAWs. We argue that the system performance is closely related to the communication delay and control strategy. In order to mitigate the detrimental effects of the S-A latency, a delay compensation algorithm based on the state estimation is applied to this system. Then, a distributed collaborative processing method is proposed to control actuator option in a coordinate way to accomplish the desired tasks. We formulate it as an optimization problem and utilize gradient descending algorithm to calculate the optimal control law for actuator nodes. On this basis, we discuss the control strategy parameters that relate to the system performance and provide a guide line how to choose properly. In our framework, the proposed collaborative processing method does no need a central sever and make an optimum usage of the available resources, which can be easily applied in the industrial automation systems.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant 61174070, the Specialized Research Fund for the Doctoral Program under grant 20110172110033.

## Bibliography

- [1] I. F. Akyildiz, I. H. Kasimoglu, Wireless Sensor and Actor Networks: Research Challenges. *Ad Hoc Networks*, Vol.2, No.4, pp.351-367, 2004.
- [2] R. Vedantham, Z. Zhuang, R. Sivakumar, Hazard Avoidance in Wireless Sensor and Actor Networks. *Computer Communications*, Vol.29, No.13, pp.2578-2598, 2006.
- [3] A. Deshpande, C. Guestrin, S. R. Madden, Resource-Aware Wireless Sensor-Actuator Networks. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, Vol.28, No.1, pp.40-47, 2005.
- [4] T. Wang, L. Zhou, P. Han, Q. Zhang, Complete Compensation for Time Delay in Networked Control System based on GPC and BP Neural Network. *Proceedings of 2007 International Conference on Machine Learning and Cybernetics*, Hongkong, China, 637-641, 2007.

- 
- [5] Y. Zeng, C. J. Sreenan, G. Zheng, A Real-time Architecture for Automated Wireless Sensor and Actuator Networks. *The Fifth International Conference on Wireless and Mobile Communications*, Cannes/La Bocca, French Riviera, France, 1-6, 2009.
- [6] E. Ngai, Y. Zhou, M. R. Lyu, J. Liu, A Delay-aware Reliable Event Reporting Framework for Wireless Sensor-Actuator Networks. *Ad Hoc Networks*, Vol.8, No.7, pp.694-707, 2010.
- [7] A. Onat, T. Naskali, E. Parlakay, O. Mutluer, Control Over Imperfect Networks: Model-Based Predictive Networked Control Systems. *IEEE Transactions on Industrial Electronics*, Vol.58, No.3, pp.905-913, 2011.
- [8] J. Chen, X. Cao, P. Cheng, Y. Xiao, Y. Sun, Distributed Collaborative Control for Industrial Automation With Wireless Sensor and Actuator Networks. *IEEE Transactions on Industrial Electronics*, Vol.57, No.12, pp.4219-4229, 2010.
- [9] R. I. Erica, V. G. Luis, Cooperation Mechanism Taxonomy for Wireless Sensor and Actor Networks. *IEEE Transactions on Industrial Electronics*, Vol.264, pp.62-73, 2008.
- [10] T. Melodia, D. Pompili, V. C. Gungor, I. F. Akyildiz, Communication and Coordination in Wireless Sensor and Actor Networks. *IEEE Transactions on Mobile Computing*, Vol.6, No.10, pp.1116-1128, 2007.
- [11] M. Nakamura, A. Sakurai, S. Furubo, H. Ban, Collaborative Processing in Mote-Based Sensor/actuator Networks for Environment Control Application. *Signal Processing*, Vol.88, No.7, pp.1827-1838, 2008.
- [12] X. Cao, J. Chen, Y. Xiao, Y. Sun, Building-Environment Control With Wireless Sensor and Actuator Networks: Centralized Versus Distributed. *IEEE Transactions on Industrial Electronics*, Vol.57, No.11, pp.3596-3606, 2010.
- [13] B. Sundararaman, U. Buy, A. Kshemkalyani, Clock Synchronization for Wireless Sensor Networks: A Survey. *Ad Hoc Networks*, Vol.3, No.3, pp.281-323, 2005.
- [14] V. C. Gungor, Ö. B. Akan, I. F. Akyildiz, A Real-time and Reliable Transport Protocol for Wireless Sensor and Actor Networks. *IEEE Transactions on Networking*, Vol.16, No.2, pp.359-370, 2008.
- [15] M. Chow, Y. Tipsuwan, Network-Based Control Systems: A Tutorial. *The 27th Annual Conference of the IEEE Industrial Electronics Society*, Denver, USA, pp.1593-1602, 2001.
- [16] R. Luck, A. Ray, Experimental Verification of a Delay Compensation Algorithm for Integrated Communication and Control Systems. *International Journal of Control*, Vol.59, No.6, pp.1357-1372, 1994.
- [17] B. McLaughlan, K. Akkaya, Coverage-based Clustering of Wireless Sensor and Actor Networks. *IEEE International Conference on Pervasive Services*, Istanbul, Turkey, 45-54, 2007.