Collaborative Estimation and Actuation for Wireless Sensor and Actuator Networks ⋆

Lei Mo, **Xianghui Cao***, **Jiming Chen**, and **Youxian Sun**

*State Key Lab. of Industrial Control Technology, Zhejiang University,
Hangzhou 310027, China.*
Emails: morleyzs@gmail.com, {xhcao, jmchen, yxsun}@iipc.zju.edu.cn
* *Corresponding author*

Abstract: Wireless sensor and actuator networks (WSANs) facilitate close interactions between human and the environment. Efficient coordination among the sensors and actuators plays a vital role in carrying out sensing and acting in WSANs. In this paper, we develop a collaborative estimation and actuation mechanism, which consists of a sensor-actuator coordination phase and an actuator-actuator coordination phase. The first phase is based on distributed Kalman filter in federated configuration, which is able to provide reliable and precise sensing data. On this basis, the second phase allocates proper tasks based on system requirements and coordinates actuators to accomplish the tasks. Particularly, the actuator-actuator coordination is formulated as an optimization problem and an effective method is proposed to search the solution based on sequential unconstrained minimization technique. Simulation results demonstrate the effectiveness of our proposed mechanism.

Keywords: Wireless sensor and actuator networks, node coordination, federated Kalman filter, actuator task allocation, sequential unconstrained minimization technique

1. INTRODUCTION

Wireless sensor and actuator networks (WSANs) consist of a number of sensors and actuators to enable close interactions between human and environment (Akyildiz and Kasimoglu, 2004). WSANs can be applied in many areas, such as fire detection in forest, home automation and precision agriculture (Stankovic, 2008). Due to the coexistence of sensors and actuators, coordination is demanded at different levels of WSANs, which can be categorized into Sensor-Sensor (S-S), Sensor-Actuator (S-A) and Actuator-Actuator (A-A) coordinations (Ruiz-Ibarra and Villasenor-Gonzalez, 2008). Extensive studies have been carried out to address the S-S coordination in collaborative sensing, cooperative transmission, sensor scheduling, etc., in the context of wireless sensor networks (WSNs) (Akyildiz et al., 2002; Yuan et al., 2006). In this paper, however, we mainly focus on S-A and A-A coordinations, which are the key differences between WSANs and WSNs.

The S-A coordination manages sensors to sense the physical world and transmit sensed information to appropriate actuators (Wu, 2011). Most of existing works in this area involves data aggregation and data transmission (Gungor et al., 2008; Ngai et al., 2010; Nakayama et al., 2011). The design of S-A coordination is challenging due to: (1) the amount, the resource and the traffic load are asymmetrical between the sensors and the actuators, (2) the sensor information is usually corrupted due to factors such as noise and sensor failure, and (3) the system must satisfy the real-time requirement. To address these issues, Gungor et al. (2008) propose a real-time and reliable transport protocol to transport event features from sensors to actuators with minimum energy dissipation. However, sensor faults are not considered. Ngai et al. (2010) propose a latency-oriented fault tolerant transport protocol for WSANs. This method combine fault tolerant with data aggregation, but doesn't take sensor measurement noise into account. Nakayama et al. (2011) develop a mobility scheme to guarantee data gathering from all nodes fairly and efficiently in order to control all the nodes in WSANs. Although it provides a reliable data collection, subsequent data fusion and occasional sensor faults are not studied. Considering computation complexity, node resource, and system requirements, we present a Federated filter (Carlson, 1996) based mechanism to coordinate sensors and actuators. Federated filter technology is a flexible distributed filtering method and particularly suitable for information fusion in WSANs. As a decentralized method, the Federated filter requires only limited node resources for communication and computation, and its cascade structure makes it easy for data fusion.

Based on the information conveyed from sensors, organizing actuators effectively is the kernel of designing A-A coordination (Salarian et al., 2012). It deals with which actuators should be scheduled to execute a specific action and how to control their actions properly. During this procedure, multiple factors should be taken into account, such as, user requirements, actuators capabilities, resource constraints (Akkaya and Janapala, 2008), and quality-of-service (QoS) guarantee, such as real-timeliness (Xia,

⋆ The work was supported in part by NSFC under Grant 61222305 and 61203036, the SRFDP under Grant 20120101110139, NCET-11-0445 and National Program for Special Support of Top-Notch Young Professionals.

2008). Melodia et al. (2007) develop a localized actuation algorithm to minimize task completion time. In the overlapping areas, the algorithm selects actuators that can complete the task with minimum energy expenditure and with a given delay bound. This method provide an efficient way to control actuator in the overlapping areas but this is a centralized way. Cao et al. (2010) propose a centralized and a distributed control schemes in WSANs for building-environment control systems. Chen et al. (2010) develop a distributed estimation and collaborative control scheme for WSANs, which can achieve robust control against inaccurate system parameters. These schemes mainly focus on the optimization problem to reduce control error, while the energy consumption and action complete time are not take into account. Ota et al. (2012) study actuators' mobility control in WSANs for efficient events detecting in terms of time and energy consumption, but this method does not mention how to handle these events. In this paper, we propose an efficient A-A coordination mechanism to accurately and timely control the events while minimizing the energy consumption of each actuator.

The major contributions can be summarized as follows. We propose a collaborative estimation and actuation mechanism, which consists of a S-A coordination phase and an A-A coordination phase, for WSANs. Specifically, we formulate the S-A coordination as an estimation based data fusion problem and propose a federated filter based solution. We formulate the A-A coordination problem as a task allocation problem and propose a distributed solution to achieve user requirements and guarantee quality of service. Simulation results validate the effectiveness of the proposed mechanism. The remainder of this paper is organized as follows: Section 2 describes the system model and control requirements. Then the corresponding S-A and A-A coordination algorithms are designed in Section 3 and Section 4, respectively. Section 5 presents simulation results. Finally, Section 6 concludes this paper.

2. SYSTEM FORMULATION

We consider a WSAN for environment monitor and control applications. Normally we are interested in controlling the system states at several points of interest (POIs). We assume n sensors and m actuators are deployed in the region of interest (ROI) to detect and track p POIs and take necessary actions to deal with events occurring there. Let $X = [x_1, \ldots, x_p]'$ denote the environmental variables, where the subscripts are the indexes of the POIs. s_i denote the ith sensor, and a_j denote the jth actuator. We define that an event ex_i occurs if the state x_i varies from its set point x_i^*, where the set point x_i^* is the desired environment state at the ith POI and is prescribed by users. Our objective is to schedule and control the sensors and actuators to counteract with the events and stabilize the states X at their set points $X^* = [x_1^*, \ldots, x_p^*]'$.

We assume that the POIs are well-separated in the geographical area of ROI such that the corresponding states $\{x_1, \ldots, x_p\}$ are mutually uncorrelated. The value of X is influenced by a number of actuators deployed in ROI, for which we define $F = [f_1, \ldots, f_m]'$ as their outputs. Since the information exchanges over the control loop are carried out by discrete wireless packets, we model the dynamics of the states at the POIs with the following discrete-time state space model:

$$X(k+1) = AX(k) + BF(k) + \omega(k) \tag{1}$$

where $\omega(k)$ represents the process noise at step $k \in \{0, 1, 2, \ldots\}$. $A \in \mathbb{R}^{p\times p}$ is a diagonal matrix based on the aforementioned uncorrelation assumption. $B \in \mathbb{R}^{p\times m}$ is an input coefficient matrix whose element b_{ij} indicates the influence of actuator a_j exerted on system state x_i.

If x_i is within s_j's sensing range r_s, s_j can take a noisy measurement of x_i:

$$z_i^j(k) = c_j x_i(k) + \nu_j(k), j \in \{1, \ldots, n\}, i \in \{1, \ldots, p\} \tag{2}$$

where c_j is a coefficient, ν_j is the measurement noise of s_j. Assume that $\omega(k)$ and $\nu_j(k)$ are Gaussian, white, zero-mean with the following properties: $E\{\omega(k)\omega'(l)\} = Q(k)\delta_{kl}$, $E\{\nu_i(k)\nu_j'(l)\} = r_i(k)\delta_{kl}\delta_{ij}$, where $\delta_{kl} = 1$ if $k = l$, and $\delta_{kl} = 0$, otherwise. Here, we consider a scenario that each sensor only convey one system state; while a system state can be sensed by multiple sensors.

Let u_j denote the control input to adjust a_j's actuation. At every step k, a_j can be modeled as:

$$f_j(k) = \alpha f_j(k-1) + \beta u_j^\gamma(k), j = 1, \ldots, m \tag{3}$$

where α, β, γ are the constants depending on the type of actuator. For ease of exposition, we assume homogeneous actuator in the sequel. From (2) and (3), we can see that each sensor is a time-driven device, since data collection is controlled by the sample time; while each actuator is an event-driven device depending on the control techniques used. Hereby, we assume network-wide clock synchronization is achieved.

Since the actuators' influence ranges are generally limited, all the actuators can be partitioned into a number of independent sets such that any two actuators from different sets do not influence the state at a common POI. Mathematically, the system partitioned can be described by the following matrix rearrangements:

$$A = \begin{bmatrix} A_1 & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & A_N \end{bmatrix}, B = \begin{bmatrix} B_1 & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & B_N \end{bmatrix} \tag{4}$$

where $A_i \in \mathbb{R}^{p_i\times p_i}$, $B_i \in \mathbb{R}^{p_i\times m_i}$, $i \in \{1, \ldots, N\}$, and $\sum_{i=1}^N p_i = p$, $\sum_{i=1}^N m_i = m$. Then, the whole system can be divided into N separated subsystems $\{GS_1, \ldots, GS_N\}$. Due to the inter-dependence among the subsystems, we can thus focus on a single subsystem in the following design. Without loss of generality, we assume the parameters of the dth subsystem GS_d is:

$$A_d = \begin{bmatrix} a_{rr} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & a_{tt} \end{bmatrix}, B_d = \begin{bmatrix} b_{rl} & \cdots & b_{rq} \\ \vdots & \ddots & \vdots \\ b_{tl} & \cdots & b_{tq} \end{bmatrix} \tag{5}$$

where $1 \leq r < t \leq p$ and $1 \leq l < q \leq m$.

Our collaborative estimation and actuation scheme consists of two phases. In the S-A coordination phase, once an event ex_i takes place, sensors that covering it are organized to sense the event and collaborate with actuators to estimate x_i. During the A-A coordination phase, the actuators collaboratively decide which actuators will execute specific

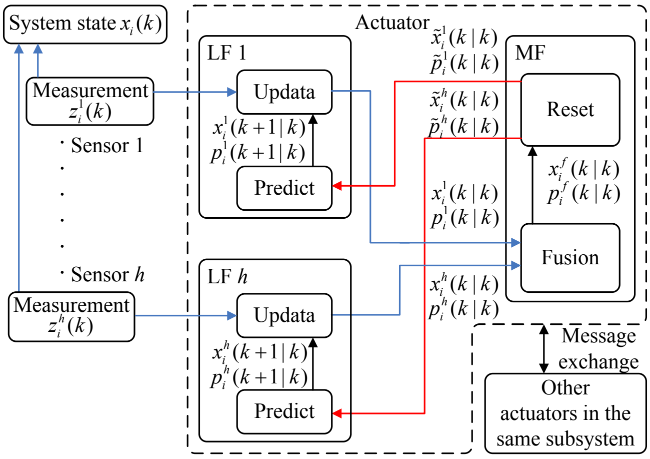

Fig. 1. The structure of FKF design for S-A coordination.

actions in response to the event based on the estimation results in S-A coordination phase. The decision also takes response time and energy consumption into consideration.

3. S-A COORDINATION

FKF is a distributed filter consisting of local filters (LFs) and a master filter (MF) as shown in Fig.1. The outputs of LFs are judged and combined by a MF to provide reliable and precise estimation. FKF has been proven to be an effective way for multi-sensor data fusion applications.

Then, we apply FKF to coordinate sensors and actuators, where LFs and MF are performed by the actuator nearest POI.

3.1 Local Filter Design

We assume sensor set $V_i^s = \{s_j, j = 1, \ldots, h\}$ is responsible for $x_i \in X_d$, i.e., x_i is within the sensing range of $s_j, j = 1, \ldots, h$. Then sensors in V_i^s perform sensing and periodical send their measurements to the nearest actuator to achieve estimation and fusion. This is because the closer the actuator to the event is, the shorter time is required to transmit the packets, and thus the quicker the actuator reacts on the event. When h sensors are scheduled, then accordingly h LFs are needed to be implemented in parallel.

Consider actuators' outputs will influence sensor measurements, in order to estimate system state more precisely, at step k, based on node deployment, s_j's measurement can be calibrated as follow:

$$\tilde{z}_i^j(k) = z_i^j(k) - \sum_{t=1}^{m} \tilde{b}_{jt} f_t(k-1) \tag{6}$$

where $\tilde{b}_{jt}$ denotes the influence of a_t exerts on s_j. Then, according to Carlson (1996), the measurement update can be expressed as:

$$(p_i^j(k|k))^{-1} = (p_i^j(k|k-1))^{-1} + c_j r_j^{-1}(k) c_j \tag{7}$$

$$\begin{aligned}&(p_i^j(k|k))^{-1} x_i^j(k|k)\\ &\quad = (p_i^j(k|k-1))^{-1} x_i^j(k|k-1) + c_j r_j^{-1}(k) \tilde{z}_i^j(k)\end{aligned} \tag{8}$$

After that, LF will transmit $\{x_i^j(k|k), p_i^j(k|k)\}$ to the fusion center MF.

3.2 Master Filter Design

Once MF collects the local estimations from LFs, the global estimation and the associated error covariance are given by:

$$p_i^f(k|k) = [(p_i^1(k|k))^{-1} + \ldots + (p_i^h(k|k))^{-1}]^{-1} \tag{9}$$

$$\begin{aligned}x_i^f(k|k) = p_i^f(k|k)[(p_i^1(k|k))^{-1} x_i^1(k|k) + \ldots\\ \ldots + (p_i^h(k|k))^{-1} x_i^h(k|k)]\end{aligned} \tag{10}$$

where $p_i^f(k|k)$ is called the information matrix. (10) shows that the fused result is a linear weighted combination of each LF's estimation result.

However, the estimations of different LFs are correlated since $\{x_i^1(k|k), \ldots, x_i^h(k|k)\}$ come from the same x_i. In order to eliminate this correlation, according to Edelmayer and Miranda (2007), the fused results are reset as follows:

$$\begin{cases}\tilde{x}_i^j(k|k) = x_i^f(k)\\ \tilde{p}_i^j(k|k) = (\beta_i^j(k))^{-1} p_i^f(k)\\ \tilde{q}_i^j(k) = (\beta_i^j(k))^{-1} q_{ii}(k)\end{cases} \tag{11}$$

where $q_{ii}(k)$ is the (i,i)th element of $Q(k)$, and $\beta_i^j(k)$ is called the information-sharing factor and must satisfy the following constraints:

$$\begin{cases}\displaystyle\sum_{j=1}^{h} \beta_i^j(k) = 1\\ 0 \le \beta_i^j(k) \le 1\end{cases} \tag{12}$$

Since the trace of $p_i^j(k|k)$ represents each LF's estimation accuracy, then we choose $\beta_i^j(k)$ based on the following equation:

$$\beta_i^j(k) = \frac{tr^{-1}(p_i^j(k|k))}{\displaystyle\sum_{j=1}^{h} tr^{-1}(p_i^j(k|k))} \tag{13}$$

In order to organize actuators in subsystem GS_d effectively, we group these actuators into a cluster and randomly or sequentially choose one actuator to serve as cluster head. After receiving the fusion results transmitted from its numbers, the cluster head accesses GS_d's system states $X_d^f(k|k) = [x_r^f(k|k), \ldots, x_t^f(k|k)]'$ and performs the following task allocation mechanism: (1) calculates actuator control law $U_d(k) = [u_l(k), \ldots, u_q(k)]'$ based on a SUMT algorithm (details will be presented in Section 4.2), and (2) sends commands $\{u_l(k), \ldots, u_q(k)\}$ to the corresponding actuators $\{a_l, \ldots, a_q\}$. Then, the state and the error covariance estimated by LF in the next step are:

$$x_i^j(k+1|k) = a_{ii} \tilde{x}_i^j(k|k) + \sum_{s=l}^{q} b_{is} f_s(k) \tag{14}$$

$$p_i^j(k+1|k) = a_{ii} \tilde{p}_i^j(k|k) a_{ii} + \tilde{q}_i^j(k) \tag{15}$$

From the above statement, we can see that the proposed S-A coordination decomposes the estimation process into several sub-processes and allocates proper tasks to each node. Note that (1) control inputs are not known at the sensors, and (2) the actuators are more powerful than the sensors. In order to reduce the messages exchange among the nodes and make an optimum usage of node's resource, the major tasks such as estimation, fusion and fault detection are performed by the resource-rich actuators rather than the resource-constrained sensors.

4. A-A COORDINATION

In the A-A coordination phase, we aim to find, for each occurring event, the optimal control law meets user requirements while minimizing the energy required to complete the action associated with the occurring event, under the constraint of meeting the time bound required by the application.

4.1 Actuator Task Allocation

Let $t_i^j(k)$ denote time required for a_j to act alone and independently to deal with ex_i (if ex_i is within a_j's action range r_a). The value of $t_i^j(k)$ is related to (1) the power that a_j uses to perform the action, (2) the distance between a_j and ex_i, and (3) the magnitude of ex_i. Without loss of generality, we assume:

$$t_i^j(k) = \mathcal{H}(P_j(k), d_{ij}, e_i(k)) \tag{16}$$

where $\mathcal{H}(\cdot)$ can be defined according to the particular application, $P_j(k) = k_p f_j(k)$ is the power that a_j uses to perform the action, k_p is a power transfer constant coefficient, d_{ij} is the Euclidean distance between ex_i and a_j, and $e_i(k) = x_i^f(k) - x_i^*$.

When event ex_i occurs, at least one actuator that covers x_i will be scheduled to perform the action. Since some off-diagonal elements of B_d are not equal to zero, the action of this actuator may change the other system states. Therefore, we consider all the actuators in the scheduling problem. In order to facilitate task allocation, based on matrix B_d, we define a binary matrix B'_d whose element b'_{ij} satisfies the following equation:

$$b'_{ij} = \begin{cases} 1, \, b_{ij} \neq 0 \\ 0, \, b_{ij} = 0 \end{cases} \tag{17}$$

If $b_{ij} \neq 0, j \in \{l, \ldots, q\}$, system state x_i is influenced by actuators $a_j, j \in \{l, \ldots, q\}$, then the time required to deal with ex_i is:

$$T_i(k) = \mathcal{G}(b'_{il}, \ldots, b'_{iq}, t_i^l(k), \ldots, t_i^q(k)) \tag{18}$$

where $\mathcal{G}(\cdot)$ can be defined according to the particular application.

If $b_{ij} \neq 0, i \in \{r, \ldots, t\}$, actuator a_j is responsible for system states $x_i, i \in \{r, \ldots, t\}$, then the energy required for a_j to deal with events occur within its action range is:

$$E_j(k) = P_j(k)\Delta k \tag{19}$$

where Δk is system's sampling period.

Consider the objective is to find an optimal set of control law U to adjust actuators so as to meet the desired states at the POIs, e.g., $X = X^*$, In GS_d, the actuator task allocation problem can be defined as follows:

$$\min \quad J(k) = \sum_{j=l}^{q} E_j(k) \tag{20}$$

$$\text{s.t.} \quad \begin{cases} X_d(k+1|k) = X_d^* & \text{(21a)} \\ E_j(k) \leq E_j^{res}(k), \forall j & \text{(21b)} \\ T_i(k) + T_d(k) \leq T_{th}, \forall i & \text{(21c)} \\ \underline{U} \leq U_d(k) \leq \overline{U} & \text{(21d)} \end{cases}$$

(20) minimizes the energy consumption of entire actuators. (21a) imposes that all system states should satisfy the desired control requirements, where $X_d(k+1|k) = A_d X_d^f(k|k) + B_d F_d(k)$. (21b) defines actuator has adequate energy to complete the task, $E_j^{res}(k)$ is a_j's available energy at step k. (21c) limits the action completion time is smaller than the threshold T_{th} ($T_{th} < \Delta k$), discount by the decision time $T_d(k)$, i.e., the communication and the computing times. (21d) bounds the control signal. The actuator task allocation problem in other subsystems can be handled in the same way.

4.2 Control Algorithm

From the above statement, it is noted that in order to solve the actuator task allocation problem, the states of the whole sub-system should be accessed at the controller. As we mentioned before, after data fusion complete, actuator will relay its fusion result to the cluster head, then, a simple method is to assign this cluster cluster responsible for task allocation. In the following, we will explain how to search the desired solution.

To deal with the nonlinear optimal problem which inequality constraints and equality constraints, the most commonly used method is the sequential unconstrained minimization technique (SUMT) (Dussault, 2011). The basic idea of SUMT is to transfer (20) and (21) into the following equation:

$$\phi(U_d, M^{(p)}) = \sum_{j=l}^{q} E_j(k) + M^{(p)} \sum_i \sum_j \{\max[g_i^j(U_d), 0]\}^2$$
$$+ M^{(p)} \sum_m \sum_n [h_m^n(U_d)]^2 = \sum_{j=l}^{q} E_j(k) + M_1 + M_2 \tag{22}$$

where

$$\begin{cases} g_1^j(U_d) = E_j(k) - E_j^{res}(k) \leq 0, j = l, \ldots, q \\ g_2^i(U_d) = T_i(k) + T_d(k) - T_{th} \leq 0, i = r, \ldots, t \\ g_3^j(U_d) = u_j(k) - \overline{u} \leq 0, j = l, \ldots, q \\ g_4^j(U_d) = \underline{u} - u_j(k) \leq 0, j = l, \ldots, q \\ h_1^i(U_d) = x_i(k+1|k) - x_i^* = 0, i = r, \ldots, t \end{cases}$$

and search the optimal U_d to minimize $\phi(U_d, M^{(p)})$. i, j, m, n are the number of constraints, $M^{(p)}, p = 1, 2, \ldots$ is a sequence tending to infinity such that $M^{(p)} > 0$ and $M^{(p+1)} > M^{(p)}$, p is the computation period of SUMT and p is much smaller than the system sample period k, M_1 and M_2 are the penalty functions related to inequality and equality constraints. In M_1,

$$\max[g_i^j(U_d), 0] = \frac{g_i^j(U_d) + |g_i^j(U_d)|}{2}$$
$$= \begin{cases} 0, & g_i^j(U_d) \leq 0, U_d \in \psi_f \\ g_i^j(U_d), & g_i^j(U_d) > 0, U_d \notin \psi_f \end{cases} \tag{23}$$

where ψ_f is the feasible region. If the search point of U_d is within ψ_f, $M_1 = 0, M_2 = 0$; otherwise, $M_1 \neq 0, M_2 \neq 0$, and the farther U_d deviates from ψ_f, the bigger penalty parameter $M^{(p)}$ is. Since our object is to minimize $\phi(U_d, M^{(p)})$, M_1 and M_2 must be forced to be zero. So, searching solutions in ψ_f to minimize (22) is equivalent to minimize (20) under (21). When minimizing

(22), problem solution U_d will change under different $M^{(p)}$, then $U_d(M^{(p)})$ can be considered as a trace with parameter $M^{(p)}$, if $0 < M^{(0)} < M^{(1)} < \cdots < M^{(p)} < M^{(p+1)} < \cdots \to \infty$, the point range $\{U_d(M^{(p)})\}$ will follow this trace to gradually converge to the optimal states of primal problem. The implementation details of SUMT is given below:

(1) Initialization step: Select a growth parameter $C > 1$, a stopping parameters $\varepsilon_s > 0$, an initial value of the penalty parameter $M^{(0)}$ and a starting point U_d^0, let $p = 1$.
(2) Iterative step: Using unconstrained search technique to find the point that minimizes $\phi(U_d, M^{(p-1)})$, call it $U_d(p)$ and determine which constraints are violated at this point.
(3) Stopping rule: If the distance between $U_d(p)$ and $U_d(p-1)$ is smaller than ε_s (i.e., $\| U_d(p) - U_d(p-1) \| \leq \varepsilon_s$) or the difference between two successive objective functions is smaller than ε_s (i.e., $|\sum_{j=l}^{q} E_j[U_d(p)] - \sum_{j=l}^{q} E_j[U_d(p-1)]| \leq \varepsilon_s$), stop with $U_d(p)$ as an estimation of the optimal solution. Otherwise, set $M^{(p)} = CM^{(p-1)}$, formulate the new $\phi(U_d, M^p)$ based on which constraints are violated at $U_d(p)$, let $p = p + 1$ and return to the Step (2).

4.3 Complexity Discussion

A key issue about node coordination mechanisms is the communication and computation complexity in terms of the number of message exchanges and calculations required by the algorithm. The analysis is as follows. For the S-A coordination, at each iteration, information exchange among sensors and actuators are the measurements $z_i^j(k)$ transmitted from sensors to their nearest actuator, where indexes i and j depend on the number of POIs in this subsystem and the number of sensors cover this POI, respectively. If h LFs run parallel, then the FKF costs $O(h)$ operations. For the A-A coordination, after receiving the sensing data relayed from its cluster numbers, the cluster head performs FKF and SUMT to calculate actuator control law and transmits the control commands to the corresponding actuators. For the cluster head, SUMT can be carried out with a complexity of $O(d^3)$ operations, where d represents the total number of constraints, and this method typically requires a few tens of iterations. Since $p \ll n$, $m \ll n$, at each iteration, each node only has to exchange a (small) constant amount of information with each other, and the implementations of FKF and SUMT are not complicated.

5. SIMULATION

We consider a WSANs based temperature control system as an illustrative example. The scenario consists of four actuators deployed in the ROI to control four POIs, i.e., $p = 4$, $m = 4$. The system model is:

$$A = \begin{bmatrix} 0.9 & 0 & 0 & 0 \\ 0 & 0.9 & 0 & 0 \\ 0 & 0 & 0.9 & 0 \\ 0 & 0 & 0 & 0.9 \end{bmatrix}, B = \begin{bmatrix} 0.3 & 0.4 & 0 & 0 \\ 0 & 0.5 & 0.3 & 0 \\ 0.4 & 0 & 0.3 & 0 \\ 0 & 0 & 0 & 0.6 \end{bmatrix}$$

According to the system partition, the entire system can be divided into the following two subsystems GS_1 and GS_2:

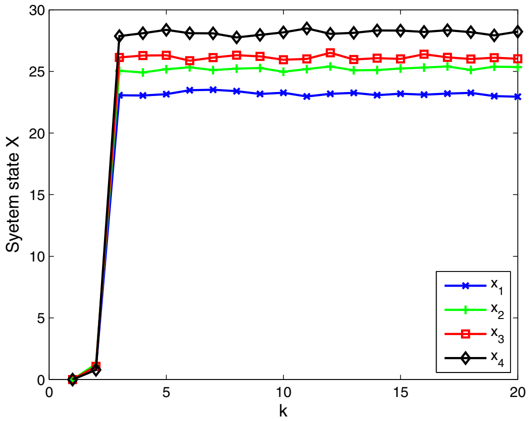

Fig. 2. Dynamic system response under the proposed node coordination.

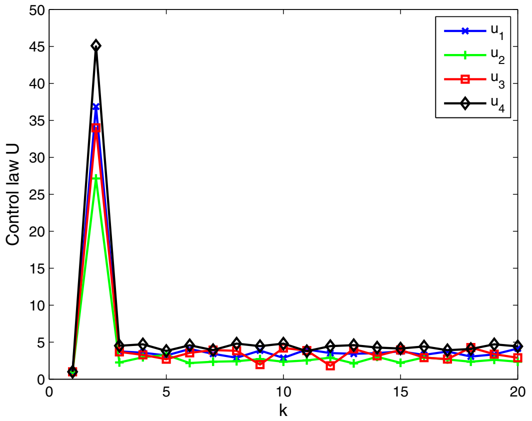

Fig. 3. The corresponding control law of actuators.

$$A_1 = \begin{bmatrix} 0.9 & 0 & 0 \\ 0 & 0.9 & 0 \\ 0 & 0 & 0.9 \end{bmatrix}, B_1 = \begin{bmatrix} 0.3 & 0.4 & 0 \\ 0 & 0.5 & 0.3 \\ 0.4 & 0 & 0.3 \end{bmatrix}$$

$$A_2 = [0.9], B_2 = [0.6]$$

After the initial nodes deployment, x_1, x_2, x_3, x_4 are covered by 3, 2, 2, 2 sensors. We assume the control signal of actuator, such as current or voltage, directly corresponds to the actuator's output, such as temperature. Then, we select $\alpha = 0$, $\beta = 1$, $\gamma = 1$, $k_p = 2$, $U \in [-50, 50]$, and the initial energy of actuator is $E_0 = 1000$. $\mathcal{H}(\cdot)$ and $\mathcal{G}(\cdot)$ are assumed to be $(k_t d_{ij} e_i(k))/(\eta_j p_j^\lambda(k))$ and $(\sum_{j=l}^{q} b'_{ij}/t_i^j(k))^{-1}$, where $k_t = 3$, $\eta_j = 2$, $\lambda = 1$. T_d, T_{th} and Δk are set to 2 s, 10 s and 20 s. The process noise $\omega(k)$ and measurement noise $\nu(k)$ have the amplitude: $q_{ii}(k) = 0.1$ and $r_i(k) = 0.5$. The initial system states are $X(0) = [0, 0, 0, 0]'(^\circ C)$, and our aim is to meet the set points $X^* = [23, 25, 26, 28]'(^\circ C)$.

Fig.2 illustrates the dynamic system response under the proposed S-A and A-A coordinations. The corresponding actuator control law is shown in Fig.3. In this context, we can see that the control system is bounded-input-bounded-output (BIBO) stable. As the introduction of error-based feedback control, system states will converge to their set points.

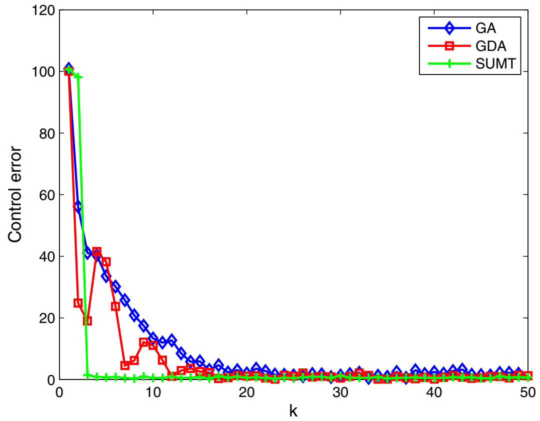

Fig. 4. System performance under different control schemes in the A-A coordination.

To deal with the optimal problem mentioned above, there are some common optimization schemes like: (1) genetic algorithm (GA), which is the most time consuming one: the larger population size is, the higher searching accuracy can be achieved. But the computational complexity will increase as well, which leads to a longer event processing delay. If the population size is selected inappropriately, the searching result will fall into local minimum or even diffuse; (2) gradient descent algorithm (GDA), which requires to calculate the partial derivative of objective function with respect to desired variables. However, if the variables are of high dimensions or subject to several constraints such as (21), the derivation is difficult to calculate and this method may be hard to apply. Fig.4 compares system performance under different optimization schemes in the A-A coordination. Although all of them are able to stabilize the control error close to 0, the SUMT has the fastest convergence speed, which can easily satisfy system's real-time requirements.

6. CONCLUSION

We have proposed a collaborative estimation and actuation scheme which effectively handles the S-A and A-A coordinations in WSANs. A multi-source data fusion algorithm based on FKF has been designed to coordinate sensors and actuators for event estimation. The A-A coordination has been formulated as an nonlinear optimization problem taking all of the user requirements, response time and energy consumption into account, and a SUMT based algorithm has been proposed to solve the optimization iteratively. The proposed scheme has been validated by simulations based on a temperature control system.

REFERENCES

Akkaya, K. and Janapala, S. (2008). Maximizing connected coverage via controlled actor relocation in wireless sensor and actor networks. *Computer Networks*, 52(14), 2779–2796.

Akyildiz, I.F. and Kasimoglu, I.H. (2004). Wireless sensor and actor networks: research challenges. *Ad Hoc Networks*, 2(4), 351–367.

Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications Magazine*, 40(8), 102–114.

Cao, X., Chen, J., Xiao, Y., and Sun, Y. (2010). Building-environment control with wireless sensor and actuator networks: centralized versus distributed. *IEEE Transactions on Industrial Electronics*, 57(11), 3596–3605.

Carlson, N.A. (1996). Federated filter for computer-efficient, near-optimal gps integration. In *IEEE Position Location and Navigation Symposium*, 306–314. IEEE.

Chen, J., Cao, X., Cheng, P., Xiao, Y., and Sun, Y. (2010). Distributed collaborative control for industrial automation with wireless sensor and actuator networks. *IEEE Transactions on Industrial Electronics*, 57(12), 4219–4230.

Dussault, J.P. (2011). Improved asymptotic analysis for sumt methods. In *Federated Conference on Computer Science and Information Systems*, 247–253. IEEE.

Edelmayer, A. and Miranda, M. (2007). Federated filtering for fault tolerant estimation and sensor redundancy management in coupled dynamics distributed systems. In *Mediterranean Conference on Control and Automation*, 1–6. IEEE.

Gungor, V.C., Akan, Ö.B., and Akyildiz, I.F. (2008). A real-time and reliable transport $(rt)^2$ protocol for wireless sensor and actor networks. *IEEE/ACM Transactions on Networking*, 16(2), 359–370.

Melodia, T., Pompili, D., Gungor, V.C., and Akyildiz, I.F. (2007). Communication and coordination in wireless sensor and actor networks. *IEEE Transactions on Mobile Computing*, 6(10), 1116–1129.

Nakayama, H., Fadlullah, Z.M., Ansari, N., and Kato, N. (2011). A novel scheme for wsan sink mobility based on clustering and set packing techniques. *IEEE Transactions on Automatic Control*, 56(10), 2381–2389.

Ngai, E., Zhou, Y., Lyu, M.R., and Liu, J. (2010). A delay-aware reliable event reporting framework for wireless sensor–actuator networks. *Ad Hoc Networks*, 8(7), 694–707.

Ota, K., Dong, M., Cheng, Z., Wang, J., Li, X., and Shen, X.S. (2012). Oracle: Mobility control in wireless sensor and actor networks. *Computer Communications*, 35(9), 1029–1037.

Ruiz-Ibarra, E. and Villasenor-Gonzalez, L. (2008). Cooperation mechanism taxonomy for wireless sensor and actor networks. In *Wireless Sensor and Actor Networks II*, 62–73. Springer.

Salarian, H., Chin, K.W., and Naghdy, F. (2012). Coordination in wireless sensor–actuator networks: A survey. *Journal of Parallel and Distributed Computing*, 72(7), 856–867.

Stankovic, J.A. (2008). When sensor and actuator networks cover the world. *ETRI Journal*, 30(5), 627–633.

Wu, Z.D. (2011). Modelling and analysis of strategies in the design of wsan coordination systems. *Journal of Parallel and Distributed Computing*, 71(7), 1055–1064.

Xia, F. (2008). Qos challenges and opportunities in wireless sensor/actuator networks. *Sensors*, 8(2), 1099–1110.

Yuan, H., Ma, H., and Liao, H. (2006). Coordination mechanism in wireless sensor and actor networks. In *The 1st International Multi-Symposiums on Computer and Computational Sciences*, 627–634. IEEE.