Distributed Node Coordination for Real-Time Energy-Constrained Control in Wireless Sensor and Actuator Networks

Lei Mo, Xianghui Cao, Ye-Qiong Song, Angeliki Kritikakou

To cite this version:

Lei Mo, Xianghui Cao, Ye-Qiong Song, Angeliki Kritikakou. Distributed Node Coordination for Real-Time Energy-Constrained Control in Wireless Sensor and Actuator Networks. IEEE Internet of Things Journal, 2018, pp.1-12. 10.1109/JIOT.2018.2839030. hal-01825524

HAL Id: hal-01825524
https://inria.hal.science/hal-01825524
Submitted on 28 Jun 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Distributed Node Coordination for Real-Time Energy-Constrained Control in Wireless Sensor and Actuator Networks

Lei Mo*, Member, IEEE, Xianghui Cao†, Senior Member, IEEE, Yeqiong Song‡, Member, IEEE, and Angeliki Kritikakou*, Member, IEEE

Abstract—Wireless Sensor and Actuator Networks (WSANs) are emerging as a new generation of Wireless Sensor Networks (WSNs). Due to the coupling between the sensing areas of the sensors and the action areas of the actuators, the efficient coordination among the nodes is a great challenge. In this paper, we address the problem of distributed node coordination in WSANs aiming at meeting the user’s requirements on the states of the Points of Interest (POIs) in a real-time and energy-efficient manner. The node coordination problem is formulated as a non-linear program. To solve it efficiently, the problem is divided into two correlated subproblems: the Sensor-Actuator (S-A) coordination and the Actuator-Actuator (A-A) coordination. In the S-A coordination, a distributed federated Kalman filter-based estimation approach is applied for the actuators to collaborate with their ambient sensors to estimate the states of the POIs. In the A-A coordination, a distributed Lagrange-based control method is designed for the actuators to optimally adjust their outputs, based on the estimated results from the S-A coordination. The convergence of the proposed method is proved rigorously. As the proposed node coordination scheme is distributed, we find the optimal solution while avoiding high computational complexity. The simulation results also show that the proposed distributed approach is an efficient and practically applicable method with reasonable complexity.

Keywords—Wireless sensor and actuator networks, distributed coordination, energy, delay, convergence analysis.

I. INTRODUCTION

WITHIN the ten last years we have observed the remarkable growth of Wireless Sensor Networks (WSNs) in a variety of areas, such as environment monitoring, target tracking, object and event detection [2], [3]. The WSNs can be viewed as open-loop systems with low-cost wireless sensors to monitor their surrounding physical world. WSNs extended with wireless communication capable actuators become Wireless Sensor and Actuator Networks (WSANs). They can not only measure specific physical variables (e.g., temperature, illumination intensity, voltage), but also change them through only measure specific physical variables (e.g., temperature, illumination intensity, voltage), but also change them through

environment control, building automation, industrial control and smart grid [6]–[13].

An important challenge in the design of such systems is that the sensing areas of the sensors and the action areas of the actuators are coupled, which introduces a significant difficulty in the efficient coordination of the sensors and the actuators [6], [7], [10]. A typical WSAN example is the industrial environment control [7], [10], where a number of wireless thermometers (sensors) and heaters/coolers (actuators) are deployed in a Region of Interest (ROI) to monitor and control the states (temperatures) of the Points of Interest (POIs). As the example shows in Fig. 1, the aim is to control the states of POIs $P_1$, $P_2$ and $P_3$ to meet the user’s requirements (e.g., 23 °C). Therefore, the actuators $A_1$ and $A_2$ need to coordinate with each other using the information received from the sensors $S_1$, $S_2$ and $S_3$. Since the actuator $A_3$ and the POI $P_3$ are uncorrelated to the other actuators and POIs, the actuator $A_3$ is able to make control decision individually, using the information from the sensors $S_4$ and $S_5$. Hence, all the sensors and the actuators, whose sensing and action ranges that cover the same POIs, should coordinate with each other to decide for the appropriate actions.

A centralized scheme [14], [15] addresses the node coordination problem by gathering the information from all sensors and actuators and making the optimal state estimation and the optimal actuator control decision. However, the communication and the computation overhead is very high, especially when the network size is large. In addition, the risk of single node failure exists (e.g., at the sink or at the node near to the sink). As an alternative, a distributed node coordination is able to reduce the computational complexity and improve the system’s scalability [7]. Distributed approaches are required at each phase of the WSAN coordination. Through a distributed Sensor-Actuator (S-A) coordination, the actuators collaborate with their ambient sensors to estimate the states of POIs (e.g., in Fig. 1, the actuators $A_1$ and $A_2$ collaborate with the sensors $S_1$, $S_2$ and $S_3$ to estimate the states of POIs $P_1$ and $P_2$). A distributed Actuator-Actuator (A-A) coordination allows the correlated actuators to coordinate with each other in order to adjust their outputs and control the states of POIs to meet the user’s requirements (e.g., in Fig. 1, the actuators $A_1$ and $A_2$ coordinate with each other to control the states of POIs $P_1$ and $P_2$). Note that a closed-loop control is formed by the distributed S-A and A-A coordinations, which implies that they should be solved concurrently.

In WSANs, the number of sensors is usually much larger than the number of actuators. An actuator may receive information from multiple sensors [16]–[18]. Hence, a key issue for the distributed S-A coordination is to fuse the sensors information at the actuators, in order to estimate the real environment states.
Although heuristics can find feasible solutions in a short amount of time, they do not provide bounds on the solution quality. In addition, they are not always easily extensible and when new assumptions or constraints must be redone. Moreover, the methods in [14], [22], [25] are hard to be applied for a distributed control.

For WSANs deployed in harsh and critical environments, multiple requirements are implied (e.g., control accuracy, energy efficiency and real-time performance). Hence, the S-A and A-A coordination joint-design problem under such constraints is essential for WSANs. However, this joint consideration is rare in the existing literature, especially for the distributed solution. Hence, this paper answers the following questions: 1) How to find a proper way to formulate the S-A and A-A coordination joint-design problem so as to minimize the control error subject to energy supply and real-time constraints? and 2) Is there a way to achieve an optimal solution while avoiding high computational complexity? This paper is initially based on our preliminary results in [1] and significantly extends this work by providing the following contributions:

1) We formulate the WSAN node coordination problem as a non-linear problem, taking control accuracy, energy efficiency and real-time performance into account. The proposed system model covers the general case, compared with the simplified model of [1] which considers only diagonal matrices. The problem is proved to be convex and it is divided into two correlated S-A and A-A coordination subproblems, which can be solved sequentially.

2) Due to the general system model, the system states are now correlated with each other. Therefore, the S-A coordination scheme has to be modified in order to deal with the coupled system states compared to the approach presented in [1], where no state correlation is possible. A distributed S-A coordination solution is proposed using a Voronoi cells method to group the sensors and the actuators into several clusters. In each cluster, one actuator is selected as fusion center (cluster head), which is responsible for collaborating with its ambient sensors (cluster members) to perform state estimation. The received information from the sensors contains measurement noises, and, thus, the cluster head performs a distributed Federated Kalman Filter (FKF)-based method to optimally estimate the system states.

3) We propose a distributed Lagrange-based algorithm to solve the A-A coordination problem. This method coordinates the actuators that cover correlated POIs. Each actuator individually makes control decision, based on the limited exchange of information between the actuators. Moreover, the algorithm convergence is analyzed theoretically. The A-A coordination scheme presented in [1] is centralized, which prohibits its application in large size networks.

4) The theoretical analysis and the simulations show that the proposed method achieves the desirable control accuracy and real-time performance and energy efficiency. In this paper, we conduct extensive simulations to validate the robustness and the scalability of the proposed method under in non-idea environments (i.e., with faulty sensor measurements and actuator failures), compared with the simulations in [1] which only evaluate the system control performance. Moreover, the proposed method achieves an optimal state estimation and an optimal control scheme with low computation overhead.
due to the distributed architecture.

The rest of this paper is organized as follows. Section II reviews the literature. Section III presents the system model and formulates the problem. Section IV and Section V describe the distributed S-A and A-A coordination schemes, respectively. Finally, Section VI shows the simulation results and Section VII concludes this paper.

II. RELATED WORK

For centralized S-A coordination, the work in [16] proposes a fault-tolerant data routing and fusion method with minimum energy consumption under a delay bound. In [26], an event data collection approach is presented to balance network lifetime and event detection reliability. To avoid obstacles in the sensing field and collect data from the sensors, an energy-efficient routing mechanism is designed in [27] for the mobile data collectors. However, the state estimation is not considered in these works. When sensor readings are noisy, the quality of fusion result degrades. A Maximum Likelihood Estimation (MLE)-based method is proposed in [8] to locate the fire source, estimate the temperature and evaluate its emergency. An enhanced Bayesian inference based method is designed in [18] to fuse the sensory data and estimate the system states.

For distributed state estimation, some methods have been successfully applied to the WSNs, e.g., [19], [20]. Although these works also utilize Kalman filter to perform state estimation, they still differ from our method in the following ways. The estimation method in [19] focuses on smoothing the system states by a diffusion Kalman filter (i.e., estimating the states of discrete system at non-sampling steps). The scenario is different from ours since we consider the state estimation at sampling steps. In [20] all the sensors apply a consensus method to obtain the same state estimation in steady-state. Our method is not based on consensus, since in our scenario different sensors may be responsible for different system states. Moreover, the above methods are not suitable for WSNs since the closed-loop control provided by the actuators is not taken into account. In WSNs, although distributed data routing and fusion methods can be found in [17], [21], the state estimation is not considered in these works.

For centralized A-A coordination, a simplex based two-stage optimization method is proposed in [6] and an LQG based control scheme [11] is applied to smart grid to implement a distributed voltage control. A heuristic based control scheme is presented in [24] to guarantee the desired levels of the control performance and reduce the energy consumption. However, the problems of control accuracy, energy-efficiency and real-timeliness are not considered simultaneously in above works.

III. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we first present the WSAN model and then formulate the node coordination problem. A summary of the notations that are used in this paper is given as follows. For a matrix $M$, $m_{ij}$ is the $(i,j)$th element of $M$, and $tr(M)$ is the trace of $M$. For a vector $v$, $v_i$ is the $i$th element of $v$, and $D(v)$ is the dimension of $v$. For a scalar $q$, $[a]_q^+$ = $\max\{0,q\}$, and $[a]_q^- = \max\{q,\min\{q,0\}\}$. $\mathbb{E}(\cdot), (\cdot)'$, and $\|\cdot\|_2$ are the operators for the expectation of a random variable, the transpose of a matrix/vector, and the 2-norm of a vector, respectively. $y \sim \mathcal{N}(\mu, \sigma)$ represents the random variables following a Gaussian distribution with mean $\mu$ and covariance $\sigma$, where $\mu$ and $\sigma$ have the appropriate dimensions. Let $p = [p_1, \ldots, p_m]'$ and $q = [q_1, \ldots, q_m]'$. $p \preceq q$ represents $p_i \leq q_i$ $(1 \leq i \leq m)$.

A. System Model

We consider $n_s$ static sensors $\{S_1, \ldots, S_{n_s}\}$ and $n_a$ static actuators $\{A_1, \ldots, A_{n_a}\}$ randomly deployed in an ROI to monitor the states of $n_p$ POIs and take necessary actions to deal with the events occurring in these POIs. The system states $x = [x_1, \ldots, x_{n_p}]'$ represent the physical variables under consideration (e.g., temperature, illumination intensity, voltage). An event is occurred when a system state exceeds a predefined threshold. The sensors measure the system states periodically and transmit their readings to the ambient actuators. Based on the collected information, the actuators coordinate with each other to make proper decisions and perform the corresponding actions to control the system states to meet the user’s requirements $x^*$.

Note that 1) a system state $x_i$ is affected by both the nearby actuators and the correlated system states, 2) the actuators are static, and 3) the information exchanging between the sensors and the actuators is carried out by discrete wireless packets [7], [10]. We consider a system which is modeled by the following discrete Linear Time-Invariant (LTI) process:

$$x(k+1) = Ax(k) + Bu(k) + \omega(k),$$

where $u(k) \in R^{n_a \times 1}$ and $\omega(k) \sim \mathcal{N}(0, Q)$ are the vectors of actuators’ outputs and system noise, respectively. $A \in R^{n_p \times n_p}$ and $B \in R^{n_p \times n_a}$ are the system matrix and the input matrix, respectively. $\Delta$ is the system sampling period, and $\mathbf{u} \preceq u(k) \preceq \mathbf{u}$ since the actuators’ outputs are bounded.

For the sensor $S_j$, if the system states $\{x_p, \ldots, x_q\}$ are within its sensing range $R_s$, the sensor $S_j$ takes a noisy measurement of them:

$$z_j(k) = [c_{ij}^p x_p(k) + \nu_j^p(k), \ldots, c_{ij}^q x_q(k) + \nu_j^q(k)]',$$

where $c_{ij}^p (p \leq i \leq q)$ is a coefficient, $\nu_j^p$ is the measurement noise, and $\nu(k) \sim \mathcal{N}(0, R)$. We assume that the system in [32] to minimize the workloads (energy consumption) of the actuators, while keeping them approximately balanced. To jointly optimize the accuracy-delay, an LMI-based control scheme [11] is applied to smart grid to implement a distributed voltage control. A heuristic based control scheme is presented in [24] to guarantee the desired levels of the control performance and reduce the energy consumption. However, the problems of control accuracy, energy-efficiency and real-timeliness are not considered simultaneously in above works.
described by (1) and (2) is controllable and observable and it is synchronized by one of the existing clock synchronization methods [33].

**B. Problem Formulation**

Our objective is to find, for each occurring event, the strategies of the sensors and the actuators to drive the system states \( x \) towards the user’s requirements \( x^* \) subject to a set of real-time and energy supply constraints. The whole problem is divided into two correlated subproblems of the S-A and the A-A coordinations described as follows.

Denote \( x(k) = [x_1(k), \ldots, x_n(k)]^T \) as the collected sensor measurements of the entire network at the \( k \)-th step. The goal of the S-A coordination is to determine a posteriori estimate \( \hat{x}(k|k) \) and a priori estimate \( \bar{x}(k|k-1) \) of system states \( x(k) \) [34] as defined below:

\[
\begin{align*}
\hat{x}(k|k) &= \mathbb{E}(x(k)|z(k)), \\
\bar{x}(k|k-1) &= \mathbb{E}(x(k)|z(k-1)).
\end{align*}
\]

**Definition 3.1 (Event):** Denote \( e_i \) as a small positive tolerance. If the state estimation \( x_i(k) \) exceeds a predefined range \([x_i^-, e_i, x_i^++e_i]\), while the state estimation \( x_i(k-1) \) was within this range, an event in the system state \( x_i \) is occurred at the \( k \)-th step.

**Definition 3.2 (Actuation delay):** For an event in the system state \( x_i \), the actuation delay is the number of sampling periods that the state estimation \( x_i(k) \) remains out of the range \([x_i^-, e_i, x_i^++e_i]\) after the occurrence of this event.

Assume that an event occurs in system state \( x_i \) at the \( k \)-th step. Let \( T_i(k) \) be the actuation delay, \( A_i = \{A_j \mid b_{ij} \neq 0, 1 \leq j \leq n_a\} \) be the set of responsible actuators of system state \( x_i \), and \( U_i(k) = \{u_j(k) \mid b_{ij} \neq 0, 1 \leq j \leq n_u\} \) be the set of corresponding actuators’ outputs. Note that \( 1) \ T_i(k) \) mainly depends on the control error \( e_i(k) = x^* - x_i(k) \) and the actuators’ outputs \( U_i(k) \), and \( 2) \) under the given control error \( e_i(k) \), if the actuator’s output \( u_j(k) \) increases, the actuation delay \( T_i(k) \) reduces. Hence, we have

\[
T_i(k) = \mathcal{G}(e_i(k), U_i(k)),
\]

where \( \mathcal{G} \) is a differentiable and decreasing function with respect to \( U_i \). We assume that the elements in the set \( U_i \) affect the value of \( T_i \) independently (i.e., \( \frac{\partial^2}{\partial u_k \partial u_q} \mathcal{G} = 0 \) \( p \neq q \)).

To drive the system states \( x \) towards the user’s requirements \( x^* \), the actuators’ outputs \( u \) can be either positive or negative. Denote \( p_j(k) = k_p u_j(k)^2 \) as the actuator \( A_j \)’s power at the \( k \)-th step, where \( k_p \) is a positive constant (i.e., the larger actuator output is, the more energy is consumed). Since each actuator maintains its output step-wise, the consumed energy of the actuator \( A_j \) at the \( k \)-th step is given by \( E_j(k) = \Delta_p p_j(k) \).

Our objective is to minimize the control error \( \|Ax(k|k) + Bu(k) - x^*\|^2 \) and the energy consumption of the actuators \( \sum_{j=1}^{n_a} \Delta k_p u_j(k)^2 \) by adjusting the actuators’ outputs \( u \).

Summarizing the objective and all the constraints mentioned above, the Primal Problem (PP) is formulated as

\[
\begin{align*}
\text{min}_{u(k)} & \quad J(k) \\
\text{s.t.} & \quad \Delta k_p u_j(k)^2 \leq E_j(k) \leq E_j', \quad 1 \leq j \leq n_a, \\
& \quad G(e_j(k), U_i(k)) + T_a \leq T_i, \quad 1 \leq i \leq n_p, \\
& \quad u_j \leq u_j(k) \leq u_j^*, \quad 1 \leq j \leq n_a.
\end{align*}
\]

where

\[
J(k) = \|Ax(k|k) + Bu(k) - x^*\|^2 + \alpha \sum_{j=1}^{n_a} \Delta k_p u_j(k)^2,
\]

and \( \alpha \) is a positive coefficient. \( \Delta k_p u_j(k)^2 \leq E_j(k) \) indicates that the energy consumption of each actuator cannot exceed its residual energy \( E_j' \). \( G(e_j, U_i(k)) + T_a \leq T_i \) implies that the actuation delay should not exceed a time threshold \( T_i \), minus by the communication and computing time \( T_a \). \( T_i \) is determined by the applications and \( \alpha \geq \Delta \).

**Theorem 3.1:** The PP (3) is a convex optimization problem.

**Proof:** See Appendix A for the proof.

The PP requires an appropriate A-A coordination to solve the actuator output adjustment problem based on the state estimation \( x(k|k) \), which is obtained by the S-A coordination. In the following section, we will show the details of S-A coordination and A-A coordination.

**IV. S-A COORDINATION AND DISTRIBUTED ESTIMATION**

In this section, a distributed fusion method is applied among the actuators and their ambient sensors to estimate the system states, based on the FKF [35]. As shown in Fig. 2, this method employs a two-stage data processing structure, where the Local Filters (LFs) (i.e., \{LF_1, \ldots, LF_{n_s}\}) perform the local state estimation using the measurements received from the sensors. Then, the estimated results from the LFs are fused in a Master Filter (MF) to yield an optimal state estimation.

**A. Preliminaries**

Denote \( r \) as the impact range of the system state \( x_i \), and \( r_a \) as the action range of the actuator \( A_j \). Usually, the values of \( r \) and \( r_a \) are limited and much smaller than the range of ROI. Hence, if the POIs and the actuators are randomly deployed in the ROI rather than concentrated on a small area, the control of the system states \( x_p \) and \( x_q \) \( p \neq q \) may be decoupled with each other (e.g., in Fig. 1, the control of system states \( x_1, x_2 \) and \( x_3 \) are decoupled). Let \( \mathcal{X} = \{x_j \mid b_{ij} \neq 0, 1 \leq j \leq n_p, j \neq i \} \) be the set of correlated system states of \( x_i \). Based on the elements in the sets \( \{\mathcal{X}, A_i \} \) \( 1 \leq i \leq n_p \), the system states \( x \)
can be rearranged as $[x'_1, \ldots, x'_p]'$ ($1 \leq i \leq n_p$). The elements in $x_i$ have the following properties: for $x_p^i$ and $x_q^j$ ($i \neq j$), if $i \neq j$, we have $X_p^i \cap X_q^j = \emptyset$ and $A_p \cap A_q = \emptyset$, where $x_q^j$ represents the $j^{th}$ element of $x_i$. Due to the independence between the system states $x_i$ and $x_j$, we focus on the node coordination design for $x_i$. To avoid confusion, we redefine the subscripts of the elements in $x_i$, as $x_i = [x_1, \ldots, x_\xi(x_i)]'$. Hence, the dynamic model of system states $x_i$ is given by

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k) + \omega_i(k), \quad 1 \leq i \leq \kappa,$$  
(4)

where $A_i$, $B_i$, $u_i$ and $\omega_i(k)$ are the corresponding parts of $A$, $B$, $u$ and $\omega(k)$, respectively, under the given $x_i$, and $\omega_i(k) \sim N(0, Q_i)$.

Denote $A_i$ as the set of responsible actuators of the system states $x_i$. If $x^* - \epsilon \leq x_i \leq x^* + \epsilon$, we have $u_i = 0$, else the actuators in the set $A_i$ should coordinate with each other to perform appropriate actions, since the control of system states $x_p^i$ and $x_q^j$ ($p \neq q$) are now correlated. Without loss of generality, we assume that at least one event occurs in the system states $x_i$ at the $k^{th}$ step.

Using the Voronoi cells method [32], the ROI can be divided into $n_a$ partitions, where all the sensors in the $j^{th}$ partition are closer to the actuator $A_j$ than to all other actuators. Hence, the actuator $A_j$ periodically collects the measurements from the sensors in its own partition. This is because: 1) the nearest actuator is probably the one that covers the corresponding POI of the reported event, and, thus, it can take the action to react to this event quickly, and 2) the transmission over the shortest path is expected to suffer less interference and lower packet loss, and, thus, it has a higher reliability. As the example shown in Fig. 3, the actuators $A_1$ and $A_2$ are the nearest actuators to the POIs $P_1$ and $P_2$, as well as the responsible actuators of the system states $x_1$ and $x_2$.

**Fig. 3.** Distributed S-A coordination illustration.

Since the sensors that cover the system states $x_i$ may belong to different partitions, multiple actuators can receive the measurements of the system states $x_i$. We group these actuators as a set (cluster) $\Xi_i$, and select one actuator in the set $\Xi_i$ as the fusion center (cluster head). For example, in Fig. 3, the actuator $A_1$ receives the measurements of the system states $x_1$ and $x_2$ from the sensors $S_1$, $S_2$ and $S_6$, while the actuator $A_2$ receives the measurements of the system states $x_1$ and $x_3$ from the sensors $S_4$, $S_5$ and $S_6$. Therefore, the actuators $A_1$ and $A_2$ are grouped as a cluster $\Xi_1$ and we select the actuator $A_1$ as the cluster head. The fusion center selection can be implemented in a fully distributed manner [36]. To perform the data fusion, the cluster members send their received measurements to the cluster head. Note that the elements in the sets $\Xi_1$ and $\Xi_i$ ($i \neq j$) may be overlapping. To reduce the communication overhead of a single actuator, we consider one actuator cannot be selected as fusion center in different clusters (e.g., if the actuator $A_1$ is the fusion center in the cluster $\Xi_1$ and $A_1 \in \Xi_2$, the actuator $A_1$ cannot be selected as fusion center in the cluster $\Xi_2$).

**B. FKF-based Data Fusion**

Since the number of sensors is usually larger than the number of POIs (i.e., $n_s \gg n_p$), the system state $x_{tp}$ could be sensed by the multiple sensors (i.e., several measurements of system state $x_{tp}$ exist). Denote $Z_p(k) = \{z_p^j(k) | p \neq 0, 1 \leq j \leq n_s\}$ as the set of measurements of system state $x_{tp}$, and $Z_p^j(k)$ as the $j^{th}$ element of $Z_p(k)$. Hence, $z^j_p(k) = [z^j_p(k), \ldots, z^j_{\xi(x_i)}(k)]'$ represents the $j^{th}$ measurement vector of the system states $x_i$, which is given by

$$z^j_p(k) = C_i^j x_i(k) + \nu^j_p(k),$$

where $C_i^j$ and $\nu^j_p(k)$ have compatible dimensions, and $\nu^j_p(k) \sim N(0, R_i^j)$. Since the sensors are randomly deployed, the number of elements in the set $Z_i$ and in the set $Z_i (j \neq i)$ are not always the same. We assume that $h$ independent measurement vectors of $x_i$ can be obtained from the sets $\{Z_1, \ldots, Z_{\xi(x_i)}\}$. Hence, $h$ LF is implemented in parallel. As the example shown in Fig. 3, the $x_i$ actuator is selected as actuator 1, and $u_i(k-1)$, and the sensor measurements at the current step $k$ (i.e. $z^j_p(k)$), the $j^{th}$ LF ($1 \leq j \leq h$) performs the following state estimation:

$$P^j_i(k|k-1) = A_i^j P^j_i(k-1|k-1) A_i^j + Q^j_i(k-1),$$

$$x^j_i(k|k) = A_i^j x^j_i(k-1|k-1) + B_i^j u_i(k-1),$$

$$P^j_i(k|k) = P^j_i(k|k-1) - C_i^j R_i^j C_i^j,$$

$$x^j_i(k|k) = P^j_i(k|k) (P^j_i(k|k-1) x^j_i(k|k-1) + P^j_i(k|k-1) x^j_i(k|k-1),$$

(8)

where $x^j_i(k|k)$ and $x^j_i(k+1|k)$ are the a posteriori estimate and the a priori estimate of the system states $x_i(k)$ obtained by the $j^{th}$ LF, respectively. $P^j_i(k|k)$ and $P^j_i(k+1|k)$ are the error covariance matrices corresponding to the state estimations $x^j_i(k|k)$ and $x^j_i(k+1|k)$, respectively.

2) Collecting the estimation results $\{z^j_p(k|k), P^j_i(k|k)\}$ ($1 \leq j \leq h$) from the LFs, the MF performs the following data fusion process:

$$P_i(k|k) = \sum_{j=1}^{h} P^j_i(k|k),$$

$$x_i(k|k) = P_i(k|k) \sum_{j=1}^{h} (P^j_i(k|k) x_i(k|k)).$$

(10)

3) The MF resets the fusion results $x_i(k|k)$ and $P_i(k|k)$ as follows:

$$\hat{x}_i(k|k) = x_i(k|k),$$

$$P_i(k|k) = \beta_i(k|\kappa) P_i(k|k),$$

$$Q_i(k) = \beta_i(k|\kappa) Q_i.$$
where $\beta_i^j(k)$ is called the information-sharing factor and it must satisfy the constraints $0 \leq \beta_i^j(k) \leq 1$ and $\sum_{j=1}^h \beta_i^j(k) = 1$ to avoid information loss. We select the value of $\beta_i^j(k)$ based on the estimation accuracy of the $j^{th}$ LF [1] (e.g., $\beta_i^j(k) = \frac{(tr(P_i^j(k)|k)))^{-1}}{\sum_{j=1}^h (tr(P_i^j(k)|k)))^{-1}}$).

4) The MF sends the reseted results $\hat{x}_i^j(k|k)$, $\hat{P}_i^j(k|k)$ and $\hat{Q}_i^j(k|k)$ back to the $j^{th}$ LF for the next round of data fusion.

Lemma 4.1: The fusion result (10) provided by the MF is an optimal estimation of the system states $x_i(k)$.

Proof: The proof is similar to that in [37], where the FKF with data reset (11) – (13) is equal to the centralized Kalman filter.

(4) and (10) show that 1) $\kappa$ fusion centers perform state estimation simultaneously, and 2) each fusion center is only responsible for the estimation of partial system states (e.g., $x_i$). Hence, the proposed S-A coordination is distributed. The implementation details are summarized in Algorithm 1. Note that 1) the actuators’ outputs are not known at the sensors, and 2) the actuators are more powerful than the sensors. In order to reduce the amount of message exchanges among the sensors and the actuators and make a usage of nodes’ resources, the computationally expensive tasks, such as state estimation and data fusion, are performed by the actuators rather than the sensors.

Algorithm 1: Distributed S-A coordination

Set initial values: $x(0)$ and $P(0)$;
for $k = 1, 2, 3, \ldots$ do
  for $i = 1, \ldots, n$, do
    for Actuators in set $A_i$ do
      Receive the data transmitted from the sensors in its own partition;
      if Actuator $A_i$ is fusion center then
        Receive the sensor data transmitted from the other actuators in the cluster $\Xi_i$;
        Construct $\{x_i^j(k), \ldots, x_i^h(k)\}$;
        for $j = 1, \ldots, h$ do
          Update $\{x_i^j(k|k), P_i^j(k|k)\}$ through (5) – (8);
        end
       Fuse $\{x_i^j(k|k), P_i^j(k|k)\}$ (1 $\leq j \leq h$) through (9) – (10);
      end
      for $j = 1, \ldots, h$, do
        Reset $\{x_i^j(k|k), P_i^j(k|k)\}$ through (11) – (13);
      end
      else
        Relay the received sensor data to the fusion center in the cluster $\Xi_i$;
      end
    end
  end
end

V. A-A COORDINATION AND DISTRIBUTED CONTROL

In this section, we describe the distributed algorithm for solving the PP (3). Our method is based on dual decomposition and we solve the dual problem using the subgradient algorithm.

To develop the dual problem of PP, we introduce the positive Lagrange multipliers $\lambda = [\lambda_1, \ldots, \lambda_{n+a+n}]^\top$ to its constraints, the corresponding Lagrangian [38] is

$$L(u(k), \lambda) = \|Ax(k|k) + Bu(k|k) - x_\ast\|^2 + \alpha \sum_{j=1}^{n_a} E_j(k) + \sum_{j=1}^{n_a+n_p} \lambda_j(E_j(k) - E'_j) + \sum_{j=\min(a+1)}^{\kappa \alpha} \lambda_j(T_j(k) + T_d - T_\theta).$$

In (14), $x(k|k) = [x_1(k|k), \ldots, x_m(k|k)]$ and $x_\ast(k|k)$ (1 $\leq i \leq \kappa$) is given by (10).

The dual function $\mathcal{R}(\lambda)$ is defined as the minimum value of the Lagrangian $L(u(k), \lambda)$ over the variables $u$, i.e., for $\lambda$,

$$\mathcal{R}(\lambda) = \min_{u \leq u \leq \pi} L(u(k), \lambda) = \min_{i=1}^{\kappa} \mathcal{R}_i(\lambda).$$

Hence, the Dual Problem (DP) associated with the PP is

$$\text{DP} : \max_{\lambda \geq 0} \mathcal{R}(\lambda).$$

Since the PP is convex, the optimal objective function values of the PP and its dual problem are equivalent due to the strong duality (i.e., solving the DP is equal to solving the PP). Based on the structure of the DP, we design a two-layer subgradient-based algorithm to solve this problem. The inner-layer iteration aims to update variables $u$ under the given multipliers $\lambda$. On the other hand, the outer-layer iteration aims to update multipliers $\lambda$ under the given variable $u$. Therefore, we introduce the indexes $m$ and $l$ to count the outer-layer and the inner-layer iterations, respectively.

1) Inner-layer Iteration: Assume that the current outer-layer iteration is $m$. Under the given multipliers $\lambda(m)$, the actuator’s output $u_j$ (the $j^{th}$ element in $u_i$) is iteratively updated by

$$u_j(l + 1) = u_j(l) - \frac{\partial L(u(l), \lambda(m))}{\partial u_j(l)}.$$ 

Moreover, $\delta$ is a positive small step-size, and

$$\frac{\partial L_i(u_i(l), \lambda_i(m))}{\partial u_j(l)} = \sum_{i=1}^{D(u_i)} \sum_{j=1}^{D(u_i)} a_{ij}x_j + \sum_{j=1}^{D(u_i)} b_{ij}u_j(l) - x_\ast.\frac{\partial T_j(l)}{\partial u_j(l)} + 2(\alpha + \lambda_j(m))\Delta u_j(l) + \sum_{j=\min(a+1)}^{\kappa \alpha} \lambda_j(m)\Delta u_j(l),$$

where $\delta$ is a positive small step-size, and

$$\frac{\partial L_i(u_i(l), \lambda_i(m))}{\partial u_j(l)} = \sum_{i=1}^{D(u_i)} \sum_{j=1}^{D(u_i)} a_{ij}x_j + \sum_{j=1}^{D(u_i)} b_{ij}u_j(l) - x_\ast.$$
Denote $\varepsilon \geq 0$ as a small tolerance. Since $L(u, \lambda)$ is a convex function with respect to $u$, the inner-layer iteration can start from an arbitrary initial point $u(0)$ and the iteration stops when $\|u(l+1) - u(l)\|_2^2 \leq \varepsilon$. Hence, by iterating (16), we obtain $u(m)$ (the optimal solution of the problem (15) under the given multipliers $\lambda_i(m)$).

2) Outer-layer Iteration: Based on the updated results of the previous inner-layer and outer-layer iterations (i.e., $u(m)$ and $\lambda_i(m)$), the multiplier $\lambda_j$ (the $j^{th}$ element in $\lambda_i$) is iteratively updated by

$$
\lambda_j(m + 1) = \left[ \lambda_j(m) + \delta \frac{\partial R_i(\lambda_i(m))}{\partial \lambda_j(m)} \right]^+ \leq 1, 1 \leq j \leq D(\lambda_i),
$$

where

$$
\frac{\partial R_i(\lambda_i(m))}{\partial \lambda_j(m)} = \begin{cases} E_j(m) - E_i^*, 1 \leq j \leq D(u_i), \\
T_j(m) + T_{ij} - T_{ih}, D(u_i) + 1 \leq j \leq D(\lambda_i).
\end{cases}
$$

Since $R(\lambda)$ is a concave function with respect to $\lambda$, the outer-layer iteration can start from an arbitrary initial point $\lambda(0)$ and the iteration stops when $\|\lambda_i(m + 1) - \lambda_i(m)\|_2^2 \leq \varepsilon$.

(16) show that $\frac{\partial R_i(\lambda_i(m))}{\partial \lambda_j(m)}$ is a function with respect to $u_i(l)$. Hence, during the inner-layer iteration, the actuator $A_j$ needs to exchange its temporary result $u_j(l)$ with the other actuators in the set $A_i$ to update $u_j(l + 1)$. On the other hand, (17) shows that $\frac{\partial R(\lambda_i(m))}{\partial \lambda_j(m)}$ is a function with respect to $u_i(l)$ and $\lambda_j(m).$ Note that $u_i(l)$ is already known by the actuator $A_j$ through the inner-layer iteration. The update of multipliers $\lambda_j(m)$ can be performed locally by each actuator in the set $A_i$. (14) and (17) show that 1) the actuator $A_i$ only needs to exchange information with the actuators in the set $A_i$, rather than all the actuators, and 2) the actuator $A_j$ calculates its output individually based on the received information. Hence, the proposed A-A coordination is distributed. The implementation details are summarized in Algorithm 2.

Since the PP is convex, the optimal values of $\lambda$ and $u$ exist. Denote them as $\lambda^*$ and $u^*$, respectively. The convergence of proposed algorithm is analyzed through the following theorems.

**Definition 5.1 (Statistical convergence [39]):** For an optimization problem $\min_u S(u)$ and an iterative algorithm with positive step-size $\varepsilon$, which generates an intermediate solution $\lambda(m)$ at the $m^{th}$ step, let $\bar{u}$ be the optimal solution and $\lambda(m) = \frac{1}{m} \sum_{\tau=1}^m u(\tau)$ be the average solution found by the $m^{th}$ step. $\lambda$ is said to statistically converge to $\lambda^*$, if for any $\gamma > 0$ there exists an $\varepsilon$ such that $\lim_{m \to \infty} \sup_{u \in U} \left\{ S(\bar{u}(m)) - S(\bar{\lambda}(m)) \right\} \leq \gamma$.

**Theorem 5.1:** Iterating (17), the Lagrange multipliers $\lambda$ statistically converge to $\lambda^*$, when $\delta$ is a small enough value.

**Proof:** See Appendix B for the proof.

**Theorem 5.2:** Iterating (16) and (17), the actuators’ outputs $u$ statistically converge to $u^*$, when $\delta$ is a small enough value.

**Proof:** See Appendix C for the proof.

From the above statements, we can see that there exist two types of information exchange among the actuators: 1) the actuators in the set $\Xi_i$ send their received measurements to the fusion center for the data fusion, and 2) the actuators in the set $\mathcal{A}_i$ exchange their temporary results $u_i(l)$ with each other to calculate their outputs $u_i(k)$. The former one only requires to exchange information once at each step, while the latter one requires several times at each step. Based on different positions of nodes and POIs, the actuators in the sets $\Xi_i$ and $\mathcal{A}_i$ are not always the same. According to the actuators in the sets $\{\Xi_i\}$ and $\{\mathcal{A}_i\}$, each actuator knows which actuators should communicate with. Hence, the topology of communication between the actuators can be determined by the available methods [36]. Moreover, by Algorithm 2, each actuator in the set $\mathcal{A}_i$ knows the values of $u_i(k)$, which can be used for the next round data fusion.

### VI. SIMULATION RESULTS

In this section, we consider 30 actuators and 100 sensors randomly deployed in a 150 m x 150 m ROI to monitor and control the temperature of 30 POIs. The values of the experimental set-up are described in Table I.

#### A. Performance Evaluation

Denote $J_1(k) = \|Ax(k) + Bu(k) - x^*\|_2^2$ and $J_2(k) = \alpha \sum_{j=1}^{\eta} E_j(k)$ as the control error and the actuator energy consumption at the $k^{th}$ step, respectively. Fig. 4(a) shows the dynamic change of the objective function $J(k) = J_1(k) + J_2(k)$ using the proposed method. From it we can see that 1) the system is Bounded-Input-Bounded-Output (BIBO) stable, and 2) the proposed method satisfies the desired user’s requirements and achieves a quick response to the events with a small energy consumption. Fig. 4(b) shows the system performance (objective function) to handle sequential events. Denote $e_i$ as the event occurs at the system state $x_i$. Let $\varepsilon_1 = \{e_1, \ldots, e_{10}\}$, $\varepsilon_2 = \{e_{11}, \ldots, e_{20}\}$ and $\varepsilon_3 = \{e_{21}, \ldots, e_{30}\}$. Case 1 represents that the events $\varepsilon_1$, $\varepsilon_2$ and $\varepsilon_3$ occur at step $k = 1$ simultaneously. Case 2 and Case 3 represent that the events

---

**Algorithm 2: Distributed A-A coordination**

Set the initial values: $u(0)$, $\lambda(0)$, $\delta$, and $\varepsilon$;
for $k = 1, 2, 3, \ldots$ do
for $i = 1, \ldots, \kappa$ do
  for Actuator $A_j$, $j = 1, \ldots, D(u_i)$ do
    $m \leftarrow 1$
    Update $\lambda_i(m)$ by (17);
    if $\|\lambda_i(m) - \lambda_i(m-1)\|_2^2 \geq \varepsilon$ then
      Update $\lambda_i(m+1)$ by (17);
      $l \leftarrow 1$
      Update $u_j(l)$ by (16);
    else $u_i(m) = u_j(l)$;
    $m \leftarrow m + 1$
    else
      $u_j(k) = u_j(m)$;
  end
end
end
TABLE I: EXPERIMENTAL SET-UP

<table>
<thead>
<tr>
<th></th>
<th>System model</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System matrix A</strong></td>
<td>$a_{ij} = \begin{cases} 1 - \frac{n_i}{r_\alpha}, &amp; \text{if } d_{ij} &lt; r_s, \ i \neq j, \ 0.9 &amp; \ i = j, \ 0 &amp; \text{else.} \end{cases}$</td>
</tr>
<tr>
<td></td>
<td>$d_{ij}$: the distance between $P_i$ and $P_j$</td>
</tr>
<tr>
<td></td>
<td>$r_s$: the impact range of system state</td>
</tr>
<tr>
<td></td>
<td>$r_\alpha$: the action range of actuator</td>
</tr>
<tr>
<td></td>
<td><strong>Input matrix B</strong></td>
</tr>
<tr>
<td></td>
<td>$b_{ij} = \begin{cases} 1 - \frac{d_{ij}}{r_a}, &amp; \text{if } d_{ij} &lt; r_a, \ 0 &amp; \text{else.} \end{cases}$</td>
</tr>
<tr>
<td></td>
<td>$d_{ij}$: the distance between $P_i$ and $A_j$</td>
</tr>
<tr>
<td></td>
<td>$r_a$: the action range of actuator</td>
</tr>
<tr>
<td></td>
<td><strong>Measurement matrix C</strong></td>
</tr>
<tr>
<td></td>
<td>$c_{ij} = \begin{cases} 1 - \frac{d_{ij}}{r_s}, &amp; \text{if } d_{ij} &lt; r_s, \ 0 &amp; \text{else.} \end{cases}$</td>
</tr>
<tr>
<td></td>
<td>$d_{ij}$: the distance between $S_i$ and $P_j$</td>
</tr>
<tr>
<td></td>
<td>$r_s$: the sensing range of sensor</td>
</tr>
<tr>
<td>$n_a = 30$</td>
<td>$n_k = 100$</td>
</tr>
<tr>
<td>$r_s = 2$ m</td>
<td>$r_a = 20$ m</td>
</tr>
<tr>
<td>$r_\alpha = 5$ m</td>
<td>$T_d = 2$ s</td>
</tr>
<tr>
<td>$\alpha = 200$</td>
<td>$\Delta = 0.5$ s</td>
</tr>
<tr>
<td>$x_s = 25$ °C</td>
<td>$e_s = 0.5$ °C</td>
</tr>
<tr>
<td>$Q = 0.1 I$</td>
<td>$R = 0.5 I$</td>
</tr>
<tr>
<td>$t_{1,m}^t(k) = \frac{kx_{d_{ij}}e_i(k)}{k_p u_j(k)}$</td>
<td>$T_{th} = \sum_{j=1}^{n_p} \max_{i=1}^{n_a} t_{1,m}^t(k) + T_d$</td>
</tr>
<tr>
<td><strong>Actuator $A_i$ characteristics</strong></td>
<td>$k_i = 2$</td>
</tr>
<tr>
<td></td>
<td>$\eta_j \in [0, 0.2]$</td>
</tr>
<tr>
<td></td>
<td>$t_{i}^t(k) = \frac{kx_{d_{ij}}e_i(k)}{p_i(k)}$</td>
</tr>
<tr>
<td></td>
<td>$G = \sum_{j=1}^{n_p} \eta_j t_{i}^t(k)$</td>
</tr>
</tbody>
</table>

$\mathcal{E}_1$, $\mathcal{E}_2$ and $\mathcal{E}_3$ occur at steps $k = 1, 3, 5$ and $k = 1, 20, 40$, respectively. In Case 1, the actuators take 4 steps to handle all the events. In Case 2, new events occur when the previous tasks haven’t finished yet. Hence, the actuators’ outputs change accordingly so as to match the characteristics of the occurring events. In Case 3, the actuators are in the idle state, when the assigned tasks are finished, and some actuators are activated again when new events occur. Fig. 4(a) and Fig. 4(b) show that the proposed method efficiently handles the parallel and sequential events.

The algorithm convergence (parallel events) is shown in Fig. 4(c), where $n_s(k)$ and $n_a(k)$ are the number of outer-layer iterations and the total number of inner-layer iterations at the $k^{th}$ step, respectively. In the beginning, the LDA requires at most eight times of outer-layer iterations, and at each outer-layer iteration, the actuator $A_j$ only needs to exchange a small amount of information (i.e., its temporary output $u_j(l)$ six times in average). With the step $k$ increasing, the outer-layer and inner-layer iterations numbers $n_s(k)$ and $n_a(k)$ decrease very fast, since the objective function $\mathcal{J}$ has converged. Hence, the communication overhead of the actuator $A_i$ is small. For the sequential events, we obtain the similar results.

Random events (i.e., the events randomly occur at the system states \{x\}) are very common in practice, especially without a priori knowledge about when and where the events will occur. In Fig. 4(d), we evaluate the system performance to handle random events. Denote $\mathcal{J}_o = \sum_{i=1}^{n} \mathcal{J}(Z_i)$ as the average value of the objective function achieved by 50 steps, and $n_a$ as the number of random events. We set $n_a = n_s = n_p = 100$, and the random events occur at step $k = 1$. The sensors, the actuators, and the POIs are randomly deployed in the ROI, and the event probability $p_e = \frac{n_a}{n}$ is changed from 0.2 to 1. Although $\mathcal{J}_o$ generally increases with $p_e$, the convergence speeds of LDA (i.e., the number of outer-layer iterations) under different $p_e$ are almost the same.

B. Comparison with Existing Algorithms

In multi-sensor scenario, one of the popular methods to perform state estimation is the Centralized Kalman Filter (CKF) [37]. CKF gathers the measurements from all the sensors in one processing block (i.e., $z(k) = [z^{1}(k)^{T}, \ldots , z^{n}(k)^{T}]$), and runs KF to get the optimal estimation of system states. Fig. 5(a) compares the Root-Mean-Square Error (RMSE) of the estimation result achieved by CKF and FKF. Since that the RMSE of FKF and CKF are the same, and, thus, the FKF with data reset is an optimal state estimator. In WSNs, the data transmission between the sensors and the actuators are prone to time delay/packet loss, due to the limited computation/communication abilities of the sensors and the existence of the internal/external interferences [11]. If the unreliable sensors’ measurements are used for state estimation, and they are undetected by both the MF and LFs, they will contaminate the fusion results. Since the measurement residuals $z^l_i(k) - C^l_i x^l_i(k) - 1$ of the $j^{th}$ LF are Gaussian distributed, a simple $\chi^2$ detector [40] can be employed to detect and isolate the faulty sensor measurement. We assume that if sensor $S_j$ suffered from time delay/packet loss at step $k$, the measurement received by the actuator is $z_j(k) = 0$. We randomly select 5 sensors to suffer from time delay/packet loss during the steps $k = 2 \sim 7, 25, 35$. The result is shown in Fig. 5(b). From it we can see that the estimation accuracy of CKF significantly drops. However, the estimation accuracy of FKF is almost not affected by the faulty measurements. Since CKF/FKF involves the multiplication and inversion of $n \times n$ matrices, the computational complexity is $O(n^3)$ [41]. For FKF and CKF, we have $n = D(z^l_i)$ and $n = hD(z^l_i)$, respectively. Although the CKF has a simple structure (i.e., no data fusion and reset), the distributed FKF-based S-A coordination achieves a better balance between the estimation accuracy and the computational complexity.

During the A-A coordination, we compare the proposed Lagrange-based Distributed Algorithm (LDA) with three ex-
isting methods: 1) PID Neural Network (PIDNN) [13], [30],
2) Gradient Descent method (GD) [7], [31], and 3) Sequential Unconstrained Minimization Technique (SUMT) [1]. The
results are shown in Fig. 6. From it we can see that all four algorithms achieve the desired control accuracy. However,
the error convergence speeds of LDA and SUMT are faster than that of PIDNN and GD. This is because: 1) PIDNN needs some steps to train the neural network in the beginning, and 2) LDA and SUMT minimize \( \mathcal{J}(k) \) at each step \( k \) but PIDNN and GD minimize \( \mathcal{J}(k) \) step by step. Since \( \mathcal{J}(k) \) takes control error as well as energy consumption into account, a fast convergence speed implies a better real-time and energy-efficiency performance.

\[
\mathcal{J}(k) = \sum_{i=1}^{n_{f}} a_{ij} - d_{ij}
\]

Fig. 6. System performance comparison between different control schemes.

C. Robustness Analysis

The system model described by (1) has a general form (i.e., system matrix \( A \) and input matrix \( B \) are not limited to diagonal matrices). (1) also shows that the system states \( x(k) \) are controlled by the actuators’ outputs \( u(k) \), and, thus, the actuator failure affects system performance. Denote \( n_{f} \) as the number of failing actuators. We assume that \( u_{j}(k) = 0 \), if the actuator \( A_{j} \) is out of function. As shown in Fig. 7(a), with \( n_{f} \) increasing, more and more POIs become uncontrollable. Hence, \( \mathcal{J}(k) \) increases with \( n_{f} \). On the other hand, since one POI may be covered by the multiple actuators, the failure of few actuators is still tolerable (e.g., \( n_{f} = 4 \)).

From (16), we can see that the system performance depends on the system model accuracy. Fig. 7(b) shows the influence of inaccuracy affect of system model. Denote \( g_{d} = \sum_{j=1}^{n_{p}} \sum_{i=1}^{n_{p}} |a_{ij} - \bar{a}_{ij}| + \sum_{j=1}^{n_{p}} \sum_{i=1}^{n_{p}} |b_{ij} - \bar{b}_{ij}| \) as a model error index, where \( \{A, B\} \) is the accurate system model used to simulate the dynamic system, and \( \{\bar{A}, \bar{B}\} \) is the inaccurate system model used to calculate actuators’ outputs \( u(k) \). When \( g_{d} \) is small, \( \mathcal{J}(k) \) remains almost unchanged. However, for a large \( g_{d} \), the system performance degrades or even becomes unstable. In order to achieve a desired control quality, a certain level of system model accuracy is essential. Therefore, we can use experimental/mechanism modeling [6], [7] or jointly

\[
\Delta J(k) = J(k) - J(k-1)
\]

estimate the system states \( x(k) \) and the system model \( \{A, B\} \) through the adaptive filtering methods [42], [43].

D. Scalability Evaluation

We compare the system performances achieved by LDA and SUMT under different network scales, with \( n_{s} = n_{p} \) and \( n_{s} = 100 \). The sensors, the actuators, and the POIs are randomly deployed in the ROI so that the system partition is varied among the experiments. To show the difference between these two algorithms, we define a metric \( \Delta \mathcal{J}(k) = \mathcal{J}(k) - \mathcal{J}_{s}(k) \times 100\% \), where \( \mathcal{J}(k) \) and \( \mathcal{J}_{s}(k) \) are the objective function values achieved by LDA and SUMT at the \( k^{th} \) step, respectively. Fig. 8 shows that 1) the performances of the two algorithms are quite similar, as the difference is less than \( 8\% \), and 2) with \( n_{p} \) increasing, the computing time of both algorithms grows. Since SUMT is an interior point method, the computational complexity is \( O(n_{s}^{3}) \) [44]. On the other hand, since the inter-layer and outer-layer iterations of LDA are based on subgradient method, the computational complexity is \( O(U^{2}R^{2} \varepsilon^{-2}) \) [45], where \( U \) is the distance between an optimal solution and the initial point and \( R \) is a Lipschitz constant for the objective function. Hence, the LDA has much shorter computing time than the SUMT. Although the centralized method is able to collect the global information and achieve a globally optimal control decision, the communication latency is also large, since the sensory data and the control command are routed through the sink rather than the nearby nodes. The communication and the computation overhead at the sink and at the nodes near to the sink increase significantly with the network scale.

\[
J(k) = \sum_{i=1}^{n_{f}} a_{ij} - d_{ij}
\]

Fig. 7. System performance under abnormal conditions.

Fig. 8. Scalability comparison between SUMT and LDA.

Fig. 9 evaluates the system performance under different node configurations. We consider following two cases: 1) \( n_{p} = 50 \) and \( n_{s} \) changes from 50 to 100; 2) \( n_{s} = 50 \) and \( n_{p} \) changes from 50 to 100. The sensors, the actuators, and the POIs are randomly deployed in the ROI. In Case 1, with \( n_{s} \) decreasing, \( \mathcal{J}_{s} \) increases. This is because the more actuators are involved
in event processing, the easier is to handle the events. In Case 2, with \( n_p \) increasing, \( \Delta_k \) increases. This is because 1) each actuator only covers a limited area in the ROI, and 2) the POIs are randomly deployed. With \( n_p \) increasing, some POIs are out of actuators’ control, which leads to a large control error.

![System performance under different node configurations.](image)

**Fig. 9.** System performance under different node configurations.

### VII. CONCLUSION

The distributed node coordination problem in WSANs is addressed in this paper. We aim at satisfying the user’s requirements on the system states in a real-time and energy-efficient manner. The node coordination problem was formulated as a non-linear programming, which was proved to be convex. We applied a distributed estimation method, which is based on Voronoi cells and FKF, to perform S-A coordination. Based on the estimated result, we proposed a Lagrange-based distributed algorithm to adjust actuators’ outputs and proved its convergence. Simulation results demonstrated the performance of the proposed method in terms of the quality of the solution and the computing time.

### ACKNOWLEDGMENT

This research was partly supported by the National Natural Science foundation of China (Grant No. 61403340 and 61573103), the INRIA post-doctoral research fellowship program, the State Key Laboratory of Synthetical Automation for Process Industries, and the Fundamental Research Funds for the Central Universities.

### APPENDIX A

**PROOF OF THEOREM 3.1**

*Proof:* From \( E_j(k) = \Delta p_j(k) = \Delta k_p u_j(k)^2 \), we have \( \frac{\partial^2 E_j(k)}{\partial u_j(k)^2} = 2 \Delta k_p, \frac{\partial^2 E_j(k)}{\partial u_j(k)^2} = 0, \frac{\partial^2 E_j(k)}{\partial u_i(k) \partial u_j(k)} = 0 \) (\( l \neq j \)). And further, \( \nabla^2 (\sum_{j=1}^{n_a} E_j(k)) = \sum_{j=1}^{n_a} \nabla^2 E_j(k) = 2 \Delta k_p I_{n_a \times n_a} \). Hence, the Hessian of \( E_j(k) \) and \( \sum_{j=1}^{n_a} E_j(k) \) are positive semi-definite (i.e., \( E_j(k) \) and \( \sum_{j=1}^{n_a} E_j(k) \) are convex). Note that the second norm is convex (e.g., \( \| A \| \otimes \| B \| = \| A \otimes B \| \)), and the sum operation preserves the convexity. The objective function of the primal problem is convex. Let \( G(k) = G(e_1(k), U_1(k)) \). Since \( \frac{\partial^2 G(k)}{\partial u_l(k) \partial u_j(k)} = 0 \) (\( l \neq j \)), the Hessian of \( T_k(k) \) is positive semi-definite. Hence, the constraints of the primal problem are convex. According to the definition of convex problem [44], the PP (3) is a convex optimization problem.

### APPENDIX B

**PROOF OF THEOREM 5.1**

*Proof:* Let \( \mathcal{H}_j(m) = \frac{\partial R(\mu(m))}{\partial \mu(m)} \). Therefore, \( \mathcal{H}_j(m) = \left[ \mathcal{H}_{j_1}(m), \ldots, \mathcal{H}_{n_a + n_p}(m) \right]^T \) is the subgradient of dual function \( R \). We construct a Lyapunov function \( \mathcal{V}(\mu(m)) = \sum_{j=1}^{n_a + n_p} (\lambda_j^* - \lambda_j(m))^2 \).

According to \( \lambda_j(m+1) = [\lambda_j(m) + \delta \mathcal{H}_j(m)]^+ \), we have \( \lambda_j(m+1) \geq 0 \) and \( \lambda_j(m) \geq 0 \). Since \( \delta > 0 \) and \( \mathcal{H}_j(m) \leq 0 \), we get \( \lambda_j(m+1) \geq \lambda_j(m) + \delta \mathcal{H}_j(m) \). And further, due to \( \lambda_j^* \geq 0 \), we have \( (\lambda_j^* - \lambda_j(m+1))^2 \leq (\lambda_j^* - \lambda_j(m) - \delta \mathcal{H}_j(m))^2 \).

Based on this inequality, we obtain

\[
\mathcal{V}(\mu(m + 1)) = \sum_{j=1}^{n_a + n_p} (\lambda_j^* - \lambda_j(m + 1))^2 \\
\leq \sum_{j=1}^{n_a + n_p} (\lambda_j^* - \lambda_j(m) - \delta \mathcal{H}_j(m))^2 \\
= \mathcal{V}(\mu(m)) + \sum_{j=1}^{n_a + n_p} 2 \delta \mathcal{H}_j(m)(\lambda_j(m) - \lambda_j^*) \\
+ \sum_{j=1}^{n_a + n_p} \delta^2 \mathcal{H}_j(m)^2.
\] (18)

Note that \( R(\mu) \) is concave. According to the definition of subgradient, we get

\[
\sum_{j=1}^{n_a + n_p} \mathcal{H}_j(m)(\lambda_j(m) - \lambda_j^*) \leq R(\mu(m)) - R(\mu^*). \] (19)

Substituting (19) into (18) and applying the inequalities recursively, we have

\[
\mathcal{V}(\mu(m + 1)) \leq \mathcal{V}(\mu(m)) + 2 \delta(R(\mu(m)) - R(\mu^*)) + \sum_{j=1}^{n_a + n_p} \delta^2 \mathcal{H}_j(m)^2 \\
\leq \mathcal{V}(\mu(1)) + \sum_{j=1}^{n_a + n_p} \delta^2 \mathcal{H}_j(m)^2 \\
+ \sum_{j=1}^{n_a + n_p} \delta^2 \mathcal{H}_j(m)^2. \] (20)

Let \( \bar{\lambda}(m) = \frac{1}{m} \sum_{j=1}^{n_a + n_p} \lambda_j(m). \) Utilizing the concavity of \( R(\mu) \) and Jensen’s inequality, we get

\[
\sum_{j=1}^{n_a + n_p} (R(\lambda(\tau)) - R(\lambda^*)) \leq m(R(\bar{\lambda}(m)) - R(\lambda^*)). \] (21)

Assume that \( \sum_{j=1}^{n_a + n_p} \mathcal{H}_j(m)^2 \leq B \) since \( \mathcal{H}(\tau) \) is bounded. Substituting (21) into (20) and noting that \( \mathcal{V}(\mu(m + 1)) \geq 0 \), we have

\[
R(\lambda^*) - R(\bar{\lambda}(m)) \leq \frac{\mathcal{V}(\lambda(1)) + mB}{2m\delta^2}.
\]

Hence, we get \( \lim_{m \to \infty} \sup_{\mu \to \lambda^*} (R(\lambda^*) - R(\bar{\lambda}(m))) \leq \frac{B}{2\delta^2} \). According to the definition of statistical convergence, given a small enough \( \delta \), \( \lambda \) statistically converges to \( \lambda^* \).

### APPENDIX C

**PROOF OF THEOREM 5.2**

*Proof:* We construct a Lyapunov function \( \mathcal{L}(\mu(m)) = \sum_{j=1}^{n_a + n_p} \lambda_j(m)^2 \).

Since \( \lambda_j(m+1)^2 \leq (\lambda_j(m) + \delta \mathcal{H}_j(m))^2 \), we have

\[
\mathcal{L}(\mu(m + 1)) \leq \sum_{j=1}^{n_a + n_p} \lambda_j(m + 1)^2.
\]

Thus, the system states converge to the given setpoints.
where $H_j^u(\mathbf{u})$ represents substituting $\mathbf{u}^*$ into $\frac{\partial R_u(\lambda(m))}{\partial \lambda(m)}$. (22a) holds under the given $\lambda(m)$, and (22b) holds since since $\delta \geq 0$, $\lambda_j(m) \geq 0$ and $H_j^u(\mathbf{u}^*) \leq 0$.

Utilizing the convexity of $\mathcal{J}(\mathbf{u})$ and Jensen's inequality, we get
\[
\mathcal{J}(\mathbf{u}(\tau)) \leq \frac{1}{\delta} \sum_{j=1}^{m} \left( \mathcal{J}(\mathbf{u}^*) - \mathcal{J}(\mathbf{u}(\tau)) \right) \leq m \delta \mathcal{J}(\mathbf{u}^*) - \mathcal{J}(\mathbf{u}(m)),
\]
(23)

where $\mathbf{u}(\tau) = \frac{1}{\delta} \sum_{j=1}^{m} \mathbf{u}(\tau_j)$. Substituting (23) into (22c) and noting that $\mathcal{J}(\mathbf{u}(m + 1)) \geq 0$, we have
\[
\mathcal{J}(\mathbf{u}(m)) - \mathcal{J}(\mathbf{u}^*) \leq \frac{\mathcal{J}(\lambda(1)) + m \delta \mathcal{J}(\mathbf{u}^*)}{2m \delta}.
\]

From $\limsup_{m \to \infty} \left( \mathcal{J}(\mathbf{u}(m)) - \mathcal{J}(\mathbf{u}^*) \right) \leq \frac{\delta \mathcal{J}(\mathbf{u}^*)}{2}$, we can get that $\mathbf{u}$ statistically converges to $\mathbf{u}^*$ when $\delta$ is a small enough value.

### REFERENCES


Lei Mo (S’13–M’17) is currently a Postdoctoral Fellow with INRIA Rennes Bretagne Atlantique research center, France. He received the B.S. degree from College of Telecom Engineering and Information Engineering, Lanzhou University of Technology, Lanzhou, China, in 2007, and the Ph.D. degree from College of Automation Science and Engineering, South China University of Technology, Guangzhou, China, in 2013. From 2013 to 2015, he was a research fellow with the Department of Control Science and Engineering, Zhejiang University, China. From 2015 to 2017, he was a research fellow with INRIA Nancy Grand Est research center, France. His current research interests include networked estimation and control in wireless sensor and actuator networks, cyber-physical systems, task mapping and resource allocation in multi-cloud systems. He serves/served as a Guest Editor for IEEE Access and Journal of Computer Networks and Communications. He also serves/served as a TPC Member for WF-5G 2018, Glebocom-MWN 2018 and COMNETSAT 2018.

Yeqiong Song received the B.S. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 1984, the Master degree from the Ecole National Supérieure des Télécommunications (ENST), Paris, France, in 1987, the M.Sc. degree (DEA) from the University of Paris 6 in 1988, the Ph.D. degree from the Institut National Polytechnique de Lorraine (INPL), Nancy, France, in 1991, and the Habilitation to lead research (HDR) degree from the University of Henri Poincaré Nancy 1 in 2004, all in telecommunications and computer science. In 1988, he joined Lorraine Laboratory of IT Research and Applications (LORIA) in Nancy, France. From 1992 to 2005, he was an Associate Professor with the University of Henri Poincaré Nancy 1. From 2001 to 2003, he was a Full-Time Researcher with INRIA Lorraine. Since 2005, he is a full Professor of computer science with the INPL and now with the University of Lorraine. His research interests include modeling and performance evaluation of networks and real-time distributed systems including networked control systems, and the implementation of real-time quality-of-service mechanisms in industrial networks, in-vehicle networks, and wireless IoT and sensor networks. He authored/co-authored more than 100 scientific papers.

Xianghui Cao (S’08–M’11–SM’16) received the B.S. and Ph.D. degrees in control science and engineering from Zhejiang University, Hangzhou, China, in 2006 and 2011, respectively. From 2007 to 2009, he was a visiting scholar with the Department of Computer Science, University of Alabama, USA. From 2012 to 2015, he was a senior research associate with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, USA. He is currently an associate professor with the School of Automation, Southeast University, Nanjing, China. His research interests include cyber-physical systems, wireless network performance analysis, wireless networked control, and network security. He serves as the Publicity Co-Chair for ACM MobiHoc 2015, Symposium Co-Chair for ICNC 2017 and IEEE/CIC ICC 2015, and a TPC Member for a number of conferences. He also serves as an Associate Editor for several journals, including ACTA Automatica Sinica, IEEE/CAA Journal of Automatica Sinica, KSI Transactions on Internet and Information Systems, Security and Communication Networks, and International Journal of Ad Hoc and Ubiquitous Computing. He was a recipient of the Best Paper Runner-Up Award from ACM MobiHoc 2014.

Angeliki Kritikakou is currently an Associate Professor at University of Rennes 1 and IRISA - INRIA Rennes Bretagne Atlantique research center. She received her Ph.D. in 2013 from the Department of Electrical and Computer Engineering at University of Patras, Greece and in collaboration with IMEC Research Center, Belgium. She worked for one year as a Postdoctoral Research Fellow at the Department of Modelling and Information Processing (DTIM) at ONERA in collaboration with Laboratory of Analysis and Architecture of Systems (LAAS) at the University of Toulouse, France. Her research interests include embedded systems, real-time systems, mixed-critical systems, hardware/software co-design, mapping methodologies, design space exploration methodologies, memory management methodologies, low power design and fault tolerance.