Distributed Consensus Algorithm for Events Detection in Cyber Physical Systems

Shancang Li, Shanshan Zhao, Po Yang, Panagiotis Andriotis, Lida Xu, and Qindong Sun

***Abstract*—In the harsh environmental conditions of cyber physical systems (CPS), the consensus problem seems to be one of the central topics that affect the performance of consensus-based applications, such as events detection, estimation, tracking, blockchain, etc. In this paper, we investigate the events detection based on consensus problem of CPS by means of compressed sensing (CS) for applications such as attack detection, industrial process monitoring, automatic alert system, and prediction for potentially dangerous events in CPS. The edge devices in a CPS are able to calculate a log-likelihood ratio (LLR) from local observation for one or more events via a consensus approach to iteratively optimize the consensus LLRs for the whole CPS system. The information-exchange topologies are considered as a collection of jointly connected networks and an iterative distributed consensus algorithm is proposed to optimize the LLRs to form a global optimal decision. Each active device in the CPS first detects the local region and obtains a local LLR, which then exchanges with its active neighbors. Compressed data collection is enforced by a reliable cluster partitioning scheme, which conserves sensing energy and prolongs network lifetime. Then the LLR estimations are improved iteratively until a global optimum is reached. The proposed distributed consensus algorithm can converge fast and hence improve the reliability with lower transmission burden and computation costs in CPS. Simulation results demonstrated the effectiveness of the proposed approach.**

.

***Index Terms*—Consensus algorithm, data gathering, security events detection, Cyber-Physical Systems, Internet of Things.**

I. Introduction

Cyber-Physical-Systems (CPS) can provide a broad range of control for complex industrial systems in the Internet of things (IoT) environment through heterogeneous architectures of integrated sensors and devices [1]. CPS systems are expected to be able to perform real-time operations, such as information sensing, processing, communication and actuation by different nodes in the CPS infrastructure. For in-network processing techniques, such as estimation, detection, and tracking in CPS, a compressed sensing based consensus method is introduced for distributed detection, estimation, and tracking, which can guarantee the performance in hash environmental conditions such as random packet losses, asymmetry of the links, etc. [2], [3].

Shancang Li, Shanshan Zhao and Panagiotis Andriotis are with the University of the West of England, Bristol, UK (Email:{shancang.li, shanshan.zhao, panagiotis.andriotis}@uwe.ac.uk).

Po Yang is with the Liverpool John Moores University, Liverpool, UK. (Email: P.Yang@ljmu.ac.uk).

Lida Xu is with the Old Dominion University, Norfolk, VA23529, USA. (Email: lxu@odu.edu).

Qindong Sun is with the Xi'an University of Technology, China. (Email: sqd@xaut.edu.cn).

It is reported that most CPS devices are not adequately designed and more than 47% of all devices in CPS and IoT distrust the security of CPS and IoT [4], [5]. When security in CPS is not sufficient in even seemingly harmless devices or systems, it presents endemic vulnerabilities and risks [4], [6]. As a CPS consists of numerous devices, it is important to develop reliable solutions to ensure that security is built-in against attacks which target connected systems and devices. The consensus is able to improve the security of all devices in a CPS system.

Wireless Sensor Networks (WSNs) are basic CPS components and they have been successfully utilized in events detection and information collection [6], [7], [8], [9], [11], [12]. Compare with WSNs, the CPS can bring several advantages: self-organization, real-time information exchange, collaborative controlling, and reliable data consensus to events status [11]. Through this way, CPS can be run at high efficiency yet in low cost [12]. However, due to the unique characteristics, to implement a CPS system involves a combination of expertise from different professional disciplines [14], [15], [16], [17]: (1) application background knowledge, which is required in CPS services development; (2) smart sensor sensing expertise, which is essential to complete a sensing task; (3) reliable wireless communication, which is required to provide information exchange between nodes or the equipment; and (4) reliable networked data processing expertise, which is needed for understanding the reliable data exchange and processing to provide flexible and scalable networking coverage. The main technical challenges in the CPS include [11], [12], [13], [18]:

1) *Resource constraints*. In CPS, many infrastructure devices are designed with limited processing ability, memory, energy, and communication range. The energy and communication limitations may restrict the coverage and connectivity of the entire system.
2) *Dynamic topologies*. The topology of a CPS changes dynamically over time due to the reconfiguration of network or failure of links or nodes. The nodes might work in both active and inactive mode to save energy, which can also cause the topology to change over time.
3) *Communication burden*. In CPS, too high communication burden will cause high bit error rate (BER) and degrade the performance of the networks. Thus, the goal in communication control is to minimize the communication burden while trying to provide sufficient link bandwidth.
4) *Data consensus in CPS*. Existing centralized schemes significantly rely on specialized routing protocols and

require a central fusion center, making the consensus result unstable for networks with topology changing or node and link failures. Distributed consensus has the advantages of improved robustness, scalability, and efficiency.

In industry, the commonly used methods to deal with the consensus events detection are conducted through the *centralized consensus algorithms* and *distributed consensus algorithms* [10]. In the context of centralized consensus algorithm, fusion centers (FCs) are used to collect all nodes' measurements (i.e., log-likelihood ratio (LLR), etc.) regarding one or more target events and combine them to reach a final decision [19], [20]. The fusion centers are selected depends a number of features required by applications, such as location, capacity, types, etc. In distributed consensus scheme, each node is able to calculate a LLR and transmit it to the FC. For the centralized scheme, the optimization performance depends significantly upon the number of nodes, if local decisions can be correctly received at the FC [8]. If measurements from CPS nodes are not correctively received at the FC due to multihop transmission impairments, the detection performance at FC can be significantly affected [19]. The centralized detection may also cause network congestion when the size of CPS increases. On the other hand, sparse events detection is intimately affected by the changes of topology of CPS. In this paper, sparse events denote events that occur very infrequently in sparse regions, but may cause quite dramatic consequences when they do occur [16], [20], [21], [22]. Eventually, the centralized scheme may increase the energy consumption in CPS and might be unreliable when the hop count increases from node to the FC [7], [16], [17].

Aiming at improving the reliability and reducing the communication burden in events detection through CPS [7], [16], [17], this paper focuses on robust events detection via a distributed consensus algorithm in CPS, in which the active CPS nodes can collaboratively detect events and seek to iteratively reach a global optimum. In the iterative procedure [8], each node is able to exchange the detection results only with its active neighbors within transmission range. Distributed consensus algorithms can be applied to overcome the problems mentioned before [7], [16], [25], in which only local communications between neighboring nodes are involved. Through the iterative updates between neighboring nodes, a consensus global decision can be achieved at all nodes [8], where a distributed consensus algorithm with a fast convergence rate is needed. Furthermore, low information exchanges and transmission is achieved for reliable and energy-effective detection in CPS [8], [9], [16].

Specifically, a distributed consensus algorithm for sparse events detection via a topology-changing CPS is proposed. The monitoring field is represented as a measurement vector, in which each component denotes the detection result at the position that the node lies. Compared with nodes in a CPS, the number of events that might occur is much smaller, which means the *measurement* vector is sparse where only a few elements are non-zeros. This feature enables the *measurements* collection by using compressed sensing (CS) based methods, for which only a few number of random sensory measurements from activate nodes would be enough to accurately reconstruct all *measurements*. The resolution for monitoring can be guaranteed by solving the problem: how to obtain a robust detection result, based on the measurements of both active nodes and sleep nodes.

Furthermore, we address the sparse events detection problem by making the following assumptions: (1) Each node in a CPS works in two switchable modes: active and sleep (inactive) modes. Nodes in active mode can actively probe the environment, and nodes in sleep mode remain idle to save energy and they can easily switch to active mode to perform detection; (2) The number of active nodes is much less than those of the sleep nodes; (3) The number of events that might occur simultaneously is much smaller than the total number of nodes (includes active and sleep nodes) in a CPS; and (4) The received *measurements* are superimposed all together from multiple events when events might occur simultaneously. At the beginning of deployment of a CPS, only a random number of nodes are configured to be in sleep mode. These nodes might be switched into active mode or kept in sleep mode depending on the sleep strategy in topology control which is defined by routing layer. In this paper, we skip the changes of topologies from the issues in routing layer by focusing on improving the reliability of detection and reducing the transmission burden to save energy. First, we propose a jointly connected network model, in which the continuous topology of CPS at different time t can be modeled with jointly connected graphs collection; then the collaborative events detection can be formulated as a consensus optimization problem over the jointly connected networks, which can be solved as a ℓ_1-norm optimization problem [7], [8], [9], [16]. The nodes in active mode can optimize the detection results for both itself and its neighbors in sleep mode. Each active node finally reaches consensus for the sleep node. By this way an event can be accurately detected even when it occurs at the point where the node is in sleep mode. The distributed consensus optimization problem can be solved by alternative direction method and details can be found in Section II.

The rest of the paper is organized as follows: in Section II, a jointly connected network model is presented, and a distributed sparse events detection problem is formulated as a consensus optimization; in Section III, a collaborative consensus algorithm is proposed; experiment simulations are provided in Section IV to evaluate the effectiveness of the proposed algorithm; Section V concludes the paper.

II. Problem Formulation

A. Jointly Connected Networks

In a CPS, consensus means that the detected states of multiple participants converge to the same state value for an event. To address this problem, in this work we define each participant as a node, and in a CPS nodes can communicate with each other in its communicaton range. Consider a CPS with N nodes and a fusion center (FC). The topology can be modeled as a graph with the interconnection links between N nodes, as $\mathcal{G} = \{V, \mathcal{E}\}$, in which $V = \{v_i, i = 1, \ldots, N\}$ is a set of locations of nodes $\mathcal{L} = \{1, \ldots, N\}$, and $\mathcal{E}$ denotes the set of edges of the graph. Let set $\mathcal{N}_a$ consist of all nodes in active mode, and set $\mathcal{N}_s$ include all nodes in sleep mode. Then we have $\mathcal{L} = \{1, \ldots, N\} = \mathcal{N}_a \cup \mathcal{N}_s$ [20], [21], [22].

For node i, if node j lies within its transmission range and $(v_i, v_j) \in \mathcal{E}$ then one can say that node i is a neighbor of node j. All one-hop neighbors of i are contained in a set $\mathcal{N}_i = \{i|(v_i, v_j) \in \mathcal{E}\}$. The FC is denoted by vertex v_0. In this case, a CPS can be represented by a graph $\bar{\mathcal{G}}$ with $\bar{V} = V \cup \{v_0\}$, which includes N nodes and vertex v_0 with directed edges. Note that there may not be any connection between the nodes and the FC at the moment. If one or more direct edges from every node to the FC v_0 can be found, then the graph $\bar{\mathcal{G}}$ is said to be *connected graph* or *sample graph* [25], [26].

In CPS, a union of connected graphs $\{\bar{\mathcal{G}_1}, \bar{\mathcal{G}_2}, \ldots, \bar{\mathcal{G}_m}\}(m \geq 1)$ that have the common vertex set $\bar{V}$ are defined as a union of simple graphs [26]. For simplicity, the union of simple graphs are denoted as $\bar{\mathcal{G}}_{1,\ldots,m}$. It is clear that $\bar{\mathcal{G}}_{1,\ldots,m}$ includes a vertex set $\bar{V}$ and the edges of $\bar{\mathcal{G}}_{1,\ldots,m}$ is the union of edges of all simple graphs. To properly describe the union of simple graphs, a new concept is introduced as "*jointly connected graphs*" $\{\bar{\mathcal{G}}_1, \ldots, \bar{\mathcal{G}}_m\}$, in which each simple graph $\bar{\mathcal{G}}_i$ is a connected graph with a common vertex set $\bar{V}$ and $\mathcal{E}_i$ might be different. It can be understood that a collection of *jointly connected graphs* contains at least one simple connected graph [23], [24], [26].

Theorem 1. *A collection of graphs $\{\bar{\mathcal{G}_1}, \bar{\mathcal{G}_2}, \ldots, \bar{\mathcal{G}_m}\}$ can be said jointly connected if its union graphs $\bar{\mathcal{G}}_{1,\ldots,m}$ are connected.*

Since each graph in the collection contains v_0, it guarantees that each graph contains at least one common node [26], [27]. It should be noticed that if one or more graphs are connected in this collection, then it is jointly connected. At a time interval $[t, \tau]$, if n nodes are connected and formed a collection of simple graphs $\{\bar{\mathcal{G}}_t, \bar{\mathcal{G}}_{t+1}, \ldots, \bar{\mathcal{G}}_\tau\}$, then the graphs with different topologies are said to be jointly connected.

Assume a node i is able to make a local decision to determine the occurrence of an event at a point v_i, which is superimposed of reading from its neighboring points, and can be modeled as

$$x_i(t+1) = x_i(t) + \sum_{j \in \mathcal{N}_i(t)} w_{ij} x_j(t), i \in \mathcal{W} \tag{1}$$

in which $x_i(t)$ is LLR at node i at time t, and w_{ij} is the weight between i and j.

For a graph that contains a number of nodes, the links between nodes change over time and that causes change of the topologies of the graph [28]. Let $\mathcal{P}$ represent a node-set, in which all simple graphs $\bar{\mathcal{G}}_p(p \in \mathcal{P})$ defined on $\bar{V}$ are well indexed.

The set of LLRs can be easily defined in a state vector form. For each $p \in \mathcal{P}$, define

$$\mathbf{x}(t+1) = (A_\sigma + I)\mathbf{x}(t) = F_\sigma \mathbf{x}(t) \tag{2}$$

in which $\mathbf{x}$ denotes a vector of LLRs: $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$ and $\sigma : \{0, 1, \ldots\} \to \mathcal{P}$ is used to represent a scheduling signal that reconfigures the networks and hence causes changes of topology at a specific time t (including CPS reconfiguration, nodes or links failures, *etc.*), $F_\sigma = (A_\sigma + I)$, and $A_p(p \in \mathcal{P})$ denotes the adjacent matrix of graph $\bar{\mathcal{G}}_p$.

In this model, σ changes as a function of locations of the active nodes in a CPS. Actually, the convergence analysis of this model is a very difficult task [29], [30]. We ignore the dependencies between σ and the node positions and instead of that σ can be any scheduling signal that is properly predefined. By doing this, the convergence difficulty can be avoided. The main goal is to obtain a stable global optimal decision from all n nodes for any initial set of node local decision, which is expected to converge to a stable state value $\mathbf{x}_S$. The convergence problem of x_i to $\mathbf{x}_S$ equals to solving the convergence problem of $\mathbf{x}_S\mathbf{1}$.

For a very small $\mathcal{P}$, σ remains constant make sure $\mathcal{G}$ is a complete graph in $p \in \mathcal{P}$. In this case, $\mathbf{x}$ can easily converge to $\mathbf{x}_S\mathbf{1}$. Let $\mathcal{Q}$ denote a subset of $\mathcal{P}$ that consists of the indices of the connected simple graphs in collection $\{\bar{\mathcal{G}}_p, p \in \mathcal{P}\}$ [31], [32].

Theorem 2. *For a scheduling signal $\sigma : \{0, 1, 2, ...\} \to \mathcal{P}$, if $\mathbf{x}(0)$ is given and for all $t \in \{0, 1, \ldots\}$, $\sigma(t) \in \mathcal{Q}$ holds, then*

$$\lim_{t \to \infty} \mathbf{x}(t) = \mathbf{x}_S\mathbf{1} \tag{3}$$

It is possible that a jointly connected collection of simple graphs converges to a common decision, which has a less strength than that in Theorem 2. Meanwhile, Theorem 2 requires the collection $\{\bar{\mathcal{G}}_{\sigma(t_i)}, \bar{\mathcal{G}}_{\sigma(t_i+1)}, \ldots, \bar{\mathcal{G}}_{\sigma(t_{i+1}-1)}\}$ in $[t_i, \tau)$ to be jointly connected. For a scheduling signal σ, Eq.(3) holds if an infinite and non-overlapping sequence of intervals is available across which the collection is jointly connected. It should be noted that during interval from t_i to t_{i+1}, at least one component in $\mathcal{Q}$ is picked up as switching signal.

Proof. As mentioned above, each F_p is non-negative, and all the sums of each row of each F_p are equal to 1 (i.e., $F_p\mathbf{1} = \mathbf{1}$). So the matrix F_p is stochastic and its diagonal elements are all non-zeros. For a simple graph $\bar{\mathcal{G}}_p(p \in \mathcal{Q})$, if m is sufficiently large then all entries in $(I + A_p)^m$ are positive. Hence both $(I + A_p)^m$ and F_p are *primitive matrices*, which means that the largest eigenvalue of F_p for $p \in \mathcal{Q}$ is 1, and all remaining eigenvalues must lie in $(-1, 1)$. Then we have

$$\lim_{t \to \infty} F_p^t = \mathbf{1}c_p \tag{4}$$

for some row vector c_p.

It is clear that all diagonal elements of a stochastic matrix $F_p(p \in \mathcal{Q})$ are positive, and they are primitive. □

Theorem 3. *For a finite set of ergodic matrices $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_m\}$, then for a stochastic matrix $\mathcal{F}_p \in \mathcal{M}$, $(\mathcal{F}_p)^i (i \to \infty)$ is a matrix of rank 1.*

For simplicity, a connected graph can be used to denote a network, and a collection of jointly connected graphs can be used to represent a set of topologies of networks, where the active nodes may be different for different topologies.

Lemma 1. *For a set of jointly connected networks, let $\{\bar{\mathcal{G}}_{p_1}, \bar{\mathcal{G}}_{p_2}, \ldots, \bar{\mathcal{G}}_{p_m}\}(\{p_1, p_2, \ldots, p_m\} \in \mathcal{P})$ denote the* *corresponding topologies, then the product of matrices $F_{p_1}F_{p_2}\cdots F_{p_m}$ is said ergodic.*

Theorem 3 can be proved as follows:

Proof. For $t \geq 0$, let $\Phi(t,t) = I$ hold and τ be an integer between 0 and t, then we have $\Phi(t,\tau) = F_{\sigma(t+1)} \cdots F_{\sigma(\tau-1)} \cdot F_{\sigma(\tau)}$. Accordingly, a matrix $\theta(t)$ can be reformatted as $\theta(t) = \Phi(t,0)\theta(0)$. Actually Eq.(5) is enough to prove Theorem 2

$$\lim_{t\to\infty} \Phi(t,0) = \mathbf{1}c \tag{5}$$

in which $\mathbf{c}$ denotes a row vector. According to Lemma 1, for an integer $j \geq 0$, matrix $\Phi(t_{j+1}, t_j)$ is said ergodic, and it can be represented by a product of finite matrices from $\{F_p, p \in \mathcal{P}\}$. Accordingly, $\Phi(t_j, 0)$ can be substituted by $\Phi(t_j, t_{j-1})\Phi(t_{j-1}, t_{j-2}) \cdots \Phi(t_1, t_0)$. Therefore, $\Phi(t_j, 0)$ is ergodic and we have

$$\lim_{j\to\infty} \Phi(t_j,0) = \mathbf{1}c \tag{6}$$

Let j_t denote the largest non-negative integer that satisfies $t_{j_t} \leq t$. Accordingly, $\Phi(t,0)$ can be represented by the product of $\Phi(t, t_{j_t})$ and $\Phi(t_{j_t}, 0)$, and we have

$$\Phi(t,0) - \mathbf{1}c = \Phi(t, t_{j_t}) \cdot (\Phi(t_{j_t}, 0) - \mathbf{1}c) \tag{7}$$

It can be seen that $\Phi(t_{j_t}, 0) \to 0$ when $t \to \infty$ due to Eqs (5) and (6), therefore Eq.(4) holds and the proof is complete. □

For two non-negative matrices $\mathcal{F}_i$ and $\mathcal{F}_j$, if all elements of $\mathcal{F}_i - \mathcal{F}_j$ are non-negative, then matrix $\mathcal{F}_i - \mathcal{F}_j$ is said a non-negative matrix.

Proof. (*Lemma 1*) Let a non-negative matrix $F = (I + A)$, in which A denotes the adjacency matrix of the collection being jointly connected graphs $\{\bar{\mathcal{G}}_{p_1}, \bar{\mathcal{G}}_{p_2}, \ldots, \bar{\mathcal{G}}_{p_m}\}$. Then matrix F is said primitive. According to Lemma 2 we have

$$F_{p_1} \cdot F_{p_2} \cdot \ldots \cdot F_{p_m} \geq \xi(F_{p_1} + F_{p_2} + \cdots + F_{p_m}) \tag{8}$$

in which ξ denotes a small positive constant. Then for a primitive matrix F_{p_i}, $F_{p_i} \geq (I + A_{p_i})$ holds, and

$$F_{p_1} \cdot F_{p_2} \cdot \ldots \cdot F_{p_m} \geq \xi(mI + A_{p_1} + A_{p_2} + \cdots + A_{p_m}) \tag{9}$$

Since m is an integer then $mI \geq I$ holds and Eq.(9) is reduced to

$$F_{p_1} \cdot F_{p_2} \cdot \ldots \cdot F_{p_m} \geq \xi F \tag{10}$$

It should be noticed that the product is bounded below by ξF, and the product is primitive as well. As mentioned in [16], [17] the product is also a stochastic matrix, so it is ergodic [26], [33]. □

Lemma 2. *For an $m \times m$ non-negative set $A_i, i \in \{1, 2, \cdots, m\}$, let μ denote the smallest diagonal element of A_i and ρ denote the largest diagonal elements of A_i. We have*

$$A_1 A_2 \cdots A_m \geq (\frac{\mu^2}{2\rho})^{(m-1)}(A_1 + A_2 + \cdots + A_m) \tag{11}$$

Proof. This can be easily proved by writing A_i as $A_i = \mu I + B_i$, where B_i is non-negative. For any j, k

$$A_j A_k = (\mu I + B_i)(\mu I + B_i) \geq \mu^2 + \frac{\mu^2}{2\rho}(B_j + B_k) \tag{12}$$

Since $(\rho I + B_j) \geq A_j$ and $(\rho I + B_k) \geq A_k$, then we have

$$A_j A_k \geq (\frac{\mu^2}{2\rho})(A_j + A_k) \tag{13}$$

Using Eq.(12) iteratively, Eq.(11) holds and the proof is complete. □

B. Jointly Connected Graphs based Consensus Algorithm

For a collection of jointly connected networks, each node is capable of exchanging information with its active neighbors directly and keeping all the local LLRs vector in the network to derive a weighted average, which eventually converges to a global decision vector [34], [35].

As proved in Theorem 2, for $\mathbf{x}(t+1) = F_\sigma \mathbf{x}(t)$, weight matrix F_σ features the sparsity pattern specified by the jointly connected graphs collection $\{\bar{\mathcal{G}_1}, \bar{\mathcal{G}_2}, \ldots, \bar{\mathcal{G}_m}\}$ and $\sigma : \{0, 1, \ldots\} \to \mathcal{P}$, in which weight matrix F_σ corresponding to the edges of connected graph. For a t-step transition matrix $\Phi(t) = F_{p_1}F_{p_2}\cdots F_{p_m}$, we have

$$\mathbf{x}(t+1) = \Phi(t)\mathbf{x}(0) \tag{14}$$

and according to Theorem 2, we have

$$\lim_{t\to\infty} \Phi(t) = \frac{1}{n}\mathbf{1}\mathbf{1}^T \tag{15}$$

which is equivalent to

$$\lim_{t\to\infty} \mathbf{x}(t) = \Big(\frac{1}{n}\mathbf{1}^T\mathbf{x}(0)\Big)\mathbf{1} \tag{16}$$

The weight matrix satisfies the condition in Eq.(16). In practice, some links might fail permanently, however the jointly connected scheme guarantees the long-term connectivity of graphs [36], [37].

III. Distributed Sparse Events Detection

When an event occurs around a local node v_j, it might influence its neighboring area by a non-zero influence function $w_j(v_j)$, which can be normalized to obey $\sum w_j(v_j) = 1$ [33]. Let y_i denote the measurement at point v_i that is the superposition of the influence of all events on v_i. LLR will be obtained at v_j, and we have $w_{ji} = w_{ij}$, which is the weight event at v_i. Let ϵ_i denote the measurement noise of zero mean. It is easy to understand that the LLRs vector $\mathbf{x}$ is sparse, but the measurement vector $\mathbf{y}_i$ can be non-sparse.

For a CPS with $\mathcal{N}_a(t)$ active nodes and $\mathcal{N}_s(t)$ inactive nodes at time t, measurement y_i at v_i can be modeled as

$$y_i = \sum_{j\in N} A_{j,i}x_j + \epsilon_i \tag{17}$$

in which $A_{i,j} = A_{j,i}$ denotes the influence event at v_i.

For node i, if node j is out of the transmission range of i, then let $A_{ji} = 0$. Accordingly, the observation y_i can be represented as $y_i = x_i + \sum_{j\in n} A_{j,i}x_j + \epsilon_i$. Furthermore, for a network with $\mathcal{N}_a$ active nodes, we have

$$\mathbf{y_a} = \mathbf{\Phi}\mathbf{A}\mathbf{x} + \epsilon_\mathbf{a} \tag{18}$$

in which $\mathbf{\Phi}$ denotes the selection matrix, $\mathbf{y_a}$ denotes the measurement vector and $\epsilon_\mathbf{a}$ denotes the noise, respectively.

Here compressed sensing can be used to perfectly recover $\mathbf{x}$ from measurements $\mathbf{y_a}$ [7], [8]

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b_a}\|_2^2 + \lambda\|\mathbf{x}\|_1 \quad s.t. \quad \mathbf{x} \geq \mathbf{0} \tag{19}$$

The neighbors list of node i is denoted by $\mathcal{N}_i$. Node i not only keeps x_i at v_i, but also keeps measurements x_k ($\forall k \in \mathcal{N}_a \cap \mathcal{N}_i$) that is evaluated at its inactive neighboring nodes. Actually, neighbors of i include active nodes belong to $\mathcal{N}_i \cap \mathcal{N}_a$ and the sleep nodes belonging to $\mathcal{N}_i \cap \mathcal{N}_s$. Then we rewrite Eq.(19) as

$$\begin{aligned} \min \sum_{i\in\mathcal{N}_a} \Big(y_i - x_i^{(i)} - \sum_{k\in\mathcal{N}_s\cap\mathcal{N}_i} A_{k,i}x_k^{(i)} - \sum_{j\in\mathcal{N}_a} A_{k,i}x_j^{(j)}\Big)^2 & \\ + \lambda\|\mathbf{x}\|_1 & \quad (20a) \\ s.t.\ x_i^{(i)} \geq 0, \forall i \in \mathcal{N}_a, & \quad (20b) \\ x_k^{(i)} \geq 0, \forall k \in \mathcal{N}_s \cap \mathcal{N}_i & \quad (20c) \end{aligned}$$

Note that here $\|\mathbf{x}\|_\mathbf{1}$ can be solved by

$$\|\mathbf{x}\|_1 = \sum_i x_i^{(i)} + \sum_{i\in N_s\cup\mathcal{N}_k} x_k^{(i)} \tag{21}$$

in which $\mathcal{N}_k$ denotes the active neighboring nodes at v_k and both $x_i^{(i)}$ and $x_k^{(i)}$ are non-negative constraints for all decision variables.

A. Collaborative Consensus Optimization

It is crucial to perform collaborative detection by fusing $\mathbf{y}$ to obtain a global optimal estimation of the sparse decision $\mathbf{x}$. For a CPS with $N = |\mathcal{N}_a| + |\mathcal{N}_s|$ nodes and a FC, $\mathbf{x}$ can be reconstructed from the following ℓ_1-norm optimization formulation

$$\begin{aligned} \min \sum_{i\in\mathcal{N}_a} a_i^2 + \sum_i x_i^{(i)} + \sum_{i\in N_s\cap\mathcal{N}_k} x_k^{(i)} & \quad (22a) \\ s.t.\ a_i = y_i - x_i^{(i)} - \sum_{k\in\mathcal{N}_s\cup\mathcal{N}_i} A_{k,i}x_k^{(i)} - \sum_{j\in\mathcal{N}_a} A_{k,i}x_j^{(j)} & \\ & \quad (22b) \\ x_i^{(i)} \geq 0, \forall i \in \mathcal{N}_a, & \quad (22c) \\ x_k^{(i)} \geq 0, \forall k \in \mathcal{N}_s \cap \mathcal{N}_i & \quad (22d) \end{aligned}$$

Eq.(22) might yield a globally optimal result, in which the linear measurements from all the nodes are centrally fused at the FC. It is costly but easy to be implemented. Not only all measurements, but also the measurement matrices of all nodes need to be collected at FC.

Let J denote the number of active nodes, then the problem reduces to J least-squares sub-optimal functions. This may cause a very expensive computation cost when the number of nodes increases.

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 + \sum_{j=1}^{J} \lambda_j\|\mathbf{y}^{(j)} - A\mathbf{x}^{(j)}\|_2^2 \tag{23}$$

where the positive parameter $\{\lambda_i\}$ describes the noise resilience of the samples $\{x_t^{(j)}\}$ and $\lambda = \sum_j \lambda_i$ describes the trade-off between noise resilience and events sparsity. As mentioned above, the centralized fusion may cause high communication burden and hence cause unstable optimal results and high energy consumption.

A distributed consensus algorithm may overcome the drawbacks of centralized consensus fusion, which uses only local optimal between one-hop neighbors to iteratively estimate the decision. In this case, each active node j keeps a local copy of the local decision $x_j^{(j)}$ and collaboratively consent on their copies [16], [35]. Let $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_m\}$ be a collection of jointly connected networks depicting the connectivity of the CPS, in which each network $\mathcal{G}_i = (V_i, \mathcal{E}_i)$ includes active nodes for the set of vertices $V_i (i = 1, \ldots, m)$ and each edge $(j, k) \in \mathcal{E}_i$ connects an unordered pair of distinct nodes within one-hop neighborhood. Different from the centralized fusion formula, each node j locally performs the following consensus optimization

$$\min_{\mathbf{x}^{(j)}} \|\mathbf{x}^{(j)}\|_1 + \lambda_j\|\mathbf{x}^{(j)} - A\mathbf{x}^{(j)}\|_2^2 \tag{24}$$

It can be seen that Eq.(24) enforces the consensus between j and its one-hop active neighbors, which can be solved as a LASSO problem [17], [34]. Each local LLR at an active node is shared with its one-hop neighbors and will be updated and percolated throughout the network after performing an iterative consensus procedure that converges to an optimal result. Upon convergence, all neighboring nodes will have consented on the same globally optimal $\mathbf{x}$. It can be seen that when $\sum_j \sum_k w_{jk} = 1$, Eq.(24) forces the LLR copy x_j to consent to a weighted neighboring LLR by a reminiscent phase. Thus, the weighting matrix $\mathbf{A}$ can be easily obtained by setting its (i, k)-th element as w_{jk} by adhering to $\mathbf{A}\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ is the all-one vector.

B. Distributed Consensus Implementation via the Alternating Direction Method of Multipliers

The iterative optimization can be implemented via the global consensus discussed above. For Eq.(24), an augmented Lagrangian function can be created as

$$\begin{aligned} \mathcal{L}\Big(\mathbf{x}^{(j)}, \lambda_j, \mathbf{z}_j, \{\mathbf{x}^{(j)}\}\Big) = \|\mathbf{y}^{(j)} - A^{(j)}\mathbf{x}^{(j)}\|_2^2 + \mathbf{z}_j^T\mathbf{x}^{(j)} & \\ + \frac{\beta}{2}\|\mathbf{x}^{(j)} - \sum_{k\in\mathcal{N}_j} w_{jk}\mathbf{x}^{(k)}\|_2^2 & \quad (25) \end{aligned}$$

where $\mathbf{z}_j$ denotes a Largrangian operator, and β denotes the augmented Lagrangian multipliers in the consensus optimization constraints. $\mathbf{z}_j^T\mathbf{x}^{(j)}$ is applied to guarantee that Eq.(25) is fulfilled by properly setting $\mathbf{z}^T(\mathbf{x}^{(j)} - \sum_{k\in\mathcal{N}_j} w_{jk}\mathbf{x}^{(k)}) = 0$. By using the alternating direction of multipliers, each node can update $\mathbf{x}^{(j)}(t)$ by iteratively solving

$$\mathbf{x}^{(j)} = arg\min_{\mathbf{x}^{(j)}} \mathcal{L}\Big(\mathbf{x}^{(j)}, \lambda_j, \mathbf{z}_j, \{\mathbf{x}^{(j)}\}\Big) \tag{26}$$

in which the multipliers can be updated via a gradient-based iteration

$$\mathbf{z}^{(j)}(t) = \mathbf{z}^{(j)}(t-1) + \frac{\beta}{2}\Big(\mathbf{x}^{(j)} - \sum_{k\in\mathcal{N}_j} w_{jk}\mathbf{x}^{(k)}\Big) \tag{27}$$

Algorithm 1: Distributed Fusion Algorithm

Input: Each node calculates LLR as $\mathbf{x}_t^{(j)}$, sets λ_j and c empirically, initializes estimate $\mathbf{x}^{(j)}(0) = 0$ and local multiplier vector $\mathbf{z}^{(j)}(0) = 0$. Let T be the maximum number of iterations and e be the tolerable deviation, both are at convergence conditions.

Output: Algorithm converges to an optimal result and each node obtains the global estimation $\mathbf{x} = \mathbf{x}^{(j)}(T), \forall j \in \mathcal{N}$.

repeat

  All nodes update $\mathbf{z}^{(j)}(t)$ and $\mathbf{x}_t^{(j)}$ via Eqs (26) and (27), $\forall j$;

  All nodes transmit $\mathbf{x}_t^{(j)}(t+1)$ to their one-hop neighbors in $\mathcal{N}_j, \forall j$;

  $t \leftarrow t + 1$.

until $t > T$ *or* $\|\mathbf{x}_t^{(j)}(t+1) - \mathbf{x}_t^{(j)}(t)\| \leq e$;

In Eqs (26) and (27), the iterative steps constitute a distributed scheme, and details can be found in Algorithm 1.

In the t-th iteration, a node j first collects $\mathbf{x}^{(j)}$ from its one-hop neighbors $k \in \mathcal{N}_i$, which is then used to update the local multiplier vector $\mathbf{z}^{(j)}(t)$ as described in Eq.(27). By doing this, Eq.(26) can be solved as a quadratic optimization problem and the updated local LLR estimation $\mathbf{x}^{(j)}(t+1)$ will be yielded. Then, all nodes locally update the decision and exchange the sparse estimation $\mathbf{x}^{(j)}(t+1)$ with their one-hop neighbors. This procedure repeats until converged to the specific condition.

During iterations, nodes are not required to synchronize the measurements, which makes it easily implemented in a large-scale CPS. The convergence of Algorithm 1 can be proved as follows:

Proof. In [17], the iterative alternating direction method have been proved to converge to a minimizer of Eq.(26) for any positive constant β. □

IV. Performance Evaluation

In this section, a CPS will be created to perform the sparse events detection using proposed distributed consensus algorithm. Considering an event detection scheme in a small network with 5 nodes ($n_i, i = 1, \ldots, 5$), which are deployed in a two-dimensional area and the distance between two neighboring nodes is 20, and communication range is 50 as shown in Fig.1.

It can be seen that the neighbor-set of n_1 includes nodes $\{n_2, n_3\}$. Similarly, n_2 has a neighbor-set as $\{n_1, n_3, n_4\}$, n_3 has a neighbor-set as $\{n_1, n_2, n_4, n_5\}$, n_4 has a neighbor-set as $\{n_2, n_3, n_5\}$, and n_5 has a neighbor-set as $\{n_3, n_4\}$, respectively. As mentioned in Section III, each node does not only hold its local decision, but also holds the decisions of its neighbors (both active and sleep neighbors). Let c_i denote the possibility that event might occur at the position that n_i lies (v_i), and all five nodes are in active mode. Assume that three events occurred at v_1, v_3, and v_5, respectively. The detection results from all five nodes are reported in Table I, where the possibilities of events occurred are reported and N/A means that the detection is not available. For example, node n_1 can successfully detect events occurred at n_1, n_2, n_3, but cannot detect events occurred at n_4 and n_5. It is reasonable that nodes n_1, n_2, and n_3 are in the neighbors list of n_1.

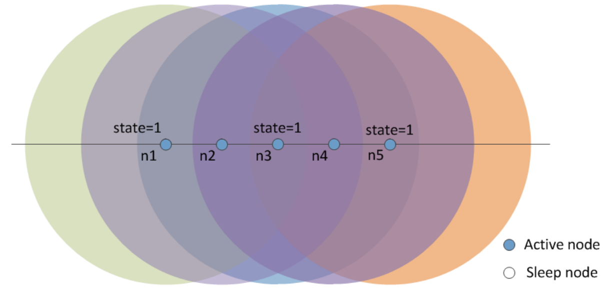


Fig. 1. Consensus algorithm on 5 active nodes

TABLE I
Detection results after 200 iterations

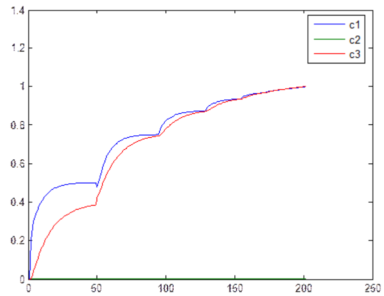
	c_1	c_2	c_3	c_4	c_5
n_1	1.000	0.000	0.9987	N/A	N/A
n_2	1.000	0.000	1.000	0.000	N/A
n_3	1.000	0.000	1.000	0.000	0.9918
n_4	N/A	0.000	1.000	0.000	1.000
n_5	N/A	N/A	0.9991	0.000	1.000

Fig.2 presents the optimization results when all nodes are active *vs* the number of iterations. In practice, in order to save energy not all nodes are in active mode. Assume the distance between these nodes is 20 and the transmission range is 25. Let nodes n_2 and n_4 be in sleep mode, then the consensus algorithm can only be performed on nodes n_1, n_2, and n_3, and can report the decision results. However, as discussed above, the active nodes are able to report the decision result by itself. Meanwhile its neighboring active nodes are able to report the decision results of its neighboring inactive nodes. In Fig.3, n_1 has an inactive neighboring node n_2, node n_3 has two inactive neighboring nodes $\{n_2, n_4\}$, and n_5 has an inactive neighbor n_4, respectively.

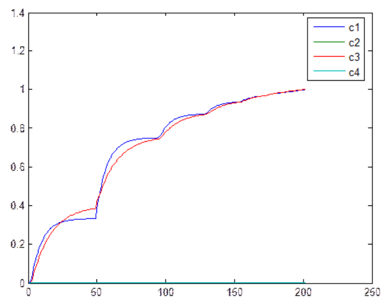
Table II depicts the detection results of CPS with 3 active nodes and 2 inactive nodes, where each node is able to keep its local decision and decisions of its one-hop neighbors. It can be seen that the decision results converge to 0.9967 for c_1, 0.9989 for c_3, and 0.9971 for c_5, respectively. The decisions near to 1 converged faster than the above scenario. It is due to the fact that each active node holds a small number of neighbors. Fig.4 shows the optimization results at node n_1, n_3, and n_5, when events c_1, c_3, and c_5 occurred at n_1 and n_3, and n_5. Nodes n_2 and n_4 are inactive.

TABLE II
Detection results after 50 iterations

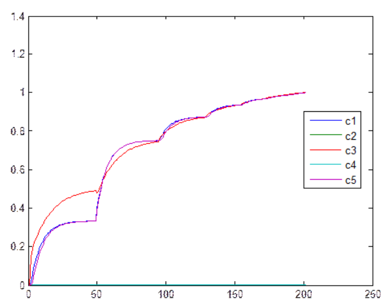
	c_1	c_2	c_3	c_4	c_5
n_1	0.9967	0.001	N/A	N/A	N/A
n_3	N/A	0.000	0.9989	0.000	N/A
n_5	N/A	N/A	N/A	0.000	0.9971



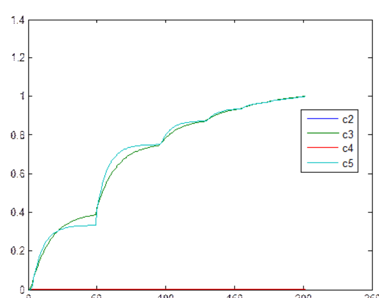
(a) Consensus result at n_1 vs Iteration number



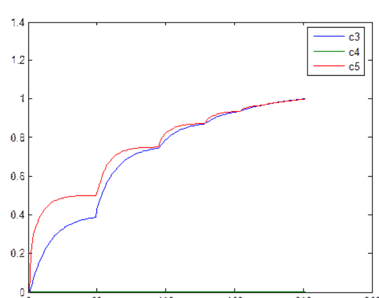
(b) Consensus result at n_2 vs Iteration number



(c) Consensus result at n_3 vs Iteration number



(d) Consensus result at n_4 vs Iteration number



(e) Consensus result at n_5 vs Iteration number

Fig. 2. Consensus algorithm on 5 active nodes

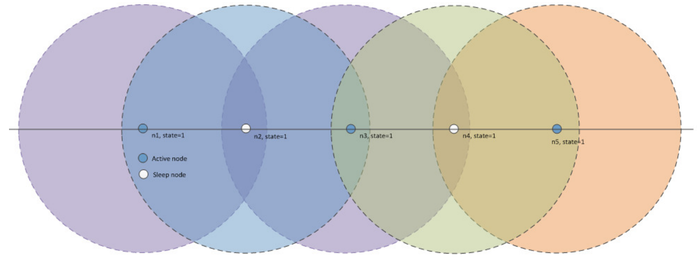
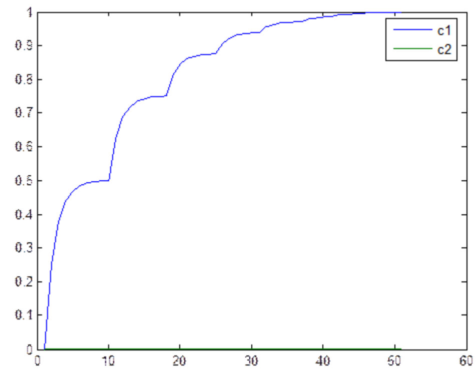
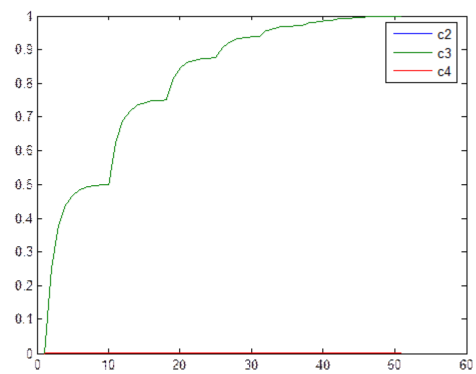


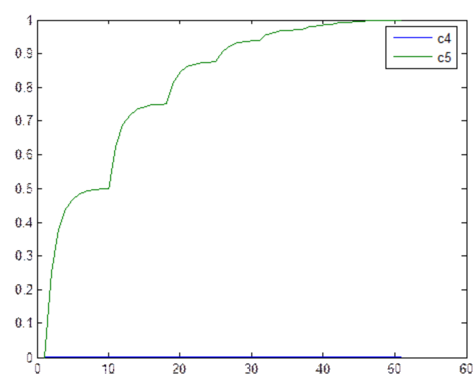
Fig. 3. Consensus algorithm on 3 active nodes and 2 inactive nodes



(a) Consensus result at n_1 vs Iteration number



(b) Consensus result at n_3 vs Iteration number



(c) Consensus result at n_5 vs Iteration number

Fig. 4. Consensus algorithm on 3 active nodes vs Iteration number

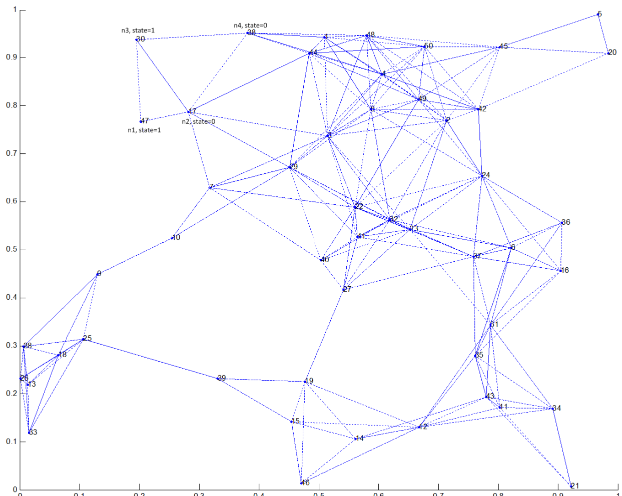


Fig. 5. Consensus algorithm on 3 active nodes

In order to evaluate the performance of sparse events detection in a large CPS, we create a network with 50 nodes. Consider 50 nodes are randomly deployed in a normalized square, as shown in Fig.5. Nodes are connected with its neighbors if its neighbors are within the transmission range. If the graph (network) is not connected, the locations of nodes are re-generated randomly until the graph is connected. In Fig.5, four active nodes n_1, n_2, n_3, and n_4 are located at points v_{47}, v_{17}, v_{30}, and n_{38}, respectively. Assume events c_1 and c_2 occur at v_{47} and v_{30}, respectively. Fig.6 shows the detection results at active nodes n_1, n_2, n_3, and n_4. At node n_1, the events c_1 and c_2 are successfully detected as shown in Fig.6(a). In Fig.6(b) n_2 successfully detected the occurrence of c_1, c_2, and c_3 since node n_1 and n_2 are its neighbors. Since the status of c_3 and c_4 are 0, which means no event occurred at v_3 and v_4, however nodes n_3 and n_4 successfully reported its neighbors' detection results, as shown in Fig.6(c) and Fig.6(d).

In this work, the following normalized mean squared error (MNSE) is used to evaluate the performance of decision making

$$\mathrm{NMSE} = \frac{\mathbb{E}[||\hat{\mathbf{x}} - \mathbf{x}||^2]}{\mathbb{E}[||\mathbf{x}||^2]} \tag{28}$$

Fig.7 shows the NMSE of decision results made by node n_1 in Fig.5 (the NMSE values obtained by simulations of 100 runs), where N_1 denotes the MNSE results when c_1 is made, N_2 denotes the MNSE results when c_2 is made, and N_3 denotes the MNSE results when $c3$ is made, respectively.

V. Discussion

This paper proposes a simple but compelling jointly connected graph model for topology-changing CPS, which can significantly improve the reliability of distributed consensus decision-making. Although several decentralized consensus schemes have been reported in [19], [37], [20], however most of them are developed for decentralized in-networking consensus without supporting the changes of topologies of WSNs. This paper uses the idea of consensus optimization and compressed sensing to develop a distributed consensus events detection scheme by exploiting sparsity, which has the following key differentiating features:

(1) This scheme can still perform consensus optimization when the topology dynamically changes, which avoids the reconfiguration of CPS caused by the switch of node work model (between active and sleep modes). This feature significantly increases the reliability of CPS and decreases the power consumption in network reconfiguration and re-organization.

(2) Different from the existing distributed consensus scheme reported in [19], [37], [20], where each node holds a local decision vector of the whole network, it is too costly for each node in a large network. In our scheme, each active node holds an LLR vector for itself and its inactive neighbors. This significantly reduces the computation and communication costs per node, and improves the reliability and scalability of the algorithm for a CPS with a large number of nodes.

(3) The proposed scheme exploits compressed sensing to recover LLRs by distributive joint sparsity local estimations. This can improve the reliability of consensus optimization by tolerating the failure of nodes or links without causing the reconfiguration or rebuilding of CPS.

VI. Conclusion

This paper proposed an iterative distributed consensus algorithm that can be employed for sparse events detection in CPS, which was originally derived for achieving consensus by providing each node the full detection information even when the topologies change. Each node is able to iteratively calculate the LLR of itself and LLRs of neighbors by communicating with the neighboring active nodes. Using the notion of compressed sensing, the estimation of decision can be converged faster. The numerical results showed that the estimation of sparse events detection can be successful calculated with the proposed distributed consensus algorithm.

References

[1] Danna B. Rawat, Joel J.P.C. Rodrigues, Ivan Stojmenovic, Cyber-Physical Systems: From Theory to Practice, *CRC Press*, 2015.

[2] Houbing Song, Glenn A. Fink, and Sabina Jeschke, Security and Privacy in Cyber-Physical Systems: Foundations, Principles and Applicaations. *UK: Wiley - IEEE Press*, 2017.

[3] Kostas Kolomvatsos; Christos Anagnostopoulos; Stathes Hadjiefthymiades, Distributed Localized Contextual Event Reasoning under Uncertainty, *IEEE Internet of Things Journal*, vol.pp, no.99, pp.1-1, May 2016.

[4] Bhavya Kailkhura; Swastik Brahma; Pramod Varshney, Data Falsification Attacks on Consensus-based Detection Systems, *IEEE Transactions on Signal and Information Processing over Networks*, vol.10, no.2, pp.1461-1468, May 2016.

[5] Bhavya Kailkhura; Swastik Brahma; Pramod Varshney, Data Falsification Attacks on Consensus-based Detection Systems, *IEEE Transactions on Signal and Information Processing over Networks*, vol.3, no.1, pp.145-158, Sep 2016.

[6] Shancang Li, George Oikonomou, Theo Tryfonas, Tom Chen, and Lida Xu, "A Distributed Consensus Algorithm for Decision Making in Service-Oriented Internet of Things", *IEEE Transactions on Industrial Informatics*, vol.10, no.2, pp.1461-1468, May 2014.

[7] Jia Meng, Husheng Li, and Zhu Han, "Sparse Event Detection in Wireless Sensor Networks using Compressive Sensing", *the 43rd Annual Conference on Information Sciences and Systems (CISS)*, 2009.

[8] Mohammed Moness; Ahmed Mahmoud Moustafa, "A Survey of Cyber-Physical Advances and Challenges of Wind Energy Conversion Systems: Prospects for Internet of Energy", *IEEE Internet of Things Journal*, vol.3, no.2, pp.134-145, 2016.

[9] Leinonen, M.; Codreanu, M.; Juntti, M., "Sequential Compressed Sensing With Progressive Signal Reconstruction in Wireless Sensor Networks", *IEEE Transactions on Wireless Communications*, vol.14, no.3, pp.1622-1635, 2015.

[10] Adnan Aijaz; A. Hamid Aghvami, Cognitive Machine-to-Machine Communications for Internet-of-Things: A Protocol Stack Perspective, *IEEE Internet of Things Journal*, vol.2, no.2, pp.103-112, 2015.

[11] Yujiao Zheng; Nianxia Cao; Wimalajeewa, T.; Varshney, P.K., "Compressive Sensing Based Probabilistic Sensor Management for Target Tracking in Wireless Sensor Networks", *IEEE Transactions on Signal Processing*, vol.63, no.22, pp.6049-6060, 2015.

[12] Dautov, R.; Tsouri, G.R., "Securing While Sampling in Wireless Body Area Networks With Application to Electrocardiography", *IEEE Journal of Biomedical and Health Informatics,* , vol.20, no.1, pp.135-142, 2016.

[13] Liqun Hou; Bergmann, N.W., "Novel Industrial Wireless Sensor Networks for Machine Condition Monitoring and Fault Diagnosis", *IEEE Transactions on Instrumentation and Measurement*, vol.61, no.10, pp.2787-2798 , 2012.

[14] Songcen Xu; de Lamare, R.C.; Poor, H.V., "Distributed Compressed Estimation Based on Compressive Sensing", *IEEE Signal Processing Letters*, vol.22, no.9, pp.1311-1315, 2015.

[15] Bertocco, M.; Gamba, G.; Sona, A.; Vitturi, S., "Experimental Characterization of Wireless Sensor Networks for Industrial Applications", *IEEE Transactions on Instrumentation and Measurement*, vol.57, no.8, pp.1537-1546, 2008.

[16] Qing Ling and Zhi Tian, "Decentralized Sparse Signal Recovery for Compressive Sleeping Wireless Sensor Networks", *IEEE Transactions on Signal Processing*, vol.58, no.7, pp.3816-3828, July 2010.

[17] Fanzi Zeng, Chen Li, and Zhi Tian, "Distributed Compressive Spectrum Sensing in Cooperative Multihop Cognitive Networks", *IEEE Transactions on Selected Topics in Signal Processing*, vol.5, no.1, pp.37-48, February 2011.

[18] Fatemeh Fazel, Maryam Fazel and Milica Stojanovic, "Random Access Compressed Sensing for Energy-Efficient Underwater Sensor Networks", *IEEE Journal of Selected Areas in Communications*, vol.29, no.8, pp.1660-1670, September 2011.

[19] Silvana Silva Pereira and Alba Pages-Zamora, "Mean Square Convergence of Consensus Algorithms in Random WSNs", *IEEE Transactions on Signal Processing*, vol.58, no.5, pp.2866-2874, May 2010.

[20] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava, "Energy-aware wireless microsensor networks", *IEEE Signal Processing Magazine*, vol.19, pp.40-50, 2002.

[21] Jin-Jun Xiao, Alejandro Ribeiro, Zhi-Quan Luo, and Georgios B. Giannakis, "Distributed Compression-Estimation Using Wireless Sensor Networks", *IEEE Signal Processing Magazine*, vol.23, no.4, pp.27-41, July 2006.

[22] S. Kumar, T. Lai, and J. Balogh, "On k-coverage in a mostly sleeping sensor network", *in Proc. ACM MOBICOM Conf.*, 2004.

[23] Minh Hoang Trinhy and Hyo-Sung Ahn, Theory and applications of matrix-weighted consensus. https://arxiv.org/abs/1703.00129.

[24] Minh Hoang Trinh, Chuong Van Nguyen, Young-Hun Lim, Hyo-Sung Ahn, Matrix-weighted consensus and its applications, *Automatica*, vol.89, pp.415-419, 2018.

[25] Lalos, A.S.; Antonopoulos, A.; Kartsakli, E.; Di Renzo, M.; Tennina, S.; Alonso, L.; Verikoukis, C., "RLNC-Aided Cooperative Compressed Sensing for Energy Efficient Vital Signal Telemonitoring", *IEEE Transactions on Wireless Communications*, vol.14, no.7, pp.2685-3699, 2015.

[26] Ali Jadbabaie, Jie Lin, and A. Stephen Morse, "Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules", *IEEE Transactions on Automatic Control*, vol.48, no.6, pp.988-1001, June 2003.

[27] G. Anastasi, M. Conti and M. Di Francesco, "Extending the Lifetime of Wireless Sensor Networks Through Adaptive Sleep", *IEEE Transactions on Industrial Informatics*, vol.5, no.3, pp.351-365, August 2009.

[28] Maxim Raginsky, Sina Jafarpour, Zachary T.Harmany, and Robert Calderbank, "Performance Bounds for Expander-Based Compressed Sensing in Poisson Noise", *IEEE Transactions on Signal Processing*, vol.59, no.9, September 2011.

[29] J.-A. Bazerque and G. B. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity", *IEEE Transactions on Signal Processing*, vol.58, no.3, pp.1847-1862, March 2010.

[30] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed Average Consensus with Least-Mean-Square Deviation", *Journal of Parallel and Distributed Computing*, vol.67, no.1, pp.33-46, 2007.

[31] Dimitri P. Bertsekas and John N. Tsitsiklis, "Parallel and Distributed Computation: Numerical Methods", *Athena Scientific*, 1997.

[32] Ioannis D. Schizas, Alejandro Ribeiro, Georgios B. Giannakis, "Consensus in Ad Hoc WSNs With Noisy Links Part I: Distributed Estimation of Deterministic Signals", *IEEE Transactions on Signal Processing*, vol.56, no.1, pp.350-364, January 2008.

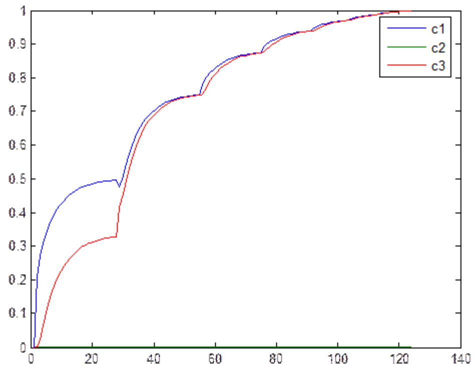
[33] T. Blumensath and M. Davies, "Sampling theorems for signals from the union of finite-dimensional linear subspaces", *IEEE Transactions on Information Theory*, vol.55, no.4, pp.1872-2882, 2009.

[34] Quang Duy La; Tony Q. S. Quek; Jemin Lee; Shi Jin; Hongbo Zhu, "Deceptive Attack and Defense Game in Honeypot-Enabled Networks for the Internet of Things", *IEEE Internet of Things Journal*, vol.3, no.6, pp.1025-1035, 2016.

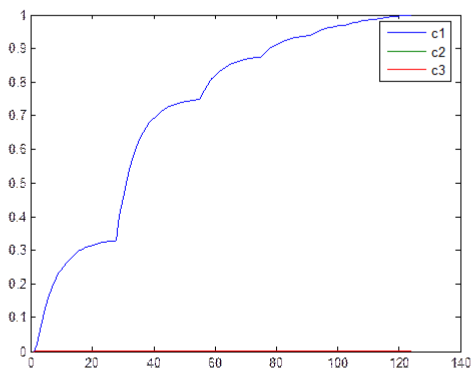
[35] Arian Maleki and David L. Donoho, "Optimally Tuned Iterative Reconstruction Algorithms for Compressed Sensing", *IEEE Journal of Selected Topics in Signal Processing*, vol.4, no.2, pp.330-341, April 2010.

[36] Mario A. T. Figueiredo, Robert D. Nowak, and Stephen J. Wright, "Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems", *IEEE Journal of Selected Topics in Signal and Processing*, vol.1, no.4, pp.586-597, December 2007.

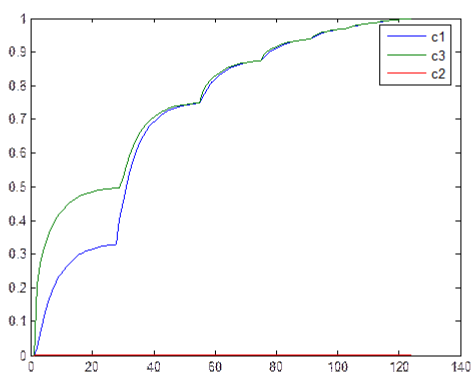
[37] Wei Ye, John Heidemann, and Deborah Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks", *IEEE/ACM Transactions on Networking*, vol.12, no.3, pp.493-506, June 2004.



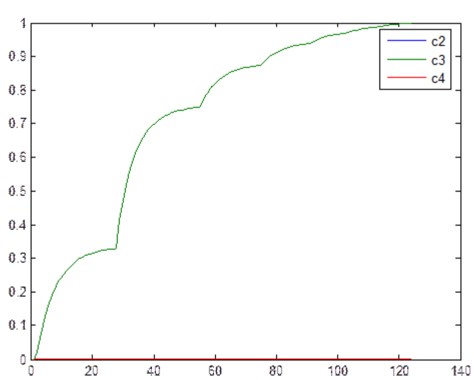
(a) Consensus result at n_1 vs Iteration number



(b) Consensus result at n_2 vs Iteration number



(c) Consensus result at n_3 vs Iteration number



(d) Consensus result at n_4 vs Iteration number

Fig. 6. Consensus algorithm on 4 active nodes vs Iteration number

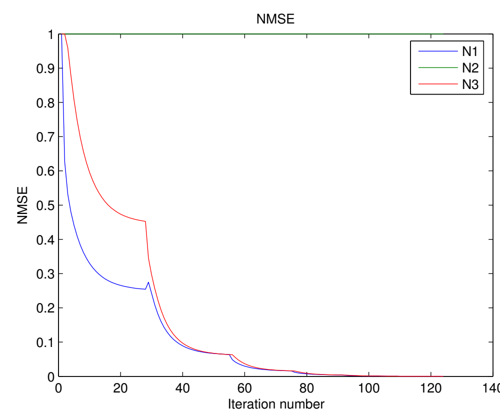


Fig. 7. NMSE at node n_1