Computer Communication & Collaboration (Vol. 6, Issue 4, 2018)
ISSN 2292-1028(Print) 2292-1036(Online)
DOIC:2292-1036-2018-11-002-96

Distributed Filtering with Consensus Updating for Internet of Things Network

Yee Ming Chen (Correspondence author)

Department of Industrial Engineering and Management, Yuan Ze University
Tao-Yuan, Taiwan, R.O.C.
E-mail: chenyeeming@saturn.yzu.edu.tw

Chu-Kai Wang[1] **,** ***Chi-Li Tsai***[2]

Department of Industrial Engineering and Management, Yuan Ze University
Tao-Yuan, Taiwan, R.O.C.
E-mail: [1]wanjk.tw@yahoo.com.tw; [2]fenikace@yahoo.com.tw

Abstract: Consensus approach for networked dynamic systems is an important research problem for distributed estimation in Internet of Things (IoT). In this paper, the distributed Kalman filter with consensus update approach was proposed to investigate the IoT of general multi-agent systems with links failure. The objective of the proposed approach is effectively to compensate for the undesirable effects of the failure links. Numerical simulations are provided to demonstrate the effectiveness of the approach results obtained.

Keywords: Kalman Filter, Consensus, Estimation

1 Introduction

Internet of Things (IoT) embedded heterogeneous devices for systems monitoring and controls have emerged. The IoT aims to interconnect devices with different capabilities such as sensors, actuators, smart objects (e.g. unmanned ground vehicles), and sensor nodes, within the same heterogeneous network. These are represented generically by networked control systems (NCS) and multi-agent systems (MAS) and can be effectively implemented by wireless sensor network. An interest in IoT has been on the rise due to its broad applications such like hazardous material handling, collaborative tracking, and distributed reconfigurable sensor networks etc. One of important problems is to estimate the state of the target by using local information. The sensors detect the fault and communicate. However, some sensors cannot get the observations of the cyber attack due to limited sensing range, obstacles, sensing faults, etc. This paper studies the distributed estimation problem with limited available observations. There have been many researchers investigating the problem of distributed estimation/filtering in sensor networks. Consensus approach is commonly applied in developing distributed filtering algorithm. For first-order processing with noise, consensus algorithm is applied in [1], [2] to optimizing the estimations. For general linear system, OlfatiSaber [3] combined the standard Kalman filter with consensus protocol to develop consensus-based Kalman filter and analyzed it for continuous-time and discrete-time systems, respectively. Meanwhile, as one of the important issues, IoT network based state estimation

problem has become a hot topic. Compared with the centralized estimation strategies, the distributed ones bring more advantages in robustness, parallel processing and so on. Many results of distributed Kalman filter (DKF) are based on consensus strategy [4]. Gao and Wang [5] proposed a consensus-based Kalman filter (CBKF), where the centralized observation is approximated by sensor network using a consensus strategy. Oh *et.al* [6] also reported a scalable suboptimal CBKF and proved that the error dynamics can converge to zero under the noise-free condition. The main idea is that each sensor carries out a standard Kalman filter with its own measurements to obtain an estimate and fuses the estimate with the ones of its neighbors so as to get an improved estimate. Regarding distributed communication strategies, a traditional one is the time-driven mechanism, in which local messages are broadcasted to neighbors at each periodic moment. The above work[7,8] on estimating general linear dynamical target focuses on a network with heterogeneous sensors. It is assumed that all sensors can obtain observations of the cyber attack. But in many applications some sensors may not able to get the observations. In this paper we consider the concept of graph theory for the consensus update approach on a time-varying sensing topology and the distributed estimation problem. We develop an algorithm to handle the time-varying IoT network topology in which not every node has local observations to generate own local tracking estimates. Our approach can be characterized as a collection of interconnected decision-making sensor nodes (or agents) with limited signal processing capabilities, locally sensed information, and limited inter-component communications, all seeking to achieve a collective (global) objective. Therefore we propose a distributed Kalman filtering filter with consensus update approach for such a IoT network model. The organization of the remaining part is presented as follows. In Section 2, provides some preliminaries on consensus problems in networked systems and graph theory. In Section 3, the main results on discussing of distributed Kalman filtering filter with consensus update approach are presented. Section 4 provides detailed numerical simulation results. Finally, concluding remarks are made in Section 5.

2 Preliminaries

Distributed tracking filter with consensus algorithm, addressed in this paper, refers to the problem that a group of sensor nodes that need to achieve an agreement over the state of a dynamical system by exchanging tracking estimates over the IoT network. For instance, IoT tracking could benefit from distributed tracking with consensus, due to the fact that individual sensor nodes may not have enough sensing of sufficient quality and different nodes may arrive at different local estimators regarding the same space object of interest. Information exchange among nodes may improve the quality of local estimators and help avoid conflicting and inefficient decisions. The common approach in establishing the theoretical background for the distributed tracking filter with consensus algorithm is based on graph theory.

In order to exchange information among IoT network, it is natural to model this by way of an undirected or directed graph, where vertices represent nodes and edges are the information exchange links among nodes. A pair (V, E) is called a directed graph where $V = \{1, \ldots, n\}$ is a nonempty finite node set and $E \subseteq V \times V$ is called an edge set. The neighbor of ith node is denoted by $N_i = \{j \in V : (i, j) \in E\}$. For the edge (i, j), i is known as parent node whereas j is mentioned as the child node. The edge $(i, j) \in E$ means that node j can get updates from node i but for node i it is not permissible. Contrary to a directed graph, the undirected graph can be seen as a special type of directed graph where un-ordered pairs of nodes are allowed. The edge $(i, j) \in E$ is referred to as node i and j and can receive updates from each other so edges (i, j) and (j, i) in the directed graph equate to an edge (i, j) in the undirected graph.

For a directed graph with a node set $V = \{1, \ldots, n\}$ the adjacency matrix $A = [a_{ij}] \in$

$R_{n\times n}$ is defined as a "weight" where a_{ij} = 1 if $(j, i) \in E$ and a_{ij} = 0 otherwise. As all the graphs have some weights so if weights are not significant in a particular situation, then a_{ij} is assumed to be equal to one for all $(j, i) \in E$. Self-edges with positive weight are also allowed. For some a_{ij} graph is referred to as balanced if $\Sigma_{i=1}^{n} a_{ij} = \Sigma_{j=1}^{n} a_{ji}$ for all i. As adjacency matrix is symmetric for undirected graph thus every undirected graph will be automatically balanced. Thus, the adjacency matrix can be written as

$$A = [a_{ij}] = \begin{cases} 1, & if \|q_j - q_i\| < r \\ 0, & otherwise \end{cases} \quad (1)$$

The directed graph Laplacian matrix $L = [l_{ij}] \in R^{n\times n}$ can be defined as $l_{ii} \sum_{j\neq i} a_{ij}$ and $l_{ij} = -a_{ij}$ for all $i \neq j$. Similarly, if $(j, i) \notin E$, then $l_{ij} = -a_{ij} = 0$. The graph Laplacian elements are defined as

$$L = [l_{ij}] = \begin{cases} -a_{ij}, & j \in N_i \\ |N_i|, & j = i \end{cases} \quad (2)$$

Here, $|N_i|$ is the number of neighbors of node i or simply its in-degree, where the degree matrix is defined as $D = diag(degin(1), \ldots, degin(N))$. For a directed graph, L is not necessarily symmetric however for an undirected graph, L is always symmetric. For a balanced graph, the degree-in and degree-out should be same.

$$deg_{in}(i) = \sum_{j=1}^{N} a_{ij} = \sum_{j=1}^{N} a_{ji} = deg_{out}(i) \quad (3)$$

Similarly, the Laplacian matrix in terms of degree matrix can be defined as "$L = D - A$" [9]. For an undirected graph, if ith smallest eigenvalue of Laplacian matrix is $\lambda_i(L)$ with $\lambda_1(L) \leq \lambda_2(L) \leq \ldots \leq \lambda_n(L)$ such that first eigenvalue is zero then convergence rate of consensus algorithms is quantified by $\lambda_2(L)$ which is known as algebraic connectivity [9].

The generic continuous consensus algorithm for updating the information state $x_i(t)$of node i, can be formulated as:

$$\dot{x}_i(t) = -\Sigma_{j=1}^{n} a_{ij}(t)[x_i(t) - x_j(t)] \quad (1)$$

Achieving consensus implies that for all $x_i(0)$ and all $i, j = 1, \ldots, n, |x_i(t) - x_j(t)| \to 0$, when t $\to \infty$. a_{ij} are positive weights in the case that $(i, j) \in \varepsilon_n$ and $a_{ij} = 0$ otherwise which correspond to the elements of the adjacency matrix $A_n \in \mathbb{R}^{n\times n}$ of the communication graph. In matrix form, consensus is expressed as:

$$\dot{x}(t) = -\mathcal{L}_n(t)x(t) \quad (2)$$

Where is the non-symmetric Laplace matrix of the directed graph, with $l_{ii} = \Sigma_{j=1, j\neq 1}^{n} a_{ij}, l_{ij} = -a_{ij}, i \neq j$.

The discrete version, for communication at discrete time instants is expressed in similar manner:

$$x_i[k+1] = \Sigma_{j=1}^{n} d_{ij}[k]x_j[k] \quad (3)$$

with the consensus condition that for all $x_i[0]$ și pentru toate $i, j = 1, \ldots, n$, $|x_i[k] - x_j[k]| \to 0$, $k \to \infty$ [10].

3 Problem Formulation

3.1 System Model

Consider an N-node IoT network with a connectivity graph $G(k)=(V, E(k))$ at time k made of both

mobile and fixed sensor nodes. Assume that the graph *G(k)* is undirected, but time varying due to nodes moving in and out of communication ranges of each other. Each node was equipped with a heterogeneous sensor with different measurement-noise covariance R_i. The state of the target is represented by vector $x \in R^p$, where p is the length of state vector x. The process-noise $w \in R_p$ is modeled as a zero-mean Gaussian random variable with covariance $Q \in R^{p \times p}$. At time instant *k*, the state dynamics of target on node N_i are modeled as follows:

$$\mathbf{x}(\mathrm{k}+1) = A(k)\,\mathbf{x}(k) + B(k)\omega\,(\mathrm{k}); \quad \mathbf{x}(0) \sim N(\bar{\boldsymbol{x}}(0), P_0) \tag{4}$$

where $\omega(\mathrm{k})$ is zero-mean white Gaussian process noise(WGN), $\mathbf{x}(0) \in \mathbb{R}^M$ is the initial state of the target, and $E[\omega(k)\omega(l)^T] = Q(\mathrm{k})\delta_{kl}$, with $\delta_{ii'} = 1$ if $i = i'$ and $\delta_{ii'} = 0$,otherwise. $A(k)$ is the model matrix, $B(k)$ is the state noise matrix. We are interested in tracking the state of this target using a IoT network with communication topology *G(k)=(V, E(k))* . Each node has a linear sensing model. The observations at node i and time k are

$$z_i(k) = H_i\ (k)\ x\ (k) + \nu_i\ (k) \tag{5}$$

where $z_i(k) \in R^{Pi}$ with $\Sigma_{i=1}^{n} pi = p$, $H_i(k) \in R^{Pi \times n}$ is the local observation matrix for node i, and $\nu_i\ (k)$ is the local observation noise. We refer to $z_i(k)$ as sensor data. Assume that $\nu_i\ (k)$ are zero mean white Gaussian noise and $E[\nu_i(k)\nu_i(l)^T] = R_i(k)\delta_{kl}$, with $\delta_{rs} = 1$ if $r = s$ and $\delta_{rs} = 0$, otherwise. Let the global observation vector $\mathbf{z}(k) \in R^p$, the global observation matrix $H(k) \in R^{p \times n}$, and the global observation noise vector $\nu(k) \in R^n$, Then the global observation model is given by

$$\mathbf{z}\ (k) = H(k)\ x\ (k) + \nu\ (k) \tag{6}$$

Given the collective information $Z(k) = \{\mathbf{z}(0), \mathbf{z}(1), \ldots, \mathbf{z}(k)\}$, the estimation of the state of the process can be expressed as

$$\begin{aligned} &\hat{x}(k) := \hat{x}\ (k \mid Z(k)) = E[x\ (k) \mid Z(k)]\ , \\ &\bar{x}\ (k) := \hat{x}\ (k \mid Z(k-1)) = E[x\ (k) \mid Z(k-1)]\ . \end{aligned} \tag{7}$$

We refer to $\hat{x}\ (k)$ and $\bar{x}\ (k)$ as estimate and prior estimate (or prediction) of the state $x\ (k)$, respectively. Then, the error covariance matrices associated with the estimates $\hat{x}\ (k)$ and $\bar{x}(k)$ are given by

$$\begin{aligned} &M(k) := E\ [\hat{x}\ ((k) - x(k))\ (\hat{x}\ (k) - x(k))^T] = E\ [\eta\ (k)\ \eta(k)^T]\ , \\ &P\ (k) := E\ [(\bar{x}\ (k) - x(k))\ (\bar{x}\ (k) - x(\mathrm{k}))^T] = E\ [\bar{\eta}\ (k)\ \bar{\eta}(k)^T]\ , \end{aligned} \tag{8}$$

where $\eta(k) = \hat{x}\ (k) - x(k)$ and $\bar{\eta}(k) = \bar{x}(k) - x(k)$ denote the estimate errors and $P(0) = P_0$. Then, the Kalman filter is a linear estimator in the form

$$\hat{x}\ (k) = \bar{x}\ (k) + K(k)\ (\mathbf{z}\ (k) - H(k)\ \bar{x}\ (k)) \tag{9}$$

with the Kalman gain $K(k)$.

Due to the importance of the node indices in this paper, we adopt a notation that is free of the time index k. The index-free form of the above the Kalman filter can be written as

$\hat{x} = \bar{x} + K(\mathbf{z} - H\bar{x})$. (10)

We also use the update operation $\{\cdot^+\}$ to rewrite the IoT network model of node i as

$$\begin{aligned} x^+ &= Ax + B\omega, \\ z_i &= H_i x + \nu_i. \end{aligned} \quad (11)$$

Then, we get the index-free recursive equations of the Kalman filter for system (11):

$$\begin{aligned} &\hat{x} = \bar{x} + K(\mathbf{z} - H\bar{x}), \\ &K = PH^T (R + HPH^T)^{-1}, \\ &M = P - PH^T (R + HPH^T)^{-1} HP, \\ &P^+ = AMA^T + BQB^T, \\ &\bar{x}^+ = A\hat{x}. \end{aligned} \quad (12)$$

3.2 Network Model

Assume that node i only receives information from its neighbors over network. In each local node Kalman filtering, let $N_i = \{j : (i, j) \in E\}$ be the set of neighbors of node i on graph G. Each node i of the IoT network communicates its measurement Z_i, covariance information R_i, and observation matrix H_i with its neighbors N_i. For node i, we assume that the information flow from neighboring nodes to node i is prohibited if there is no nodes except for its neighbors N_i exist. Therefore, node i can use foregoing the Kalman filter that only utilizes the observation vectors and observation matrices of the nodes in $J_i = N_i \cup \{i\}$. Then, the distributed Kalman filtering have the iterations of node i in local node as

$$\begin{aligned} &y^i = \sum_{j \in Ji} H_j^T R_j^{-1} z_j = \sum_{j \in Ji} y_j \\ &S^i = \sum_{j \in Ji} H_j^T R_j^{-1} H_j = \sum_{j \in Ji} S_j, \\ &\hat{x}_i = \bar{x}_i + M_i(y^i - S^i \bar{x}), \\ &M_i = (P_i^{-1} + S^i)^{-1}, \\ &P_i^+ = AM_iA^T + BQB^T, \\ &\bar{x}_i^+ = A\hat{x}_i, \end{aligned} \quad (13)$$

where y^i and S^i are local aggregate information vector and matrix, respectively and node i locally computes both y^i and S^i.

3.3 Consensus on Estimates

Consider a team of n agents to agree on specific consensus states, and at any discrete-time instant τ, the communication IoT topology between n agents can be described by the graph $G[\tau] = (V, E[\tau])$, the graph G is undirected, $V = \{1,2,\mathrm{K}\ ,n\}$ is the vertex set, and $E[\tau] \subseteq V \times V$ is the

edge set. In the consensus approach, each agent in the network maintains a local node copy of the consensus state $\zeta_i \in R^n$ and updates this value using its neighbors' consensus states according to the rule:

$$\zeta_i[\tau+1] = \zeta_i[\tau] + \sum_{j=1}^{n} \beta_{ij}[\tau](\zeta_i[\tau] - \zeta_i[\tau]), \tag{14}$$

where τ indicates the consensus update iteration step. The weights $\beta_{ij}[\tau]$ can use the Metropolis weights which preserves the averaging in consensus approach and can be represented by

$$\beta_{ij}[\tau] = \begin{cases} \left(1+\max\left\{d_i[\tau], d_j[\tau]\right\}\right)^{-1} \\ 1 - \sum_{(i,l)\in E(\tau)} \beta_{il}[\tau] \\ 0 \end{cases} \tag{15}$$

where $d_i[\tau]$ is the degree of node i in the graph $G[\tau]$. Arrange the local consensus states into the vector $\zeta[\tau] = \left[\zeta_1^T[\tau], \mathrm{K}, \zeta_n^T[\tau]\right]^T$, and define the matrix $(B[\tau])_{ij} = \beta_{ij}[\tau]$ for $i \neq j$; otherwise $(B[\tau])_{ii} = 1 - \sum_{(i,l)\in E(\tau)} \beta_{il}[\tau]$, and we can rewrite the update in (11) as

$$\zeta[\tau+1] = (B[\tau] \otimes I)\zeta[\tau], \tag{16}$$

where I is the appropriate size identity matrix and $\otimes$ denotes the matrix Kronecker product.The ij -th element of $B[\tau]$ in (16) satisfies the following conditions:

(1) $(B[\tau])_{ij} \geq 0$,

(2) $\sum_i (B[\tau])_{ij} = 1$,

(3) $\sum_j (B[\tau])_{ij} = 1$,

We now present the distributed Kalman filtering with consensus update approach. The distributed Kalman filtering uses consensus update approach on the state estimate, where each node maintains a local node Kalman filter. Corresponding to (11), let $\zeta_i[\tau] = \bar{x}_i$ be the prior estimate at time τ and $\zeta_i[\tau+1] = \bar{x}_i^c$ be fused prior estimate; each node fuses the prior estimates from its neighbors according to the rule:

$$\bar{x}_i^c = \bar{x}_i + \sum_{j\in N_i} \beta_{ij}[\tau](\bar{x}_j - \bar{x}_i). \tag{17}$$

Using the fused prior estimate $\bar{x}_i^c$, the filter estimate at node i could be implemented by

$$\begin{aligned} \hat{x}_i &= \bar{x}_i^c + M_i\left(y^i - S^i \bar{x}_i^c\right) \\ &= \bar{x}_i + M_i\left(y^i - S^i \bar{x}_i\right) + \left(I - M_i S^i\right)\sum_{j\in N_i} \beta_{ij}[\tau](\bar{x}_j - \bar{x}_i). \end{aligned} \tag{18}$$

The distributed Kalman filtering with consensus update approach is summarized in the following algorithm as

Initialization (for node i):

$$\bar{x}_i = x(0) \qquad P_i = P_0$$
$$\tau = 1 \qquad \tau_p = \tau + T_p$$

Loop {Local iteration on node i }

Step 1. Consensus update

$$\bar{x}_i^c = \bar{x}_i + \sum_{j \in N_i} \beta_{ij}[\tau](\bar{x}_j - \bar{x}_i)$$
$$y^i = \sum_{j \in J_i} H_j{}^T R_j^{-1} z_j$$

$$S^i = \sum_{j \in J_i} H_j{}^T R_j{}^{-1} H_j$$

$$\tau \leftarrow \tau + 1$$

Step 2. If new observations are taken then the distributed Kalman consensus state estimate are updated

$$\hat{x}_i = \bar{x}_i^c + M_i\left(y^i - S^i \bar{x}_i^c\right)$$
$$= \bar{x}_i + M_i\left(y^i - S^i \bar{x}_i\right) + \left(I - M_i S^i\right)\sum_{j \in N_i} \beta_{ij}[\tau](\bar{x}_j - \bar{x}_i)$$

$$M_i = \left(P_i^{-1} + S^i\right)^{-1}$$

Step 3. If time for a predication step (i.e., $\tau = \tau_p$) then prediction step

$$P_i^+ = AM_iA^T + BQB^T$$
$$\bar{x}_i^+ = A\hat{x}_i$$
$$\tau_p = \tau + T_p$$

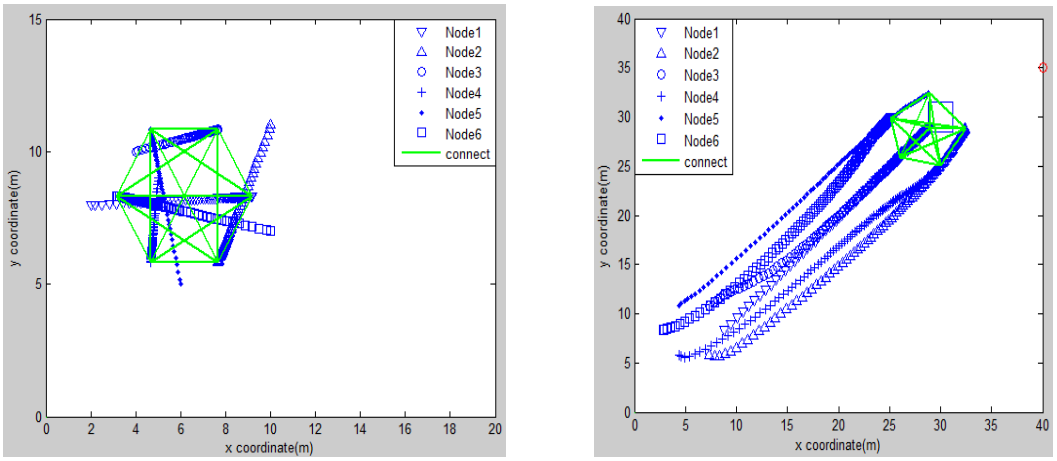
End loop

where τ is the time index for the consensus update approach and $T_p \in Z^+$ is the time interval between prediction updates. One-time step $k\text{-}1 \to k$ is equivalent to T_p time steps of the consensus time index $\tau \to \tau + 1$; that is, for each node, the information exchanges between neighboring nodes occurred faster than the prediction update step. The three steps in distributed Kalman filtering

prediction, local filter estimate, and consensus approach are update sequential.

4 Numerical Simulations

In this section, we apply the distributed Kalman filtering with consensus update approach to the IoT of networked control problems. We have assumed that nodes can communicate in a connected graph. Consider a simple IoT network consisting of six sensor nodes, under a communication protocol and link connected network nodes, while each monitoring its neighbors to detect a cyber attack and identify the faulty in the network which is connected according to the communication graph. The distributed Kalman filtering with consensus update approach can be achieved for multi-agent networks with quantization. The simulations are done using MATLAB. First, a fault/attack free formation mobile of six nodes (or agents) presented to see if the approach managed to keep a specified hexagon formation mobile of the agents. In the following trajectory graphs Fig 1(a) and Fig 1(b), the movement of the nodes from its initial position to the final position and the dotted curve represents the position of nodes (or agents) at different instances of time. As can be seen for Fig. 1, the proposed approach ensures that all nodes very closely follow the formation trajectory.



(a) (b)

Figure 1. Formation of nodes and trajectory of hexagon formation of agents

Internally, the control values are [9.42 , 9.35 , 9.18 , 9.04 , 8.91 , 8.85] ^{T}, respectively. The IoT network topology is displayed in Fig. 1. The main result of this case is illustrated in Fig. 2. It can be seen how from initial control states uniformly distributed across a given interval, the six nodes converge by information exchange and consensus(average=9.127 with red dotted line). Convergence time in this case is around 4 seconds. Suppose some links fail at time t = 20. By using the proposed approach, it can be seen from Fig. 3 that the network can achieve consensus (average=9.228, denoted with black dotted line)since the links of the network are destroyed by the external attack.

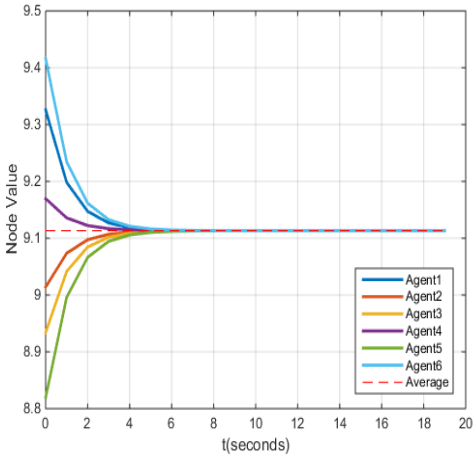


Figure 2. The state estimation trajectory of IoT network

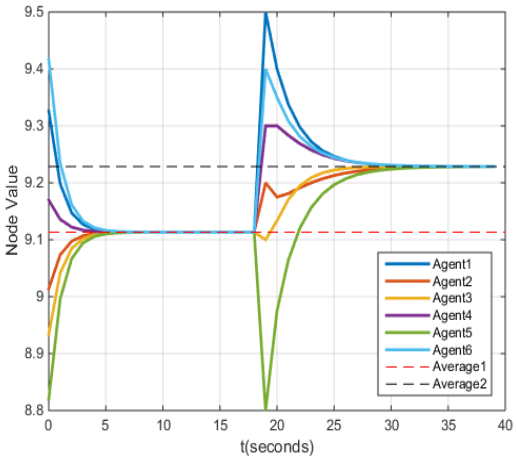


Figure 3. The state estimation trajectory of IoT network after links failure

To demonstrate the recovery effectiveness of the proposed approach, we assume that the links fail (denoted red circles in Fig. 5)between time 20 and 39 and recovery after t= 40, respectively(Fig. 4). If the approach is applied, we can see from Fig. 5 that the IoT network will converge to the same final state as the original network in Fig. 5.

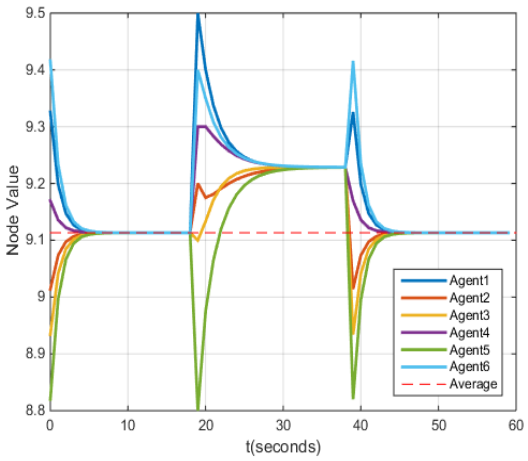


Figure 4. The evolution of the IoT network From links failure to recovery

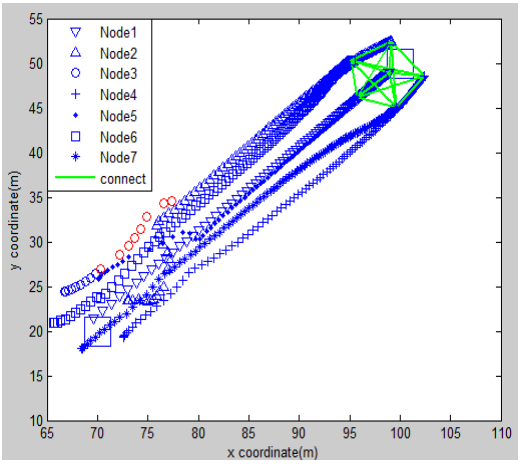


Figure 5. The state estimation trajectory of IoT network after recovery

5 Conclusions

In this paper, detection of cyber attack or fault has been considered on a network of IoT in formation movement. The proposed approach not only able to detect a fault /cyber attack but also successfully identified and recovery nodes in the formation network. Furthermore, an approach has been proposed to safely and automatically recovery the faulty node under attack while keeping the formation with degraded but functioning performance. Finally, a numerical simulation cases study

have been given with a typical example of six nodes in a hexagon formation with a possible node and communication cyber attacks. Finally, a numerical case study has demonstrated that the residual generated at the monitoring node able to successfully detect and isolate the cyber attack. Also, the faulty node recovery has been shown effectively to maintain the formation accordingly. Future work includes extension of the proposed approach to handle more complex attack patterns and applying the method for other types of sensor nodes coordination missions.

Acknowledgments

This research work was sponsored by the Ministry of Science and Technology, R.O.C., under project number MOST 106-2221-E-155-028.

References

[1]. Guo, M., Dimarogonas, D. V., and Johansson, K. H.(2012), "Distributed real-time fault detection and isolation for cooperative multi-agent systems,", *2012 American Control Conference (ACC), IEEE*, 5270-5275.

[2]. Mirali, F. and Werner, H. (2017), "Distributed weighting strategies for improved convergence speed of first order consensus", *American Control Conference.*

[3]. Olfati-Saber, B.R., Fax, J.A., and Murray, R.(2007)," Consensus and Cooperation in Networked Multi-Agent Systems. *Proc. IEEE 2007*, 95, 215–233.

[4]. Andreasson, M., Dimarogonas, D.V., Sandberg, H.; and Johansson, K.H.(2014)," Distributed control of networked dynamical systems: Static feedback, integral action and consensus". *IEEE Trans. Autom. Control* 2014, 59, 1750–1764.

[5]. Gao, Y., and Wang, L. (2011), "Sampled-data based consensus of continuous-time multi-agent systems with time-varying topology. IEEE Trans. Autom. Control 56, 1226–1231.

[6]. Oh, K.K., Park, M.C., Ahn, H.S. (2014), "A survey of multi-agent formation control", *Automatica*, 53, 424–440.

[7]. Chen, C., Chen, G., and Guo, L.(2015)," Consensus of flocks under M-nearest-neighbor rules", *J. Syst. Sci. Complex*.28, 1–15.

[8]. Patterson, S., and Bamieh, B. (2008), "Distributed Consensus with Link Failures as a Structured Stochastic Uncertainty Problem", in Proceedings of the 46th Allerton Conference on Communication, Control and Computing, pp. 623–627

[9]. Abderrazak, A.; El Fouly, T.M.(2013), "On the distributed binary consensus algorithm in wireless sensor networks," in Signal Processing and Communication Systems (ICSPCS), 7th International Conference , 12-18.

[10]. Gupta, S.K.; Kar, K.; Mishra, S.; Wen, J.T., (2015),"Collaborative Energy and Thermal Comfort Management Through Distributed Consensus Algorithms," in Automation Science and Engineering, IEEE Transactions ,12(4), 1285-1296.