An Optimal Distributed Fault Detection Scheme for Large-Scale Systems with Deterministic Disturbances

Jiarui Zhang [*] **, Linlin Li** [*]

[*] *Institute for Automatic Control and Complex Systems (AKS),
University of Duisburg-Essen, 47057 Duisburg, Germany,
(e-mail: jiarui.zhang@uni-due.de, linlin.li@uni-due.de).*

Abstract: The main objective of this paper is to develop an optimal distributed fault detection (FD) approach for large-scale systems in the presence of unknown deterministic disturbances using the measurement of sensor networks. To be specific, the design approach consists of two phases: the distributed offline training phase and the online implementation phase. The offline training phase includes distributed iterative learning and average consensus algorithm. It is worth mentioning that, the distributed approach avoids enormous computational costs and complex information exchanges. Finally, a numerical example is illustrated to show that the distributed approach can successfully and efficiently accomplish the FD task.

Keywords: Distributed fault detection, optimal fault detection, deterministic disturbances

Notation: The notation adopted throughout this paper is fairly standard. In addition, we use

$\|x(k)\|_{RMS}$	$\|x(k)\|_{RMS} = \left(\dfrac{1}{m} \sum_{l=0}^{m-1} x^\top (k+l) x(k+l) \right)^{1/2}$
(A, B, C, D)	shorthand for $C(zI - A)^{-1}B + D$
$G^*(z)$	shorthand for $G^\top(z^{-1})$
$\lambda_{max}(A)$	maximum eigenvalue of matrix A
$\rho(A)$	spectral radius of matrix A
$\mathbf{1}$	column vector with all elements one

1. INTRODUCTION

In last decades, intensive attention has been paid to the distributed FD strategies, which provide the possibility to combine advanced FD approaches, as reported in (Gao et al., 2015), and the distributed methods (Garin and Schenato, 2010) for more effective and automatic approaches in large-scale system when compared with centralized manners. For such methods, each subsystem can itself execute the FD algorithms based on the local measurement data as well as the data received via networks from its neighboring subsystems. For model-based approaches, (Olfati-Saber, 2005) provides us an approach to use consensus technique to achieve Kalman filter in a distributed way. (Boem et al., 2016) deals with the FD task for interconnected systems with measurement noise and time delays. (Zhang and Yang, 2018) presents a distributed method for detecting and isolating the faults in multi-agent systems. For data-driven instances, in (Jiang et al., 2017) and (Chen et al., 2019), distributed data-driven fault detection approaches based on canonical correlation analysis (CCA) are proposed to address the process monitoring problem. In addition, based on principal component analysis (PCA), (Jiang et al., 2015) proposes a fault-relevant variable selection and bayesian inference-based distributed method for efficient distributed FD. However, most of them deal with the system influenced by stochastic noise.

Motivated by aforementioned observations, this paper focuses on developing a distributed fault detection approach for large-scale systems with deterministic disturbances using sensor network. The approach consists of two parts: distributed offline training, which is designed to train important parameters, and distributed online implementation, which is applied to achieve distributed FD purpose.

This paper is organized as follows: in Section 2, system model, sensor network model and the optimal centralized FD approach are briefly described. It is followed by the corresponding distributed realization scheme, which consists of distributed offline learning and online implementation in Section 3. A numerical example shows the effectiveness of the distributed algorithm in Section 4.

2. PRELIMINARIES AND PROBLEM FORMULATION

In this section, we introduce the models of the dynamic system and sensor network, an optimal FD scheme and the problem formulation.

2.1 Process Description

We consider an linear time-invariant (LTI) system:

$$x(k+1) = Ax(k) + E_l d_l(k) + E_f f(k), \tag{1}$$

where $x(k) \in \mathcal{R}^q$ denotes the state vector, $f(k) \in \mathcal{R}^f$ denotes the fault, $d_l(k) \in \mathcal{R}^p$ denotes the unknown deterministic disturbance, which is norm bounded by

$$\|d_l(k)\|_{RMS} \leqslant \delta_d.$$

⋆ This work has been supported by the programme for the promotion of excellent early career researchers in University of Duisburg-Essen.

To monitor the process, the system under consideration is equipped with a sensor network, which consists of n sensor nodes and each sensor node is modelled by

$$y_i(k) = C_i x(k) + F_i d_l(k), \tag{2}$$

where $i \in \{1, \cdots, n\}$ denotes the index of sensor node and $y_i(k) \in \mathcal{R}^{m_i}$ is the measurement of sensor node i.

The communication topology of sensor network is described by an undirected and connected graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, where $\mathcal{N} = \{1, \cdots, n\}$ denotes the node set and $\mathcal{E} \subseteq \{\mathcal{N} \times \mathcal{N}\}$ stands for the edge set. The edge $(i,j) \in \mathcal{E}$ between two nodes i and j means that they are neighbors and the data can be transferred between them, and in undirected graph $(i,j) \in \mathcal{E}$ also means $(j,i) \in \mathcal{E}$. In a connected graph, each two nodes are connected by at minimum one path, which consists of several edges. We define $d(i,j)$ as the minimal length of the path connecting node i and j,

$$D_g = \max_{i,j \in \mathcal{N}} d(i,j)$$

as the diameter of $\mathcal{G}$, $\mathcal{N}_i$ as the set of all neighbors of node i, and $c_i = |\mathcal{N}_i|$ as the cardinality of $\mathcal{N}_i$. Moreover, we set

$$\Sigma = [\Sigma_{ij}],\ \Sigma_{ij} = F_i F_j^\top. \tag{3}$$

In this study, the communication topology is established based on the correlation relation between F_i and F_j, i.e.,

$$\begin{array}{l} i,j \in \{1, \cdots, n\},\ j \in \mathcal{N}_i \text{ if } \Sigma_{ij} \neq 0 \text{ and } i \neq j, \\ i,j \in \{1, \cdots, n\},\ j \notin \mathcal{N}_i \text{ if } \Sigma_{ij} = 0. \end{array} \tag{4}$$

2.2 An Optimal FD Scheme

In this section, we recall the existing optimal FD scheme for system (1) proposed in (Ding, 2008), which lays the basis for further distributed studies.

Under the assumption that all C_i, F_i and y_i, $i \in \{1, \cdots, n\}$, can be collected at one central station, we stack all sensor nodes together to model the overall system as

$$\begin{array}{r} x(k+1) = Ax(k) + E_l d_l(k) + E_f f(k) \\ y_l(k) = C_l x(k) + F_l d_l(k) \end{array}. \tag{5}$$

where

$$y_l(k) = \begin{bmatrix} y_1(k) \\ \vdots \\ y_n(k) \end{bmatrix},\ C_l = \begin{bmatrix} C_1 \\ \vdots \\ C_n \end{bmatrix},\ F_l = \begin{bmatrix} F_1 \\ \vdots \\ F_n \end{bmatrix}.$$

For purpose of FD, we adopt the following observer-based residual generator:

$$\hat{x}(k+1) = A\hat{x}(k) + L(y_l(k) - C_l \hat{x}(k)), \tag{6}$$

$$r(k) = V(y_l(k) - C_l \hat{x}(k)), \tag{7}$$

where $\hat{x}(k) \in \mathcal{R}^q$ represents the state estimation, $r(k) \in \mathcal{R}^r$ denotes the residual signal, and L and V represents the matrices to be determined. It is straightforward that the residual dynamics can be described by

$$r(z) = V(G_d(z) d_l(z) + G_f(z) f(z)), \tag{8}$$

$$G_d(z) = (A - LC_l, E_l - LF_l, C_l, F_l), \tag{9}$$

$$G_f(z) = (A - LC_l, E_f, C_l, O). \tag{10}$$

Without loss of generality, an fault detection problem is formulated by designing the residual generator such that the residual signal is robust against disturbance and meanwhile sensitive to fault. In this paper, an optimal fault detection problem is formulated as

$$\max_{L,V} \frac{\|VG_f(z)\|_\infty}{\|VG_d(z)\|_\infty}. \tag{11}$$

To deal with it, (Ding, 2008) provides a design scheme, which is reviewed in the following lemma.

Lemma 1. Given system (5), and residual generator (6) and (7), L and V, which are calculated by

$$R = C_l X C_l^\top + F_l F_l^\top, \tag{12}$$

$$V = R^{-1/2}, \tag{13}$$

$$L = (AXC_l^\top + E_l F_l^\top)R^{-1}, \tag{14}$$

$$X = AXA^\top - LRL^\top + E_l E_l^\top, \tag{15}$$

with $X > 0$, deliver the solution to the optimal FD problem (11).

Proof. See (Ding, 2018). ■

Moreover, according to (Ding, 2014), with L and V as the solution to (12)-(15), the transfer function VG_d is co-inner, which means $VG_d G_d^* V^* = I$, and $\|r(k)\|_{RMS} \leqslant \delta_d$ in fault-free case. Thus, for the purpose of FD, we set the evaluation function as $J(k) = \|r(k)\|_{RMS}$, threshold as $J_{th} = \delta_d$, and the decision logic as

$$\begin{cases} J(k) \leqslant J_{th}, \text{ fault-free} \\ J(k) > J_{th}, \text{ faulty} \end{cases}. \tag{16}$$

In order to ensure that discrete time algebraic Riccati equation (DARE) (15) has stabilizing solution, we make the following assumptions:

1. (C_l, A) is detectable;
2. $F_l F_l^\top$ is invertible;
3. $\forall \theta \in [0, 2\pi]$, $\begin{bmatrix} A - e^{j\theta} I & E_l \\ C_l & F_l \end{bmatrix}$ has full row rank.

It is evident that Σ in (3) yields $\Sigma = F_l F_l^\top$, also combining (3) and the second assumption shows that Σ is symmetric and positive definite matrix with all elements being real number.

2.3 Problem Formulation

It is worth mentioning that to achieve the optimal fault detection approach given in (6)-(16), all the information about sensors and measurement data should be collected at one central station to perform the FD actions, which requires significant computation and communication efforts. To deal with this issue, we investigate a distributed realization of the proposed optimal FD scheme, which delivers the exactly same optimal FD performance at each node.

3. A DISTRIBUTED FAULT DETECTION SCHEME

As mentioned above, in this section, a distributed realization of the proposed optimal FD scheme is presented, which is achieved by performing the following two phases:

– distributed offline training;
– distributed online fault detection.

In this study, we assume that at node i, only system information A_l, E_l, C_i and F_i, and the local measurement y_i are available; and the information can be transmitted between neighbors according to the communication topology (4).

3.1 Distributed Offline Training

For our purpose, (15) should be solved to obtain X at each node in a distributed way. To deal with this issue, we define

$$I_1 = C_l^\top \Sigma^{-1} C_l,\ I_2 = F_l^\top \Sigma^{-1} C_l,$$
$$I_3 = F_l^\top \Sigma^{-1} F_l,\ \Omega = (X^{-1} + I_1),$$

and apply the matrix identity

$$R^{-1} = \Sigma^{-1} - \Sigma^{-1} C_l \Omega^{-1} C_l^\top \Sigma^{-1} \tag{17}$$

to reformulate (15) into

$$\begin{aligned} X =& A\Omega^{-1}A^\top - A\Omega^{-1} I_2^\top E_l^\top - E_l I_2 \Omega^{-1} A^\top \\ &+ E_l(I - I_3 + I_2 \Omega^{-1} I_2^\top) E_l^\top. \end{aligned} \tag{18}$$

It follows from (18) that, for solving X, I_1, I_2 and I_3 should be calculated first. Since they have similar structure, for simplicity, we construct them as the unified form

$$\Lambda^\top \Sigma^{-1} \Phi = \begin{bmatrix} \Lambda_1^\top & \cdots & \Lambda_n^\top \end{bmatrix} \begin{bmatrix} \Sigma_{11} & \cdots & \Sigma_{1n} \\ \vdots & \ddots & \vdots \\ \Sigma_{n1} & \cdots & \Sigma_{nn} \end{bmatrix}^{-1} \begin{bmatrix} \Phi_1 \\ \vdots \\ \Phi_n \end{bmatrix},$$

where Λ and Φ can be C_l or F_l, and Σ is the same as introduced in (3). We now introduce a new parameter Z as

$$\Lambda^\top \Sigma^{-1} = Z^\top = \begin{bmatrix} Z_1^\top & \cdots & Z_n^\top \end{bmatrix} \tag{19}$$

and divide the calculation of $\Lambda^\top \Sigma^{-1} \Phi$ into two phases: the calculation of $\Lambda^\top \Sigma^{-1} = Z^\top$ and further $Z^\top \Phi$. Due to the reason that the first phase includes matrix inverse, which can cause huge computational costs, we apply distributed iteration learning to avoid this problem and to solve the first phase; and the average consensus technique can be used to achieve the second phase to obtain a common $\Lambda^\top \Sigma^{-1} \Phi$ in each sensor node.

3.1.1 Distributed Iteration Learning

For calculating (19) and avoiding the inverse computation, we adopt the iteration learning method

$$Z(\zeta + 1) = Z(\zeta) + \lambda(\Lambda - \Sigma Z(\zeta)), \tag{20}$$

with ζ as the iteration number and λ being a constant, which is designed to guarantee the iteration convergence

$$\lim_{\zeta \to \infty} Z(\zeta) = Z. \tag{21}$$

It is evident that equation (19) leads to $\Lambda = \Sigma Z$. By setting $e(\zeta) = Z(\zeta) - Z$, we identify the dynamic of $e(\zeta)$ as

$$e(\zeta + 1) = (I - \lambda\Sigma) e(\zeta). \tag{22}$$

It is clear that when $(I - \lambda\Sigma)$ is a Schur matrix, which means all eigenvalues of $(I - \lambda\Sigma)$ are located inside the unit disk,

$$\lim_{\zeta \to \infty} e(\zeta) = 0, \tag{23}$$

which also implies (21). Thus, $(I - \lambda\Sigma)$ being Schur matrix is the requirement to guarantee (21). In what follows, we introduce the condition for λ to ensure that $(I - \lambda\Sigma)$ is Schur matrix.

Proposition 1. The condition

$$0 < \lambda < \frac{2}{\|\Sigma\|_\infty} \tag{24}$$

can ensure that $(I - \lambda\Sigma)$ is Schur matrix, when Σ is symmetric and positive definite matrix with all elements being real numbers.

Proof. See Appendix A. ■

However, the above algorithm and analysis are under centralized manner. For our purpose, in the sequel, a distributed realization of (20) and a distributed determination of λ are introduced.

For the distributed realization of (20), we execute

$$Z_i(\zeta + 1) = Z_i(\zeta) + \lambda(\Lambda_i - \sum_{j \in \{i,\ \mathcal{N}_i\}} \Sigma_{ij} Z_j(\zeta)) \tag{25}$$

at node i, $i \in \{1, \cdots, n\}$. It is clear that in (25), node i only uses its local and neighbors' information. And according to (4), by stacking all (25) together, we obtain exactly the same equation as (20). Thus, (25) can be used as a distributed realization of (20).

For the distributed determination of λ, based on Proposition 1, we provide a distributed algorithm (**Algorithm 1**) as shown below to calculate λ at each node in a distributed way.

```
Algorithm 1
Step 1   Communicate $F_j$ to node $i$ for $j \in \mathcal{N}_i$
Step 2   Calculate $Q_i = \| \begin{bmatrix} F_i F_1^\top & \cdots & F_i F_n^\top \end{bmatrix} \|_\infty$ at node $i$
Step 3   Communicate $Q_j$ to node $i$ for $j \in \mathcal{N}_i$
Step 4   Update $Q_i = max\{Q_i, Q_{\mathcal{N}_i}\}$ at node $i$ with
         $Q_{\mathcal{N}_i} = \{Q_j | j \in \mathcal{N}_i\}$
Step 5   Repeat Step 3 and Step 4 $D_g$ times
Step 6   Calculate $\lambda = \lambda_i = \frac{\alpha}{Q_i}$ at node $i$ with $0 < \alpha < 2$
```

Proposition 2. With **Algorithm 1**, we have

$$i, j = 1, \cdots, n,\ Q_i = Q_j = \|\Sigma\|_\infty,$$

and $\lambda_i = \lambda_j = \lambda$ are obtained to fulfill (24) in a distributed way.

Proof. See Appendix B. ■

After calculating λ in each node, the distributed iteration learning (25) can be executed and terminate when

$$\|Z_i(\zeta + 1) - Z_i(\zeta)\|_2 \leqslant \epsilon,$$

where ϵ is a predefined tolerance, is fulfilled. Define

$$I_4 = C_l^\top \Sigma^{-1} = \left[I_{4,1} \ \cdots \ I_{4,n} \right],$$
$$I_5 = F_l^\top \Sigma^{-1} = \left[I_{5,1} \ \cdots \ I_{5,n} \right].$$

After executing distributed iteration learning, $Z_i := Z_i(\zeta)$ is obtained. And as a final result, $Z_i^\top \Phi_i$ is available at node i, $i \in \{1, \cdots, n\}$, where Z_i can be $I_{4,i}$ or $I_{5,i}$ and Φ_i can be C_i or F_i.

3.1.2 Average Consensus Approach

Since $Z_i^\top \Phi_i$ is available after performing distributed iteration learning, the average consensus algorithm is further applied to calculate $\Lambda^\top \Sigma^{-1} \Phi$ in each node. To this end, we first introduce the basis of average consensus algorithm. Without loss of generality, the average consensus algorithm is achieved by performing

$$\vartheta_i(\xi + 1) = w_{ii} \vartheta_i(\xi) + \sum_{j \in \mathcal{N}_i} w_{ij} \vartheta_j(\xi),$$

where ξ is the iteration number and $w_{ij} = 0$ if $j \notin \mathcal{N}_i$ and $j \neq i$. It is clear that, if $W = [w_{ij}]$ is chosen such that

$$\lim_{\xi \to \infty} W^\xi = \frac{1}{n} \mathbf{1}\mathbf{1}^\top, \tag{26}$$

we have

$$\lim_{\xi\to\infty}\begin{bmatrix}\vartheta_1(\xi)\\ \vdots\\ \vartheta_n(\xi)\end{bmatrix} = \mathbf{1}\otimes(\frac{1}{n}\sum_{i=1}^{n}\vartheta_i(0)). \tag{27}$$

To guarantee (26), (Xiao and Boyd, 2004) provides conditions for W, which are reviewed in following lemma.

Lemma 2. The equation (26) holds if and only if

$$\mathbf{1}^\top W = \mathbf{1}^\top, \tag{28}$$

$$W\mathbf{1} = \mathbf{1}, \tag{29}$$

$$\rho(W - \frac{1}{n}\mathbf{1}\mathbf{1}^\top) < 1. \tag{30}$$

Proof. See (Xiao and Boyd, 2004). ■

Numerous methods have been proposed to solve W, which yield Lemma 2. In this study, we use Metropolis-Hastings weights method (Xiao et al., 2007), which can be calculated under distributed manner as

$$w_{ij} = \begin{cases}\dfrac{1}{max\{c_i, c_j\}+1} & j \in \mathcal{N}_i\\ 0 & j \notin \mathcal{N}_i \text{ and } j \neq i\\ 1 - \sum\limits_{q\neq i} w_{iq} & i = j\end{cases}. \tag{31}$$

Then by setting $\vartheta_i(0) = Z_i^\top\Phi_i$ and after executing average consensus, we have $\vartheta_i = n^{-1}Z^\top\Phi$ at node i. By multiplying ϑ_i with n, we can obtain $\Lambda^\top\Sigma^{-1}\Phi$, which means that we obtain and save I_1, I_2 and I_3 at each node.

With I_1, I_2 and I_3 at hand, we calculate (15) by the iteration (32) at each node,

$$\begin{cases} X(\eta+1) = \Xi(\eta) - \Psi(\eta) - \Psi^\top(\eta) + \Pi(\eta)\\ \Omega(\eta) = (X^{-1}(\eta) + I_1)\\ \Xi(\eta) = A\Omega^{-1}(\eta)A^\top\\ \Psi(\eta) = A\Omega^{-1}(\eta)I_2^\top E_l^\top\\ \Pi(\eta) = E_l(I - I_3 + I_2\Omega^{-1}(\eta)I_2^\top)E_l^\top\end{cases}, \tag{32}$$

where η is the iteration number. According to (Komaroff, 1994), it holds that

$$\lim_{\eta\to\infty} X(\eta) = X > 0,\ X(0) > 0.$$

Then

$$\lim_{\eta\to\infty}\Omega(\eta) = \Omega = (X^{-1} + I_1)$$

can be obtained at each node. Moreover, we apply (17) to reformulate (14) into

$$L = (A - E_lI_2)\Omega^{-1}I_4 + E_lI_5 = [L_1\ \cdots\ L_n].$$

Or equivalently,

$$L_i = (A - E_lI_2)\Omega^{-1}I_{4,i} + E_lI_{5,i}. \tag{33}$$

In (33), with all parameters known in the right side at each node, L_i can be identified at each node.

3.2 Distributed Online Fault Detection

After calculating the important parameters through distributed offline training, we execute online implementation to realize optimal FD scheme in a distributed way.

For the distributed realization of online FD, we run the state observer (6) as

$$\hat{x}_i(k+1) = A\hat{x}_i(k) + L\bar{r}(k) = A\hat{x}_i(k) + \sum_{i=1}^{n}L_i\bar{r}_i(k) \tag{34}$$

at node i, $i \in \{1,\cdots,n\}$. Here, $\hat{x}_i(k)$ denotes the estimation of $x(k)$ at node i and

$$\bar{r}(k) = \begin{bmatrix}\bar{r}_1(k)\\ \vdots\\ \bar{r}_n(k)\end{bmatrix} = \begin{bmatrix}y_1(k) - C_1\hat{x}_1(k)\\ \vdots\\ y_n(k) - C_n\hat{x}_n(k)\end{bmatrix}.$$

Since all observers are identical, it holds that for $i, j \in \{1,\cdots,n\}$, $\hat{x}_i(k) = \hat{x}_j(k) = \hat{x}(k)$, which leads to $\bar{r}(k) = y_l(k) - C_l\hat{x}(k)$. Thus, (34) can be proved as a distributed realization of (6).

After the estimation of $x(k)$, we directly compute the evaluation function $J_i(k)$ at node i by

$$\begin{aligned} j_i(k) =& \bar{r}^\top(k)R^{-1}\bar{r}(k) = I_6\bar{r}(k) - (I_4\bar{r}(k))^\top\Omega^{-1}I_4\bar{r}(k)\\ =& \sum_{i=1}^{n}I_{6,i}\bar{r}_i(k) - (\sum_{i=1}^{n}I_{4,i}\bar{r}_i(k))^\top\Omega^{-1}(\sum_{i=1}^{n}I_{4,i}\bar{r}_i(k)),\end{aligned} \tag{35}$$

$$J_i(k) = \left(\frac{1}{m}\sum_{l=0}^{m-1}j_i(k+l)\right)^{1/2}, \tag{36}$$

where

$$I_6 = \bar{r}^\top\Sigma^{-1} = [I_{6,1}\ \cdots\ I_{6,n}]$$

and can be identified by running distributed iteration learning online.

It is clear that in (34), (35) and (36), unknown items are $L\bar{r}(k)$, $I_4\bar{r}(k)$ and $I_6\bar{r}(k)$. Since they have the same structure as $Z^\top\Phi$, we use average consensus to calculate them online. Now, we can run (34) and (35). And the distributed online implementation is achieved in a distributed way. However, in online implementation, if there is a strict limitation on execution time, the distributed FD algorithm may cause inaccuracy.

Finally, we summarize the overall algorithm into the following table.

Offline Training

Step 1	Save A_l, E_l, n, C_i, F_i, J_{th}, c_i and D_g at each node
Step 2	Calculate λ using **Algorithm 1**
Step 3	Calculate W using (31)
Step 4	Calculate I_4 and I_5 using distributed iteration learning
Step 5	Calculate I_1, I_2 and I_3 using average consensus
Step 6	Calculate X using (32), Ω and L

Online Training

Step 1	Obtain $y_i(k)$ and compute $\bar{r}_i(k)$ at each node
Step 2	Calculate I_6 using distributed iteration learning
Step 3	Calculate $L\bar{r}(k)$, $I_4\bar{r}(k)$ and $I_6\bar{r}(k)$ using average consensus
Step 4	Calculate $J_i(k)$ using (35) and (36)
Step 5	Update $\hat{x}_i(k)$ using (34)
Step 6	Make decision using (16)
Step 7	Repeat **Step 1** – **Step 6**

4. NUMERICAL EXAMPLE

Consider a dynamic model (1) with $x(k) \in \mathcal{R}^{18}$, $d_l(k) \in \mathcal{R}^{20}$, and the sensor network (2) with $n = 20$ and $y_i(k) \in \mathcal{R}^1$, $i \in \{1,\ldots,20\}$. Due to the reason that A, E_l, C_i, F_i and c_i are too complicated, they are not introduced here.

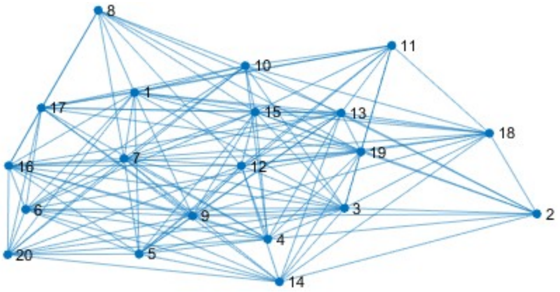

Fig. 1. Communication topology of sensor network

Further, the communication topology of sensor network is shown in Fig. 1. And according to Fig. 1, we have the diameter of the graph is $D_g = 2$. In addition, we set

$$J_i(k) = \left(\frac{1}{5} \sum_{l=0}^{4} j_i(k+l) \right)^{1/2}, \ J_{th} = 11.9193,$$

the sampling time $T = 0.02$s , and an additive step fault with amplitude 10 happening at 6s. Simulation results of the evaluation functions for sensor node 1-20 are shown as $J(1)$-$J(20)$ in Fig. 2-5 respectively. The figures indicate that our proposed distributed FD scheme can achieve the similar result when compared with centralized approach, which is plotted as $J(0)$ in each figure.

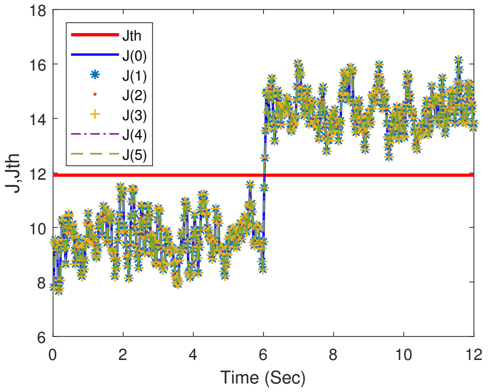

Fig. 2. Evaluation functions of sensor nodes 1-5

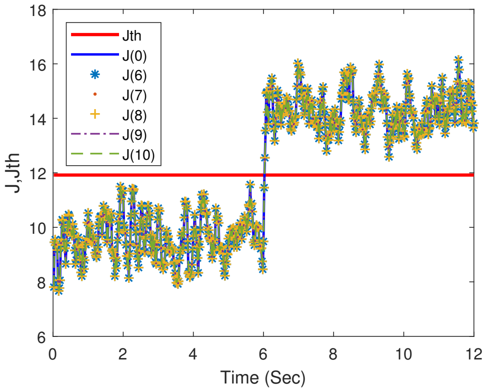

Fig. 3. Evaluation functions of sensor nodes 6-10

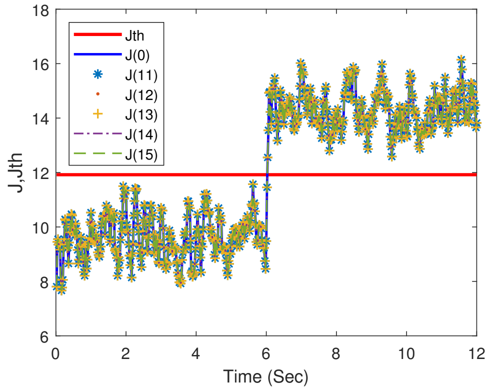

Fig. 4. Evaluation functions of sensor nodes 11-15

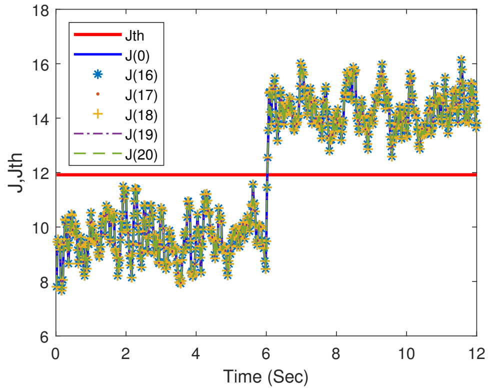

Fig. 5. Evaluation functions of sensor nodes 16-20

However, with the limited online execution time, both online distributed iteration learning and average consensus may not converge to the accuracy value, which may lead to inaccuracy of the distributed online implementation as shown in Fig. 6, which is a local enlarged figure of Fig. 2.

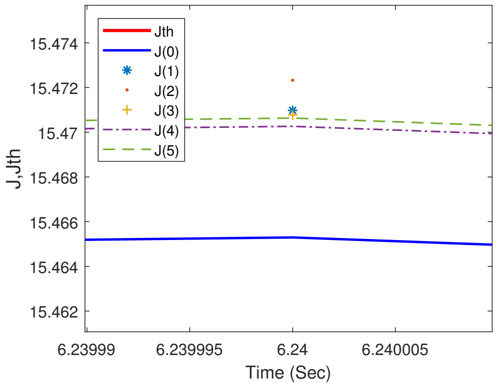

Fig. 6. Comparison of evaluation functions of nodes 1-5

5. CONCLUSION

In this paper, a novel distributed FD scheme was introduced for large-scale systems influenced by deterministic disturbances using sensor networks. Theoretical analysis, which is based on centralized optimal FD scheme, distributed iteration learning and average consensus, and the simulation results show that the proposed scheme can detect fault effectively when compared with the centralized approach. We would like to remark that, offline training and online implementation are all realized in a distributed way, which means that the sensor node can only use its local and neighbors' information for data fusion and corresponding purpose. Possible future research may include the design scheme of general distributed observer, the corresponding stability issues and the data-driven realization of distributed FD scheme.

APPENDIX

A. Proof of Proposition 1

If Σ is symmetric and positive definite matrix with all elements real numbers, we have $\lambda_{max}(\Sigma) = \|\Sigma\|_2$. And we can do singular value decomposition to obtain

$$\Sigma = U\Omega U^\top,$$

where U is orthogonal matrix, which means $U^{-1} = U^\top$, and Ω is a diagonal matrix with diagonal elements being the eigenvalues of Σ. Further, we have

$$\begin{aligned} I - \lambda\Sigma &= U(I - \lambda\Omega)U^\top \\ &= U \begin{bmatrix} 1-\lambda\lambda_1 & & \\ & 1-\lambda\lambda_2 & \\ & & \ddots \end{bmatrix} U^\top, \end{aligned} \tag{37}$$

where λ_i denotes the eigenvalue of Σ. Because Σ is positive definite, we have $\lambda_i > 0$. In order to ensure that $I - \lambda\Sigma$ is Schur matrix, from (37), it is clear that

$$0 < \lambda < \frac{2}{\lambda_{max}(\Sigma)}. \tag{38}$$

Moreover, according to (Golub and van Loan, 2013),

$$\lambda_{max}(\Sigma) = \|\Sigma\|_2 \leqslant \|\Sigma\|_\infty. \tag{39}$$

If we combine (38) and (39), it is evident that Proposition 1 is proved.

B. Proof of Proposition 2

Partition Σ into n rows.

$$\Sigma = \begin{bmatrix} \Sigma_1 \\ \vdots \\ \Sigma_n \end{bmatrix} = \begin{bmatrix} F_1F_1^\top & \cdots & F_1F_n^\top \\ \vdots & \ddots & \vdots \\ F_nF_1^\top & \cdots & F_nF_n^\top \end{bmatrix} \tag{40}$$

With the definition of ∞-norm, we have

$$\|\Sigma\|_\infty = \max_{1\leqslant i\leqslant n} \|\Sigma_i\|_\infty \tag{41}$$

In **Algorithm 1**, we know $\|\Sigma_i\|_\infty = Q_i$, and after D_g-th iteration, it is clear that each node obtain the information of $\max\limits_{1\leqslant i\leqslant n} Q_i$, then Proposition 2 is proved.

REFERENCES

Boem, F., Ferrari, R.M., Keliris, C., Parisini, T., and Polycarpou, M.M. (2016). A distributed networked approach for fault detection of large-scale systems. *IEEE Transactions on Automatic Control*, 62(1), 18–33.

Chen, Z., Cao, Y., Ding, S.X., Zhang, K., Koenings, T., Peng, T., Yang, C., and Gui, W. (2019). A distributed canonical correlation analysis-based fault detection method for plant-wide process monitoring. *IEEE Transactions on Industrial Informatics*, 15(5), 2710–2720.

Ding, S.X. (2008). *Model-based fault diagnosis techniques: design schemes, algorithms, and tools.* Springer Science & Business Media.

Ding, S.X. (2014). *Data-driven design of fault diagnosis and fault-tolerant control systems.* Springer.

Ding, S. (2018). Optimal fault detection and estimation: a unified scheme and least squares solutions. *IFAC-PapersOnLine*, 51(24), 465–472.

Gao, Z., Cecati, C., and Ding, S.X. (2015). A survey of fault diagnosis and fault-tolerant techniques—part i: Fault diagnosis with model-based and signal-based approaches. *IEEE Transactions on Industrial Electronics*, 62(6), 3757–3767.

Garin, F. and Schenato, L. (2010). A survey on distributed estimation and control applications using linear consensus algorithms. In *Networked control systems*, 75–107. Springer.

Golub, G.H. and van Loan, C.F. (2013). *Matrix Computations.* JHU Press, fourth edition.

Jiang, Q., Ding, S.X., Wang, Y., and Yan, X. (2017). Data-driven distributed local fault detection for large-scale processes based on the ga-regularized canonical correlation analysis. *IEEE Transactions on Industrial Electronics*, 64(10), 8148–8157.

Jiang, Q., Yan, X., and Huang, B. (2015). Performance-driven distributed pca process monitoring based on fault-relevant variable selection and bayesian inference. *IEEE Transactions on Industrial Electronics*, 63(1), 377–386.

Komaroff, N. (1994). Iterative matrix bounds and computational solutions to the discrete algebraic riccati equation. *IEEE Transactions on Automatic Control*, 39(8), 1676–1678.

Olfati-Saber, R. (2005). Distributed kalman filter with embedded consensus filters. In *Proceedings of the 44th IEEE Conference on Decision and Control*, 8179–8184. IEEE.

Xiao, L. and Boyd, S. (2004). Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1), 65–78.

Xiao, L., Boyd, S., and Kim, S.J. (2007). Distributed average consensus with least-mean-square deviation. *Journal of parallel and distributed computing*, 67(1), 33–46.

Zhang, Z.H. and Yang, G.H. (2018). Distributed fault detection and isolation for multiagent systems: An interval observer approach. *IEEE Transactions on Systems, Man, and Cybernetics: Systems.*