

Reduced-Order Distributed Fault Diagnosis for Large-Scale Nonlinear Stochastic Systems

Elaheh Noursadeghi

Autonomous Robotic System Laboratory,
Department of Mechanical Engineering,
University of Massachusetts Lowell,
Lowell, MA 01854
e-mail: Elaheh_Noursadeghi@student.uml.edu

Ioannis A. Raptis

Autonomous Robotic System Laboratory,
Department of Mechanical Engineering,
University of Massachusetts Lowell,
Lowell, MA 01854
e-mail: ioannis_Raptis@uml.edu

This paper deals with the distributed fault detection and isolation problem of uncertain, nonlinear large-scale systems. The proposed method targets applications where the computation requirements of a full-order failure-sensitive filter would be prohibitively demanding. The original process is subdivided into low-order interconnected subsystems with, possibly, overlapping states. A network of diagnostic units is deployed to monitor, in a distributed manner, the low-order subsystems. Each diagnostic unit has access to a local and noisy measurement of its assigned subsystem's state, and to processed statistical information from its neighboring nodes. The diagnostic algorithm outputs a filtered estimate of the system's state and a measure of statistical confidence for every fault mode. The layout of the distributed failure-sensitive filter achieves significant overall complexity reduction and design flexibility in both the computational and communication requirements of the monitoring network. Simulation results demonstrate the efficiency of the proposed approach. [DOI: 10.1115/1.4037839]

1 Introduction

The majority of contemporary industrial and commercial control systems are composed of a large number of spatially distributed feedback modules with heterogeneous sensors, actuators, and controllers that exchange information over a band-limited communication network that is embedded within the system. These large-scale systems are characterized by high-dimensional state-spaces and nonlinear dynamics. Typical applications are water distribution networks, power grids, automated highway systems, swarms of unmanned aerial vehicles, and environmental control systems, just to name a few. Large-scale systems are much more vulnerable to faults since the effects of a single malfunction to an individual part may rapidly diffuse to the entire system due to the interconnection of the various subcomponents.

Availability, dependability, and resiliency are becoming major design goals for large-scale technological systems due to stringent economic, ecological, and safety demands. These attributes are of major importance, primarily for safety-critical systems, e.g., airplanes, automobiles, and nuclear reactors, since they ensure public safety. Therefore, there is a growing need for reliable real-time monitoring and supervision especially in the case of safety-critical systems. Fault diagnosis (FD) describes the dual objective of detecting the occurrence of a fault (detection) and identifying it (identification or isolation). A timely diagnosis of a fault mode may improve the system's availability and maintainability by avoiding down-times, breakdowns, and catastrophic failures rates.

Research in the field of FD has attracted significant attention since the beginning of the 1970s. The significant majority of existing FD methods [1–6] have a centralized architecture in the sense that the sensor measurements and the diagnostic algorithm are collected and executed by a singleton processing unit. Centralized FD is considered a matured field that has established reliable solutions to many engineering applications. However, the applicability of this traditional approach is limited to concentrated low-order systems.

Modern processes involve high-dimensional state-spaces as well as highly nonlinear dynamics. In the case of large-scale and spatially distributed systems, centralized FD becomes ill-suited. Every monitoring system has certain limitations in terms of

computational power and communication bandwidth. When the dimensionality and complexity of the system increases, it is likely that these limitations will not be satisfied by a centralized configuration. The online monitoring of a high-dimensional system would require extensive computations from the central processing unit. Processes with geographically remote subcomponents necessitate long distance and energy demanding broadcasts or complex multi-hop routing protocols to transmit information to the central fusion center. In both cases, a centralized architecture exhibits poor scalability.

In the literature, the majority of distributed fault detection and identification methods are developed for discrete-event systems and for multiprocessor computing applications [7–9]. A growing interest in the development of distributed fault detection algorithms has also been reported by the wireless sensor networks community [10]. In this work our attention is spotlighted to model-based FD methods for dynamic systems that utilize a mathematical model of the process. A rudimentary classification of existing distributed FD model-based methodologies can take place based on the data type that is exchanged between the nodes of the diagnostic system. The diagnostic nodes (DNs) can exchange: raw measurements of the interconnected states [11–13], state estimates [14,15], or fault signatures [16]. The most prominent work on distributed, observed-based FD for nonlinear dynamic systems has been reported by Ferrari et al. [11] and Boem et al. [17]. The authors apply overlapping decomposition techniques to subdivide the monolithic process to a set of reduced order subsystems. Each subsystem is monitored by a local nonlinear observer. Seminal work in distributed estimation-based FD using Kalman filtering is reported in Ref. [18]. The algorithm is based on the distributed version of Kalman filter (KF) established by Olfati-Saber [19,20]. The KF is restricted to linear system, while linearization (extended KF) leads to high false alarms rates [21].

Foundational work on estimation-based FD for nonlinear systems that employ large number of correlated sensors is introduced in Refs. [22,23]. The author combines a distributed particle filtering algorithm for state estimation with fused hypothesis testing through likelihood tests, to determine the occurrence of fault modes. The proposed method is mainly geared toward relative low-order systems monitored by a high number of interconnected sensors. The layout of the algorithm does not accommodate subdivisions of the original process. The applied classical likelihood tests require a bank of estimators equal to the number of fault modes. Such replication of the state is unsuitable for large-scale

Contributed by the Dynamic Systems Division of ASME for publication in the JOURNAL OF DYNAMIC SYSTEMS, MEASUREMENT, AND CONTROL. Manuscript received February 27, 2017; final manuscript received August 24, 2017; published online December 19, 2017. Assoc. Editor: Jongeun Choi.

systems. A full-order distributed failure-sensitive filter has also been introduced by Noursadeghi and Raptis [24,25]. In this scheme, a detection network is assigned to monitor the entire state of the monolithic system using only local measurements. Similarly to the work of Cheng, the authors did not consider any form of subdivision in the dynamics of the original process. Instead, the algorithm introduced in Refs. [24,25] provides an estimate of the entire state of the monolithic system.

The particle filter (PF) is an ideal estimator for fault diagnosis since it avoids linearity and Gaussian noise assumptions. A comprehensive analysis on distributed PF algorithms is given by Hlinka [26,27] and Mohammadi [28,29]. A distributed PF scheme for FD diagnosis that accounts for system decomposition is reported in Ref. [30]. The authors propose a hybrid modeling approach where every potential fault is treated as a system mode. This approach assumes that the transition probability between the fault modes is known a priori. This probabilistic information may not be available in most real-life applications.

In this work, we present a distributed, model-based and sequential fault diagnosis methodology for large-scale, stochastic nonlinear systems that are subject to multiple fault modes. This approach targets systems where the state dimension is significantly large (10^2 states and higher). A distributed version of the particle filtering method will serve as the foundation of the derived diagnostic algorithms. We introduce a reduced-order fault diagnostic algorithm that allows the subdivision of the original process dynamics to low-order interconnected subsystems with state overlap. A DN is assigned to monitor every partition of the monolithic system and triggers alarm indicators based on its local observations and information exchange between neighboring units. Each local failure sensitive filter outputs an estimate of the subsystem's state vector and the probabilities of failure of the local fault modes.

Our reduced-order FD technique achieves a dramatic decrease to the computational complexity of the original problem and provides significant design flexibility to the layout of the algorithm. The PF is an ideal estimator since it eliminates complex Lyapunov arguments that are required by the observed-based methods to guarantee convergence. A binary update rule is used to repopulate the particles and estimate the system modes without the need for transition probabilities. The failure sensitive filter can simultaneously detect and identify faults without the need for a bank of estimators. The proposed algorithm takes advantage of the decentralized architecture and computational strength of modern embedded systems such as wireless sensor networks and multicore processors.

This paper is organized as follows: A brief description of the PF algorithm is presented in Sec. 2. The synthesis of a centralized PF fault diagnosis algorithm is outlined in Sec. 3. The centralized algorithm serves as a benchmark framework for its distributed counterpart. The reduced-order distributed version of the failure sensitive filter is presented in Sec. 4. The performance of the proposed methodology is evaluated in Sec. 5 via numerical simulations. Finally, concluding remarks are given in Sec. 6.

2 Centralized Particle Filtering

The filtering problem is formulated based on the discrete time state-space approach. The purpose is to estimate the state of the system by using a sequence of noisy measurements. Consider a time-dependent, state vector $x(k) \in \mathbb{R}^{n_x}$, where $k \in \mathbb{Z}^+$ is the time index. The state-transition model of the state $x(k)$ is defined according to

$$x(k) = f(x(k-1), v(k-1)) \quad (1)$$

where $f(k) : \mathbb{R}^{n_x \times n_v} \rightarrow \mathbb{R}^{n_x}$ is a known, nonlinear function, and $v(k) \in \mathbb{R}^{n_v}$ stands for system's process noise. At time step k , the measurement equation of the state $x(k)$ is expressed by

$$z(k) = h(x(k)) + \omega(k) \quad (2)$$

where $z(k) \in \mathbb{R}^{n_z}$ represents the measurement vector, $h(k) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_z}$ is a known nonlinear function, and $\omega(k) \in \mathbb{R}^{n_w}$ stands for the measurement noise. It is assumed that both the process noise $v(k)$ and measurement noise $\omega(k)$ are white and independent with known probability density functions (pdf). From a Bayesian perspective, the objective is to recursively quantify some degree of belief in the state $x(k)$, given the measurement data $z(1:k)$ up to time k . The belief is expressed by the calculation of the *posterior* pdf $p(x(k)|z(1:k))$. The calculation of the posterior density $p(x(k)|z(1:k))$ allows the computation of various measures of the state $x(k)$, such as the minimum mean square error

$$\hat{x}(k) \triangleq E\{x(k)|z(1:k)\} = \int x(k) \cdot p(x(k)|z(1:k)) dx(k) \quad (3)$$

The PF is a sequential Monte Carlo method that uses a finite set of "particles" to represent probability density functions [31]. The basic idea is to represent the non-Gaussian posterior pdf $p(x(k)|z(k))$ by a set of randomly drawn particles $x^i(k)$ and corresponding weights $w^i(k)$. Using set of particles and the weights $\{x^i(k), w^i(k)\}_{i=1}^{N_s}$, the posterior can have the following discrete approximation:

$$\hat{p}(x(k)|z(k)) \approx \sum_{i=1}^{N_s} w^i(k) \delta(x(k) - x^i(k)) \quad (4)$$

where $\delta(\cdot)$ denotes the multivariate delta Dirac function. As the number of particles becomes very large, the sequential Monte Carlo representation is closer to the analytical description of the posterior. Using the representation in Eq. (4), one can obtain various estimates of $x(k)$. For example, the minimum mean square error estimate is approximated as

$$\begin{aligned} \hat{x}(k) &= \sum_{i=1}^{N_s} w^i(k) \int x(k) \delta(x(k) - x^i(k)) dx(k) \\ &= \sum_{i=1}^{N_s} w^i(k) x^i(k) \end{aligned} \quad (5)$$

Different variations of the PF algorithm exist depending on the choice of the importance density function and the resampling step. The most standard form of the PF algorithm is the sequential importance resampling filter (SIR). The SIR filter forms the foundation for some well-known PFs including the bootstrap filter [32], the auxiliary PF [33], and the regularized PF [33]. These PFs are derived using a suboptimal choice of the proposal pdf $q(x(k)|z(k-1), z(k))$.

The most frequently used and easiest to implement SIR algorithm is the bootstrap filter that is employed during this work. In this filter, the particles are updated using the state-transition density function $p(x(k)|x(k-1))$ as the importance pdf. In this case, the weight update equation simplifies to

$$w^i(k) \propto w^i(k-1) p(z(k)|x^i(k)) \quad (6)$$

The measurement noise ω is considered as a Gaussian distribution with zero mean and covariance matrix Σ_ω , the likelihood function for each particle is calculated by

$$p(z(k)|x^i(k)) \sim \mathcal{N}(e^i(k), 0, \Sigma_\omega) \quad (7)$$

where $\mathcal{N}(e^i(k), 0, \Sigma_\omega)$ is the normal distribution with zero mean and covariance matrix Σ_ω evaluated at the points $e^i(k) = z(k) - h(x^i(k))$, where $e^i(k)$ is the prediction error of the i th particle. For a detailed description of various PF algorithms and resampling techniques, the reader is referred to Ref. [33]. The

Table 1 Pseudocode of the BOOTSTRAP PF algorithm [31]

```

function BOOTSTRAP PF
Inputs :  $x^i(k-1), w^i(k-1), z(k)$ 
Outputs :  $x^i(k), w^i(k)$ 
Required : RESAMPLE

1: for  $i = 1 : N_s$  do
2:    $x^i(k) = f(x^i(k-1), v^i(k-1))$   $\triangleright$  Particles update
3:    $w^i(k) = w^i(k-1) \cdot p(z(k)|x^i(k))$   $\triangleright$  Weights update
4: end for
5:  $w^i(k) = \frac{w^i(k)}{\sum_{j=1}^{N_s} w^j(k)}$   $\triangleright$  Weight normalization
6:  $\{[x^i(k), w^i(k)]\}_{i=1}^{N_s} = RESAMPLE[\{x^i(k), w^i(k)\}_{i=1}^{N_s}]$   $\triangleright$  Resampling
7:  $\hat{x}(k) = \sum_{i=1}^{N_s} w^i(k) x^i(k)$   $\triangleright$  Minimum mean-square error

```

Table 2 Pseudocode of the RESAMPLING algorithm [33]

```

function RESAMPLE
Inputs :  $x^i(k), w^i(k)$ 
Outputs :  $x^{j*}(k), w^j(k)$ 
1:  $c(1) = 0$             $\triangleright$  Initialize the cumulative distribution function (CDF)
2: for  $i = 2 : N_s$  do
3:    $c(i) = c(i-1) + w^i(k)$             $\triangleright$  Construct CDF
4: end for
5:  $i = 1$             $\triangleright$  Start at the bottom of the CDF
6:  $u(1) = U[0, N_s^{-1}]$             $\triangleright$  Draw a starting point
7: for  $j = 1 : N_s$  do
8:    $u(j) = u(1) + N_s^{-1}(j-1)$             $\triangleright$  Move along the CDF
9:   while  $u(j) > c(i)$  do
10:     $i = i + 1$ 
11:   end while
12:    $x^{j*}(k) = x^i(k)$             $\triangleright$  Assign samples
13:    $w^j(k) = N_s^{-1}$             $\triangleright$  Assign weights
14: end for

```

pseudocode of the bootstrap filter and resampling are provided in Tables 1 and 2, respectively. The block diagram of the bootstrap algorithm is shown in Fig. 1.

3 Centralized Particle Filtering Fault Diagnosis

This work extends the methodology introduced in Ref. [34] from one-dimensional fault-growth models to dynamic state-space systems of nonlinear processes introducing the centralized particle filtering fault diagnosis (CPFFD) algorithm. The CPFFD algorithms generate two outputs. The first is the system's state estimate from a sequence of noise infested measurements. The second output is a statistical characterization for the occurrence of each fault mode that can trigger fault alarms.

Consider the uncertain, nonlinear, and discrete-time dynamic system S described by the following state-space model:

$$S : \quad \sum_{j=1}^M \beta(k - k_0^j) \cdot g^j(x(k-1), u(k-1)) + v(k-1) \quad (8)$$

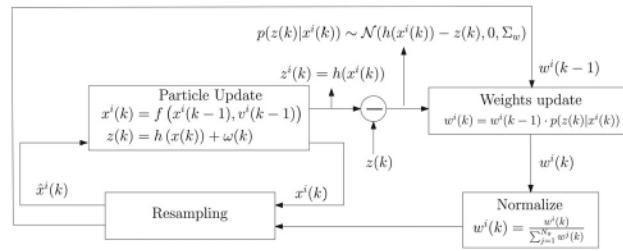


Fig. 1 Block diagram of the bootstrap PF

Table 3 Pseudocode of the CPFFD algorithm

```

function CPFFD
Inputs :  $\mathcal{X}^i(k-1), w^i(k-1), u^i(k-1), \mathcal{Z}(k)$ 
Outputs :  $\mathcal{X}^i(k), w^i(k), \{\hat{b}_2^j(k)\}_{j=1}^M$ 
Required : RESAMPLE

1: for  $i = 1 : N_s$  do
2:    $\mathcal{X}^i(k) = \mathcal{F}(\mathcal{X}^i(k-1), u^i(k-1), \mathcal{V}(k-1))$             $\triangleright$  Particles update
3:    $w^i(k) = w^i(k-1) \cdot p(\mathcal{Z}(k) | \mathcal{X}^i(k))$             $\triangleright$  Weights update
4: end for
5:  $w^i(k) = \frac{w^i(k)}{\sum_{j=1}^{N_s} w^j(k)}$             $\triangleright$  Weight normalization
6:  $[\{\mathcal{X}^i(k), w^i(k)\}_{i=1}^{N_s}] = \text{RESAMPLE}[\{\mathcal{X}^i(k), w^i(k)\}_{i=1}^{N_s}]$             $\triangleright$  Resampling
7:  $\hat{\mathcal{X}}(k) = \sum_{i=1}^{N_s} w^i(k) \mathcal{X}^i(k)$             $\triangleright$  Minimum mean-square error
8: for  $j = 1 : M$  do
9:    $\hat{b}_2^j(k) = E[b_2^j(k) | \mathcal{Z}(k)]$             $\triangleright$  Probabilities of failure
10: end for

```

where the terms $x(k) \in \mathbb{R}^{n_x}$, $u(k) \in \mathbb{R}^{n_u}$, and $z(k) \in \mathbb{R}^{n_z}$ refer to the state, input, and measurement vector, respectively; $f(k) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$, and $h(k) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_z}$ denote the known nonlinear functions of the system's healthy dynamics and measurement model, while $v(k) \in \mathbb{R}^{n_x}$ and $\omega(k) \in \mathbb{R}^{n_z}$ stand for the process and measurement noise sequences, respectively.

The monolithic system's healthy dynamics are subjected to M potential fault modes described by the nonlinear functions $\{g^j(x(k), u(k))\}_{j=1}^M$ with $g^j(k) : \mathbb{R}^{n_x \times n_u} \rightarrow \mathbb{R}^{n_x}$. The term $\beta(k - k_0^j)$ is a scalar function representing the time profile of the fault mode j occurring at some unknown time k_0^j . We can consider both *abrupt* (step-like) or *incipient* (exponential-like) fault modes, defined as

$$\beta(k - k_0^j) = \begin{cases} 0 & k < k_0^j \\ \underbrace{1}_{\text{abrupt}} \text{ or } \underbrace{1 - c^{-(k - k_0^j)}}_{\text{incipient}}, c > 1 & k \geq k_0^j \end{cases} \quad (9)$$

It is assumed that the system is initiated from the healthy mode ($\beta(\cdot) = 0$ at $k = 0$). Due to the random occurrence of the possible faults, the monolithic system may be viewed as a hidden Markov model, where the transition probabilities between the different system modes are unknown.

The proposed failure sensitive filter embeds the dynamics of the monolithic system S given in Eq. (8), as well as a binary variable (for every potential fault), that identifies the changes in the process dynamics expressed by the terms $\beta(k - k_j^i)$. Hence, the binary state vector $b^i(k) = [b_1^i(k) \ b_2^i(k)]^T$, with $b_1^i, b_2^i \in \{0, 1\}$

and $j = 1, \dots, M$, is introduced to estimate the occurrence of each fault mode. More specifically, $b_1^j(k) = 1$ indicates that the absence of failure mode j , while $b_2^j(k) = 1$ denotes that the fault mode j is detected to the system. The continuous-valued states are coupled with the discrete-valued binary fault occurrence estimates resulting in a hybrid model.

The operating condition of the system (normal or faulty), as well as the detection and isolation of the faults, is determined by employing a particle filtering scheme for the statistical characterization of both the binary and continuous-valued states, as new measurements are received. Hence, the state vector that is used by the PF algorithm is the combination of the continuous state vector and the binary state vector as $\mathcal{X}^T(k) = [(x^c(k))^T \ (b^1(k))^T \ \dots \ (b^M(k))^T]^T \in \mathbb{R}^{n_c+2M}$, where $x^c(k) \triangleq x(k)$. Therefore, the state-transition dynamics of \mathcal{X} are described by

$$\begin{aligned} x^c(k) &= \underbrace{f(x^c(k-1), u(k-1)) + \dots}_{\text{evolution of continuous states}} \\ S^f : \quad \sum_{j=1}^M g^j(x^c(k-1), u(k-1)) \cdot b_2^j(k-1) + \tilde{v}(k-1) \\ b^j(k) &= \underbrace{\Phi(b^j(k-1) + n^j(k-1))}_{\text{evolution of binary states}}, \quad j = 1, \dots, M \\ z(k) &= \underbrace{h(x^c(k)) + \tilde{\omega}(k)}_{\text{measurement}} \end{aligned} \quad (10)$$

where $\tilde{v}(k) \in \mathbb{R}^{n_c}$ and $\tilde{\omega}(k) \in \mathbb{R}^{n_z}$ are approximations of the failure sensitive filter's process and measurement noise, respectively. These noise sequences should be as close as possible to the actual ones ($v(k)$ and $\omega(k)$). The nonlinear function $\Phi: \mathbb{R}^2 \rightarrow \{[0 \ 1]^T, [1 \ 0]^T\}$, represents the evolution of the binary states driven by the identically independent distributed (i.i.d) uniform white noise $n^j(k)$. The function $\Phi(\cdot)$ is defined such that the previous state $b^j(k-1)$ is randomly excited at each time step by $n^j(k-1)$. This random vector of \mathbb{R}^2 is assigned to one of the binary states (normal/faulty operating condition) based on the distance metric of the perturbed vector $b^j(k-1) + n^j(k-1)$ to the coordinates $[0 \ 1]^T$ and $[1 \ 0]^T$.

By using this technique, when one of the fault modes occurs, the weights will gradually converge the corresponding binary variable $b_2^j(k)$ to one ($b_2^j(k) \rightarrow 1$). This is due to the fact that the likelihood of the measurements will diminish the weights of the particles associated with the healthy condition. This way, the occurrence of the each fault is estimated exclusively through the measurement data and the process model, without the knowledge of the fault modes transition probabilities. The function $\Phi(\cdot)$ serves as "likelihood feedback" that drives the sample population of the binary states. A choice of $\Phi(\cdot)$ that has been successfully used in Refs. [34–36] is

$$\Phi(x) = \begin{cases} e_1 & \text{if } \|x - e_1\| \leq \|x - e_2\| \\ e_2 & \text{else} \end{cases} \quad (11)$$

where $e_1 = [1; 0]^T$ and $e_2 = [0; 1]^T$. The state model of the CPFFD algorithm can be written in a more compact form as

$$\begin{aligned} S^f : \quad \mathcal{X}(k) &= \mathcal{F}(\mathcal{X}(k-1), u(k-1), \mathcal{V}(k-1)) \\ \mathcal{Z}(k) &= \mathcal{H}(\mathcal{X}(k)) + \tilde{\omega}(k) \end{aligned}$$

where $\mathcal{Z}(k) = z(k)$, $\mathcal{V}(k) = [\tilde{v}(k) \ n^1(k) \ \dots \ n^M(k)]$, and $\mathcal{F}(\cdot)$, $\mathcal{H}(\cdot)$ are nonlinear functions of appropriate dimensions and structure. The aforementioned definition will be used to ease the notation in subsequent parts of the analysis. The outputs of the CPFFD module are the estimation of the system's state vector and the probabilities of failure of each fault mode. These probabilities are the expectations of the binary states $\hat{b}_2^j(k) = E[b_2^j(k) | \mathcal{Z}(k)]$. This measure is used to trigger alarm indicators if its value exceeds a

certain threshold $\alpha \in (0; 1)$ that marks the probability of detection (i.e., $\hat{b}_2^j(k) < \alpha$ indicates that the system is in healthy operating condition). With this layout, two or more different co-existing fault modes can be simultaneously detected. The pseudocode of the CPFFD algorithm is given in Table 3.

The probability of failure is a much more computationally attractive measure compared to classical change detection methods such as hypothesis testing. In the context of fault isolation, detection algorithms using hypothesis testing through logarithm likelihood ratio test requires the execution of a bank of estimators that is equal to the fault modes. For large-scale systems, this computational load is prohibited. The proposed CPFFD algorithm is significantly more efficient, since it increases the dynamics of the detector by only M binary state vectors.

4 Distributed Particle Filtering Fault Diagnosis

The CPFFD algorithm described in Sec. 3 is not scalable or robust to complex large-scale dynamical systems that employ scattered measurement sensors over large geographical regions. For high-dimensional large-scale systems, this methodology becomes impractical due to limitations in the observation range of sensors, communication bandwidth, and computation power of the centralized computing node.

In this section, we present a reduced-order distributed particle filtering fault diagnosis (DPFFD) algorithm for large-scale nonlinear systems. The original diagnostic problem is subdivided to a number of lower-order interconnected fault sensitive filters. With this technique, each low-order filter can balance its computation power requirements and volume of data transfers. Similar to Ref. [37], we take into account subdivisions with state overlap. The states that are common between two or more subsystems are referred to as shared states. Shared states between subsystems appear when state variables are mutually monitored by sensors that correspond to different subsystems.

Here, we briefly illustrate the three most characteristic types of decomposition based on a similar description given in Ref. [37]. The most communication intensive decomposition involves nonoverlapping subsystems of order one (Fig. 2(a)). This fragmenting is the most computationally effective, however, most likely the communication limitations will be reached. On the contrary, the decomposition depicted in Fig. 2(b) provides a balanced compromise between computational labor and communication broadcasts. It is important to note that there exists a trade-off between computation power and communication capacity for the nodes of the network. The third case (Fig. 2(c)) is similar to the previous scenario with the difference that there is overlap between the subsystems. In principal, overlapping dynamics increase both the complexity and the communication requirements of the overall design. This additional complexity overhead is due to the fusion of the common measurements and shared state estimates between the nodes. The overlap can increase the fidelity of states and measurements that are exposed to higher uncertainty, since they are monitored by more than one sensor. This fact can justify the additional effort in terms of complexity and communications that stems from overlapping states. Decomposition techniques are out of the scope of this paper, and the interested readers are referred to Refs. [38,39].

Graph theoretical tools are deployed to represent the dynamical interdependence of the system's states [37]. A *directed graph* or *digraph* provides a pictorial representation of the system's structure [40]. The digraph of system S is defined as the pair $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$, where $\mathcal{V}_s = \chi \cup \mathcal{V} \cup \mathcal{B}$ is the set of vertices consisted of the system states $\chi = \{X_j\}_{j=1, \dots, n_s}$, the noise inputs $\mathcal{V} = \{v_j\}_{j=1, \dots, n_s}$, and the scalar terms $\mathcal{B} = \{\beta_j\}_{j=1, \dots, M}$. The set \mathcal{E}_s represents the oriented edges defined by the ordered pairs $\{\nu_l, \nu_m\}$, where $\nu_l \in \mathcal{V}_s$ and $\nu_m \in \chi$. An oriented edge exists between the state X_l (or v_l and β_l) and state X_m , if the former appears at the dynamic equation of the latter. If an edge exists between vertices ν_l and ν_m , we call them *adjacent* and denote this relationship by $\nu_l \sim \nu_m$. We define the neighborhood $\mathcal{V}_m \subseteq \mathcal{V}_s$ of

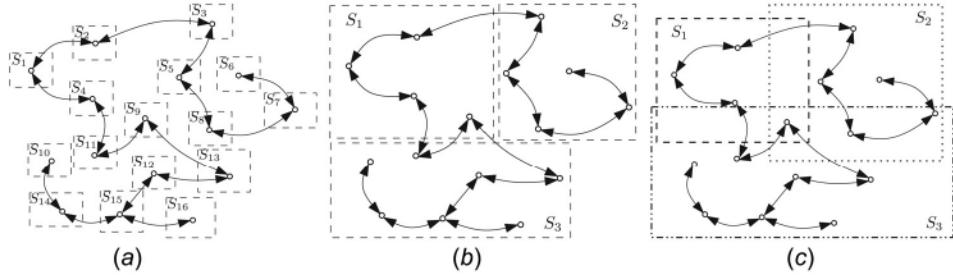


Fig. 2 Characteristic types of system decomposition: (a) low complexity, many interconnections, (b) balanced complexity/communications with no shared states, and (c) balanced complexity/communications, with shared states. This figure is based on a similar one in Ref. [37].

the vertex $X_m \in \chi$ as the set $\mathcal{V}_m = \{X_l \in \chi | \{X_l, X_m\} \in \mathcal{E}_s\}$ of all adjacent states to X_m . The digraph \mathcal{G}_s is also referred to as *structural graph* of the system S .

From a graph theoretical perspective, each subsystem S_I of the monolithic process is represented by a cut-point set of vertices $\mathcal{V}_s^{(I)}$, where $\mathcal{V}_s^{(I)} \subseteq \chi$. Each cut-point set includes states that are observed locally by sensors of its corresponding subsystem. The components of χ that belong to the cut-point set $\mathcal{V}_s^{(I)}$ comprise the local states of subsystem S_I . States from subsystems with departing directed edges that enter the vertices of a cut-point set determine the *interconnection variables* or *forcing terms*.

Following the previous definitions [41], every subsystem S_I has a local state vector $x_I \in \mathbb{R}^{n_{x_I}}$, local interconnection variables vector $d_I \in \mathbb{R}^{n_{d_I}}$, and local process noise input vector $v_I \in \mathbb{R}^{n_{v_I}}$. Each of the M fault modes will have their own presence in every subsystem. Let $g_I^j(k)$ denote the appearance of the fault mode J ($J = 0, 1, \dots, M$) at subsystem I ($I = 1, \dots, N$). Based on the earlier definition, the state space model of each subsystem S_I is described by

$$S_I : \begin{aligned} x_I(k) &= f_I(x_I(k-1), d_I(k-1), u_I(k-1)) \\ &+ \sum_{j=1}^M \beta_I^j(k - k_0^j) \cdot g_I^j(x_I(k-1) d_I(k-1), u_I(k-1)) \\ &+ v_I(k-1) \\ z_I(k) &= h_I(x_I(k)) + \omega_I(k) \end{aligned} \quad (12)$$

where $u_I(k) \in \mathbb{R}^{n_{u_I}}$ and $z_I(k) \in \mathbb{R}^{n_{z_I}}$ refer to the control input and measurement vector of subsystem I , respectively. The nonlinear functions $f_I(\cdot, \cdot, \cdot) : \mathbb{R}^{n_{x_I}} \times \mathbb{R}^{n_{d_I}} \times \mathbb{R}^{n_{u_I}} \rightarrow \mathbb{R}^{n_{x_I}}$ and $h_I(\cdot) : \mathbb{R}^{n_{x_I}} \rightarrow \mathbb{R}^{n_{z_I}}$ denote to the local subsystem and measurement

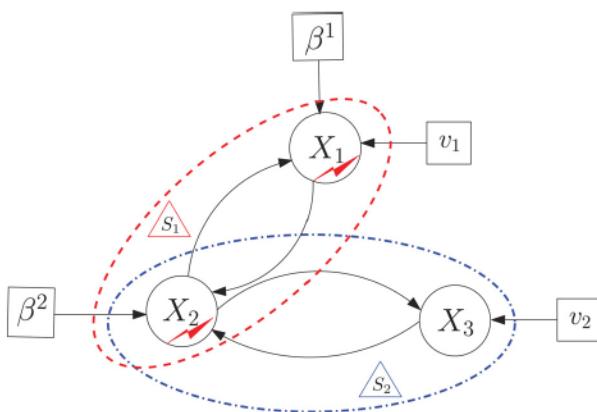


Fig. 3 System digraph of the running example with three states. The circles represent the states, and the squares represent the noise input and scalar variables vectors (fault occurrence terms). The thunderbolt marks illustrate the location of the potential failure modes.

dynamics; while $v_I(k) \in \mathbb{R}^{n_{v_I}}$ and $\omega_I(k) \in \mathbb{R}^{n_{\omega_I}}$ stand for the subsystem and measurement noise, respectively.

Likewise to the centralized approach, the formulation of the reduced order local PF for fault diagnosis will include a vector of binary states to represent the absence or presence of each fault mode. The binary vector of failure mode j at subsystem I is represented by $b_I^j(k) = [b_{1,I}^j(k) \ b_{2,I}^j(k)]^T$ with the local fault function $g_I^j(\cdot)$.

The aforementioned definitions are illustrated with a simple example. Consider a three-dimensional system with the global state vector $x(k) = [X_1(k), X_2(k), X_3(k)]^T$, the noise vector $v(k) = [v_1(k), v_2(k), v_3(k)]^T$, and the set of change step functions $B = \{\beta_1(k), \beta_2(k)\}$. The digraph of this example is shown in Fig. 3. Each sensor set monitors one subsystem. The monolithic system of this example is decomposed into two subsystems represented by the cut-point sets $\mathcal{V}_s^{(1)} = \{X_1, X_2\}$ and $\mathcal{V}_s^{(2)} = \{X_2, X_3\}$. The local dynamic models of these two overlapping subsystems are

$$\begin{aligned} x_1(k) &= f_1(X_1(k-1), X_2(k-1), \underbrace{X_3(k-1)}_{d_1(k-1)}) + \dots \\ S_1 : \quad & \beta_1^1(k - k_0^1) \cdot g_1^1(X_1(k-1)) + \dots \\ & \beta_1^2(k - k_0^2) \cdot g_1^2(X_2(k-1)) + v_1(k-1) \\ z_1(k) &= h_1(\underbrace{X_1(k), X_2(k)}_{x_1(k) = [X_1(k) \ X_2(k)]^T}) + \omega_1(k) \end{aligned} \quad (13)$$

$$\begin{aligned} x_2(k) &= f_2(X_2(k-1), X_3(k-1), \underbrace{X_1(k-1)}_{d_2(k-1)}) + \dots \\ S_2 : \quad & \beta_2^1(k - k_0^1) \cdot g_2^1(X_2(k-1)) + v_2(k-1) \\ z_2(k) &= h_2(\underbrace{X_2(k), X_3(k)}_{x_2(k) = [X_2(k) \ X_3(k)]^T}) + \omega_2(k) \end{aligned} \quad (14)$$

The compact formulation of the local reduced-order failure sensitive filter that incorporates the evolution of the binary states is given by

$$S_I^f : \begin{aligned} \mathcal{X}_I(k) &= \mathcal{F}_I(\mathcal{X}_I(k-1), \mathcal{U}_I(k-1), \mathcal{V}_I(k-1)) \\ \mathcal{Z}_I(k) &= \mathcal{H}_I(\mathcal{X}_I(k)) + \omega_I(k) \end{aligned} \quad (15)$$

where $\mathcal{X}_I = [(x_I)^T \ (b_I^1)^T \ \dots \ (b_I^M)^T]^T \in \mathbb{R}^{n_{x_I} + 2M}$, $\mathcal{U}_I = [(u_I)^T \ (d_I)^T]^T$, $\mathcal{Z}_I = z_I$, $\mathcal{V}_I = [v_I^T \ n_I^T]^T$, and $\mathcal{F}_I(\cdot)$, $\mathcal{H}_I(\cdot)$ are nonlinear functions with appropriate dimensions and structure. The diagnostic algorithm includes the design of one DN for each subsystem S_I . Each DN consists of a processing unit that executes the local PF algorithm. The nodes can measure their own local states and communicate with their neighbors to obtain a processed estimate of their forcing term vector. The layout of the proposed DPFFD algorithm is depicted in Fig. 4. The algorithm can be separated into three main parts.

Particles update: In the first part, each DN executes a local bootstrap PF. For every subsystem, N_s particles are drawn

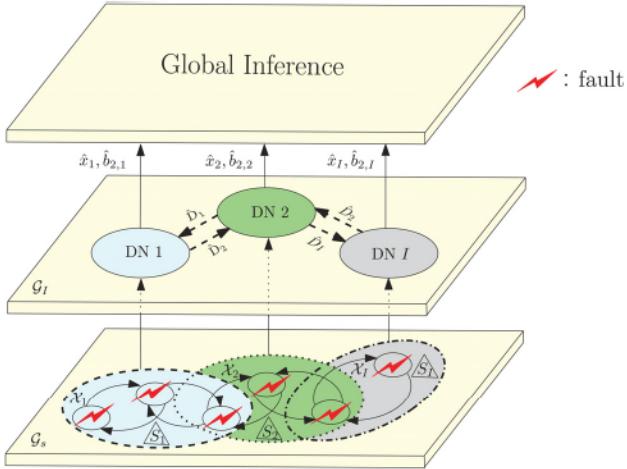


Fig. 4 Schematic of the reduced-order DPFFD algorithm. The thunderbolt markers represent the location of the potential fault modes in the system.

according to the state transition propagation given in Eq. (15). This action requires estimates of the forcing terms \hat{b}_I obtained by the neighbors of DN I . At time $k-1$ all subsystems have already generated an estimate of their own states.

Weights update: In this step, each node uses its local observation and updates its particles' weights based on $p(\mathcal{Z}_I(k)|\mathcal{X}_I^i(k))$. By taking into account the estimates of the neighboring states \hat{d}_I , and by choosing the proposal distribution similar to the local state transition function, the weight update is given by

$$w_I^i(k) \propto w_I^i(k-1)p(\mathcal{Z}_I(k)|\mathcal{X}_I^i(k)) \quad (16)$$

The local PF is concluded after the weight normalization and resampling steps of the bootstrap filter (Sec. 2). The outputs of each DN are an estimation of the subsystems' state vector $\hat{x}_I(k)$, and the probabilities of failure $\hat{b}_{2,I}^j(k) = E[b_{2,I}^j(k)|\mathcal{Z}_I(k)]$; $j = 1, \dots, M$.

Shared states fusion: The last part of the reduced-order DPFFD algorithm involves the fusion of state variables that belong to overlapping cut-point sets. At each time step, the estimates of all DN are collected by a central fusion center that assembles the final global output of the diagnostic network. The data transmitted to the central unit contain only postprocessed information. This is the only centralized processing action that takes place on the fusion center and does not add significant computational overhead to the algorithm. A running average filter is executed between the shared states to calculate a common estimate of their value.

The state estimates $\hat{x}_I \in \mathbb{R}^{n_{x_I}}$ of each DN do not have the same dimensions and cannot be added directly. To this end, the local state matrices $H_I \in \mathbb{R}^{n_x \times n_{x_I}}$ are introduced that convert the local state estimates \hat{x}_I to the vectors \hat{x}_I^g of dimension n_x . The vector \hat{x}_I^g has nonzero entries, equal to the components of \hat{x}_I , at the states that correspond to the elements of the cut-point set \mathcal{V}_s^I . The rows of H_I correspond to the components of the global continuous valued state vector \hat{x} , while its rows to the components of \hat{x}_I . The binary matrix H_I has a nonzero entry (equal to one) to its $i-j$ element, if the global state variable X_i appears to the j th component of the local state x_I . For the local state vectors one has

$$\hat{x}_I^g = H_I \hat{x}_I \quad (17)$$

where the local state matrix H_I is defined as follows:

$$[H_I]_{ml} = \begin{cases} 1 & \text{if } [x_I]_l \equiv X_m \\ 0 & \text{else} \end{cases} \quad (18)$$

In the context of the running example given in Eqs. (13) and (14), the local state matrixes for each DN are

$$H_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad H_2 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (19)$$

with corresponding full dimensional local vectors

$$\hat{x}_1^g = \begin{bmatrix} \hat{X}_1^{S_1} \\ \hat{X}_2^{S_1} \\ 0 \end{bmatrix} \quad \hat{x}_2^g = \begin{bmatrix} 0 \\ \hat{X}_2^{S_2} \\ \hat{X}_3^{S_2} \end{bmatrix} \quad (20)$$

The superscripts are added to clarify the subsystem that each component belongs to. We further introduce the binary selection system-to-state (StS) matrix $T \in \mathbb{R}^{I \times n_x}$. Each entry indicates the presence of a global state to each subsystem. The StS matrix T has a nonzero entry (equal to one) to its $i-j$ element, if the state variable X_j appears to subsystem S_i . The StS matrix is defined as

$$[T]_{ij} = \begin{cases} 1 & \text{if } X_j \in S_i \\ 0 & \text{else} \end{cases} \quad (21)$$

The n_x dimensional local state vectors \hat{x}_I^g are added together. Every element of the resulting summation vector is divided by the number of appearances of the corresponding state variable to the cut-point sets. Denote by $\sigma = \sum_{I=1}^N H_I \hat{x}_I^g$ the sum of the n_x dimensional state vectors. The i th element of the global fused state vector \hat{x} is given by

$$[\hat{x}]_i = \frac{1}{\sum_{j=1}^N [T]_{ji}} [\sigma]_i \quad (22)$$

For the running illustration, the StS matrix is

$$T = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad (23)$$

Hence, the global fused state vector \hat{x} according to Eq. (22) is given by

$$\hat{x} = \begin{bmatrix} \hat{X}_1^{S_1} \\ \frac{\hat{X}_2^{S_1} + \hat{X}_2^{S_2}}{2} \\ \hat{X}_3^{S_2} \end{bmatrix} \quad (24)$$

The block diagram, with the breakdown of the reduced-order DPFFD algorithm's steps, is depicted in Fig. 5. The corresponding pseudocode of the proposed algorithm is outlined in Table 4.

The reduced-order DPFFD algorithm results to a significant reduction in the computational complexity and communication bandwidth requirements of each DN. Suppose that the large-scale system has $n_x + 2 \cdot M$ states. The computational complexity of the centralized architecture, according to Ref. [42] and by considering N_s particles is approximated by $O((n_x + 2 \cdot M)^2 \cdot N_s)$ floating point operations (flops). By decomposing the system into N subsystems, the number of the state variables is decreased to $(n_x + 2 \cdot M)/N$ at each subsystem. Assuming that N_s particles are generated in every reduced-order estimator node, and with the assumption of no shared states, the total computational complexity of the reduced-order DPFFD algorithm reduces to $O((n_x + 2 \cdot M)^2 \cdot N_s/N)$.

5 Numerical Results

This section provides an evaluation of the proposed DPFFD algorithm via extensive numerical simulations. Two systems of

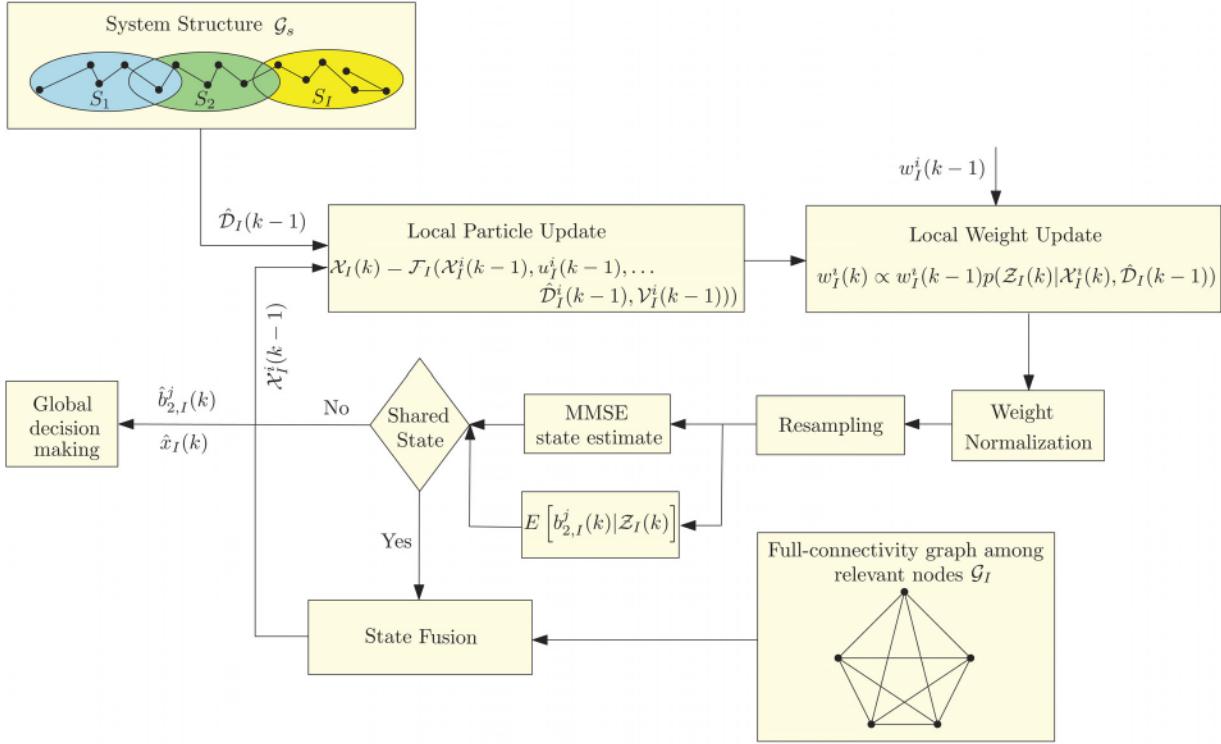


Fig. 5 Block diagram of the i th DN

Table 4 Pseudocode of the REDUCED-ORDER DPFFD algorithm

```

function REDUCED-ORDER DPFFD
Inputs :  $x_I(k-1), H_I, \{b_I^j(k-1)\}_{j=1}^M, w_I^i(k-1), u_I(k-1), Z_I(k), \mathcal{G}_s$ 
Outputs :  $\{X_I^i(k), w_I^i(k), \{b_{2,I}^j(k)\}_{j=1}^M, \tilde{x}\}$ 
Required: CPFFD

1: for  $I = 1 : N$  do
2:   for  $j = 1 : n_x$  do
3:     Find  $\mathcal{V}_j$  based on  $\mathcal{G}_s$  ▷ Find the neighborhood set of node j
4:      $d_I = \cup_{j \in \mathcal{V}_j} \{X_j\} - \cup_{j=1}^{n_q} \{X_j\}$  ▷ Calculation of forcing term vector
5:   end for
6:    $U_I = [(u_I)^T (d_I)^T]^T$  ▷ Control input at DN I
7:    $X_I = [(x_I)^T (b_I^1)^T \dots (b_I^2)^T \dots (b_I^M)^T]^T$  ▷ State vector of DN I
8:    $[X_I^i(k), w_I^i(k), \{b_{2,I}^j(k)\}_{j=1}^M] = CPFFD[X_I^i(k-1), w_I^i(k-1), U_I(k-1), Z_I(k)]$  ▷ Calculation of the particles, particles weights and probabilities of failure
9:    $\hat{x}_I^g = H_I \hat{x}_I$  ▷ Conversion of the reduced-order vector  $x_I$  to the full-order vector  $x_I^g$ 
10:  end for
11:   $\sigma = \sum_I \hat{x}_I^g$  ▷ Summation of the full-order local state vectors
12:  for  $i = 1 : n_x$  do
13:     $[\tilde{x}]_i = \frac{[\sigma]_i}{\sum_{j=1}^N [\sigma]_j}$  ▷ Calculation of the global state fusion vector's elements
14:  end for

```

different dimensionality are analyzed to validate the efficiency of the algorithm. In both cases, the process model under investigation is a water tank system. This process was selected since its dynamics are nonlinear and its physical subcomponents (water tanks) are clearly identified.

The first case study involves the water tank system illustrated in Fig. 6. This process consists of nine identical cylindrical tanks of

cross-sectional area S_c . The tanks are connected with pipes of cross section area S_p . The flow rate $Q_{i,j}$ between tank i and tank j is defined by means of Torricelli's rule as

$$Q_{i,j} = \mu_i \cdot S_p \cdot \text{sign}(X_i - X_j) \cdot \sqrt{2g|X_i - X_j|} \quad \text{with } i, j \in 1, 2, \dots, 10 \quad (25)$$

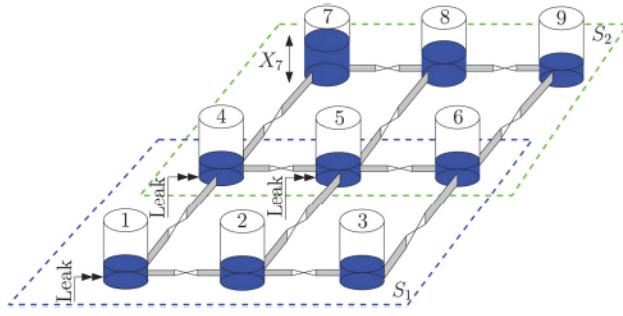


Fig. 6 Schematic of the nine-tank system that has been decomposed into two subsystems. The subsystems are specified by the dashed lines while the fault mode locations are noted by double arrows.

Table 5 Model parameters of the water tank system

Parameter	Meaning	Value
S_c	Tank cross-sectional area	0.0154 m ²
S_p	Cross section of interacting pipes	2×10^{-5} m ²
μ_i	Flow correction term	1
g	Gravity constant	9.81 m/s ²

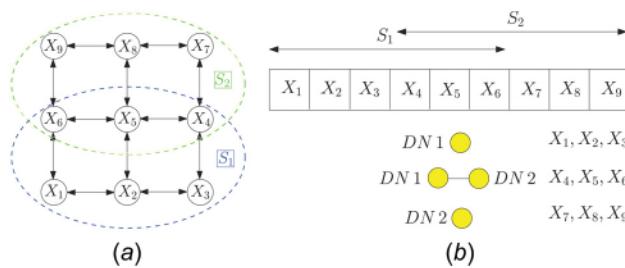


Fig. 7 (a) Structural graph and subsystem decomposition and (b) separation of shared and unshared states for observation fusion. The presence of the β term and the noise V are omitted for illustrative purposes.

where μ_i is the flow correction term of tank i , g is the gravity constant, and X_i is the water level of the i th tank. The water level dynamics of tank i is determined by means of mass balance equation as

$$\dot{X}_i = \frac{1}{S_c} \left(\sum_{j \in \mathcal{N}_i} Q_{ij} \right) \quad (26)$$

where \mathcal{N}_i refers to the neighboring tanks of tank i . The nominal values of the process's parameters are given in Table 5 and are based on the benchmark process described in Ref. [43]. The fault modes under consideration are abrupt leaks to the water tanks. The leakage dynamics of tank j are given by

$$g^j(X_j(k)) = \left(\frac{2 \cdot \mu_j \cdot S_p}{S_c} \right) \text{sign}(X_j(k)) \sqrt{2g|X_j(k)|} \quad (27)$$

By discretizing the continuous dynamics, the difference equation of the i th tank takes the form

$$X_i(k) = X_i(k-1) + \frac{T_s}{S_c} \sum_{j \in \mathcal{N}_i} Q_{ij}(k-1) + T_s \cdot \beta(k - k_0^i) \cdot g^i(X_i(k-1)) + v_i(k-1) \quad i = 1, \dots, 9 \quad (28)$$

where $T_s = 0.1$ s is the sampling period, and the process noise $v_i \in \mathbb{R}$ is drawn from the normal distribution $\mathcal{N}(0, 0.05)$. The goal of this simulation scenario is to investigate the case of decomposition with overlapping states. To this end, the monolithic process is decomposed into two reduced-order subsystems, namely S_1 and S_2 , as shown in Fig. 6. Figure 7(a) depicts the structural graph of the monolithic process and its partitions. The local observation vectors of the two DNs are expressed by

$$\begin{aligned} z_1(k) &= x_1(k) + \omega_1(k) \\ z_2(k) &= x_2(k) + \omega_2(k) \end{aligned} \quad (29)$$

where $x_1 = [X_1, \dots, X_6]^T$, $x_2 = [X_4, \dots, X_9]^T$, $d_1 = [X_7, X_8, X_9]^T$, $d_2 = [X_1, X_2, X_3]^T$, and the measurement noise sequences ω_1 and ω_2 are generated by the multivariate normal distribution $\mathcal{N}(0, 0.1)$. The subgraphs of the observation fusion are depicted in Fig. 7(b). As shown in this figure, the states $\{X_4, X_5, X_6\}$ are shared between the two DNs. Three failure modes are seeded at tanks 1, 4, and 5 at the time steps $T = 290, 250$, and 200 s, respectively. The time horizon of the simulation is set to 360 time steps. The number of particles at each DN is set to $N_s = 200$.

During the execution of the reduced-order DPFFD algorithm, the i.i.d noise that drives the binary states is generated by the distribution $\mathcal{U}(-0.6, 0.6)$. Figure 8 shows the population of the particles on the $b_1 - b_2$ plane during the healthy and faulty operating

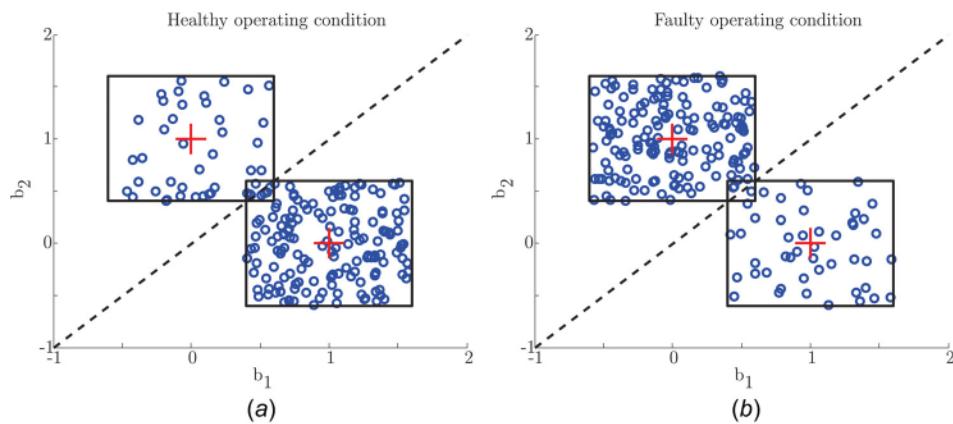


Fig. 8 Spatial distribution of the binary particles during the execution of the failure sensitive filter. The cross markers denote the locations of e_1 and e_2 , respectively.

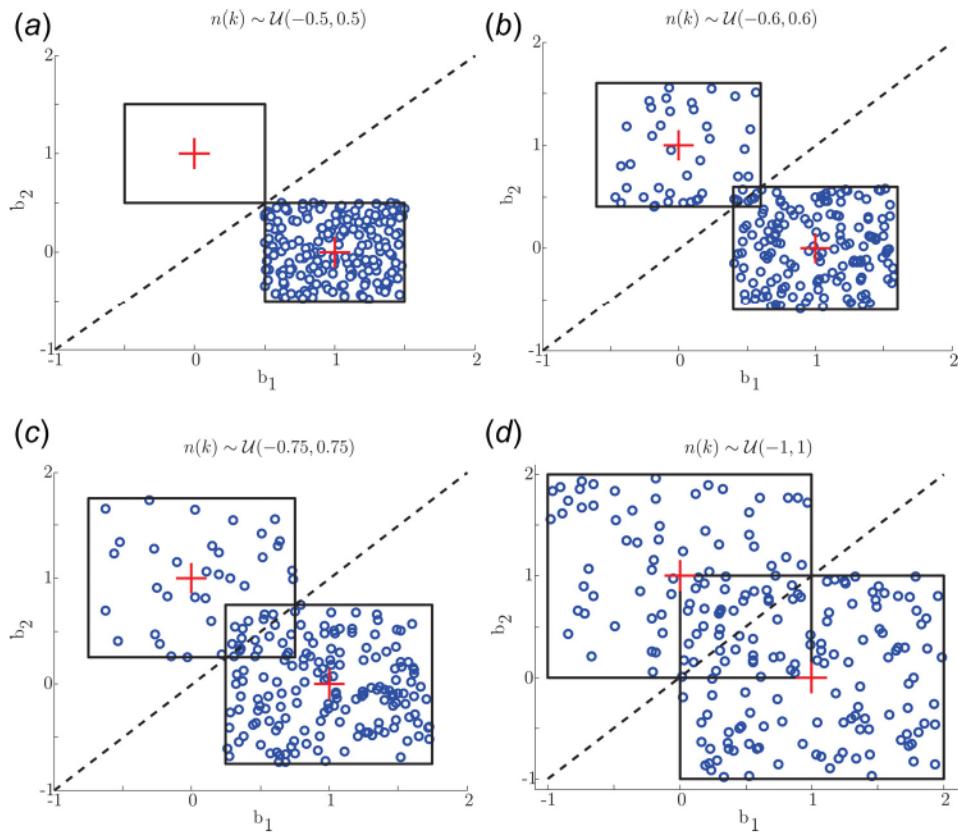


Fig. 9 Spatial distribution of the binary particles with different range of i.i.d driving noise when the process is in healthy operating condition. The cross markers denote the locations of e_1 and e_2 .

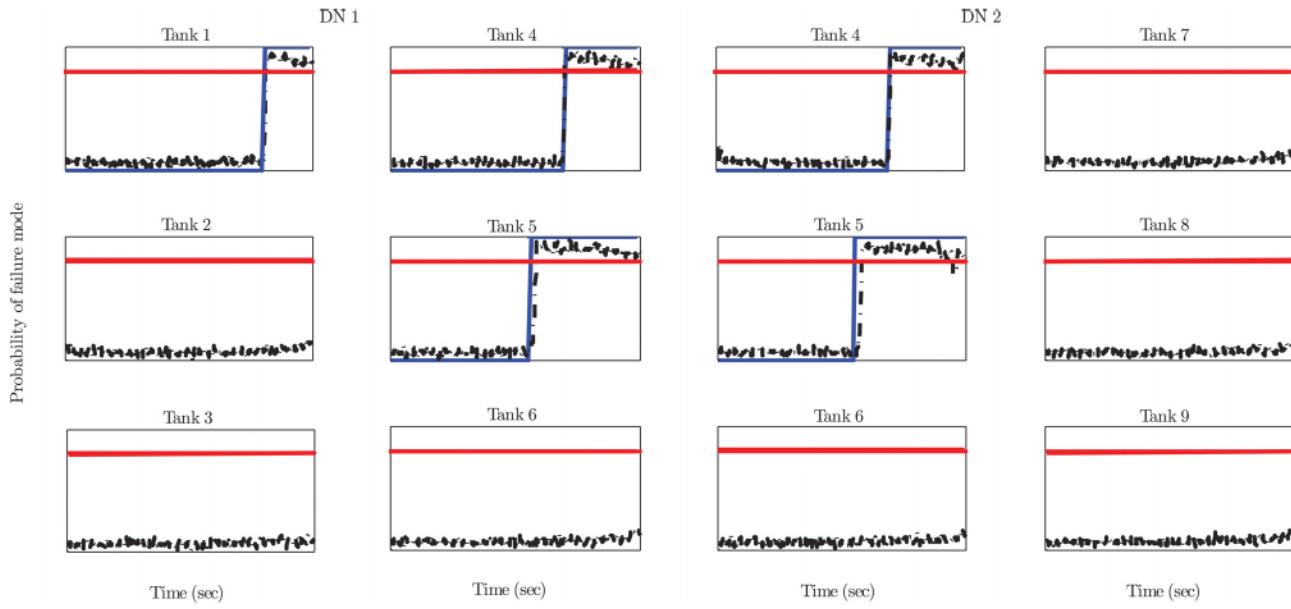


Fig. 10 *First case study:* probabilities of failure (dashed dotted line) for each fault mode as generated by the two DNs. The solid horizontal line marks the detection threshold. The solid vertical line indicates the occurrence instant of the fault. The states X_4 , X_5 , and X_6 are shared between the two DNs.

condition of the system at a given time instant. The selection of this noise range plays a crucial role in the performance of the algorithm. The effect of the i.i.d uniform white noise is illustrated in Fig. 9. When the noise is $U(-a, a)$ with $a = 0.5$ (too small), there is no overlap between the two regions; thus, the particles remain trapped in the healthy state even in the presence of a fault.

On the contrary, when the overlap increases ($a \geq 0.8$), the particles keep transitioning between the states and the output of the failure filter is indecisive.

The binary state's update function $\Phi(\cdot)$ of Eq. (10) is essentially a “data-driven feedback” for the failure sensitive filter. This way, when the process is healthy, the filter will diminish the particles

Table 6 Occurrence time of each fault mode

Fault mode number	Time of occurrence
Mode 82	10
Mode 91	30
Mode 13	50
Mode 92	70
Mode 64	90
Mode 10	110
Mode 28	130
Mode 55	150
Mode 96	170

that correspond to fault modes indirectly through the likelihood function [34]. A compromising value that ensures the optimal operation of the diagnostic filter was shown to be $\alpha = 0.6$.

The probabilities of failure for each fault mode are illustrated in Fig. 10. Due to the overlap of the two DNs, the estimates of common states $\{X_4, X_5, X_6\}$ are fused using the central averaging step described in Sec. 4. As it can be seen, both DNs can timely detect and isolate their respective fault modes.

The second simulation scenario involves a grid of 100 water tanks organized in a lattice of ten rows and ten columns. In this scenario, the dimensionality of the system is significantly increased compared to the first case. To the authors knowledge this is the highest dimension simulation example encountered in the literature of FD. The goal is to validate the DPFFD algorithm when applied to a large-scale system. A DN is assigned to each water tank. The DNs monitor the water level of their respective tank and exchange information with their adjacent nodes. The observation equation of each DN is expressed by

$$z_i(k) = x_i(k) + \omega_i(k) \quad (30)$$

where $x_i = X_i$ and $i = 1, \dots, 100$. The two-dimensional location of the tank in the grid array is converted to a single index. The one-dimensional index i is calculated by

$$i = n_{\text{row}} + (n_{\text{col}} - 1) \times N_{\text{rows}} \quad (31)$$

where n_{row} and n_{col} are the row and column number of the tank in the grid, and N_{rows} denotes the total rows in the array.

Abrupt leaks are seeded randomly to nine tanks at the time instances listed in Table 6. The nominal values of the system parameters are identical to the first scenario (Table 5). The measurement/process noise are drawn from the normal distribution $\mathcal{N}(0, 1)$. The time horizon of the simulation is set to 190 time

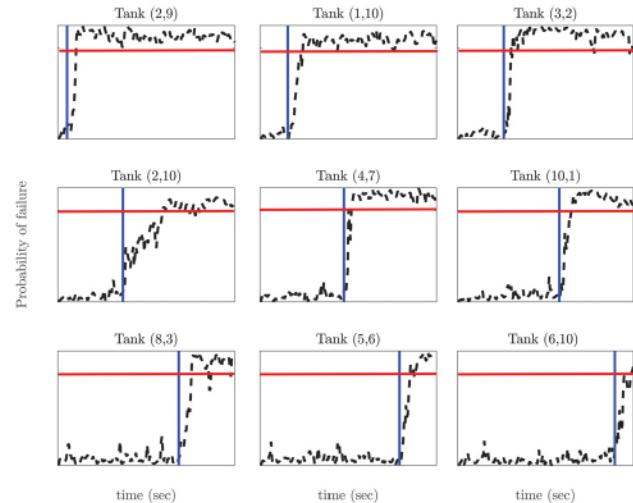


Fig. 12 Second case study: probabilities of failure (dashed line) with respect to time of the DNs with a leak occurring to their respective tank. The solid horizontal line marks the detection threshold. The solid vertical line signals the fault occurrence instant.

steps. The same tuning guidelines for the failure sensitive filters hold with the first example. The initial values of the estimated tank water levels are set to 14 m.

Due to the high dimensionality of the system's states, the simulation results are presented with respect to both time and space. The illustration of the probabilities of failure, the actual and estimated values of the water tank levels are shown in the first, second, and the third row of Fig. 11, respectively. The output values of the DNs are depicted as color-coded pixels based on their location in the lattice, for different time instances. The probabilities of failure with respect to time, only for the leaked tanks, are shown in Fig. 12. When a leak is seeded in one of the tanks, its water level will gradually reduce. For a transient interval, the neighboring tanks will try to compensate for this loss due to the pressure difference until their level will also start to decrease as well. The diagnostic performance is deemed satisfactory since each DN can promptly detect and isolate its own fault mode in spite of having access to local information. This case study involves only nonoverlapping subsystems; therefore, state fusion was not necessary. The computational reduction compared to the CPFFD algorithm is dramatic. Instead of processing 100 states, each node is responsible of monitoring a one-dimensional system.

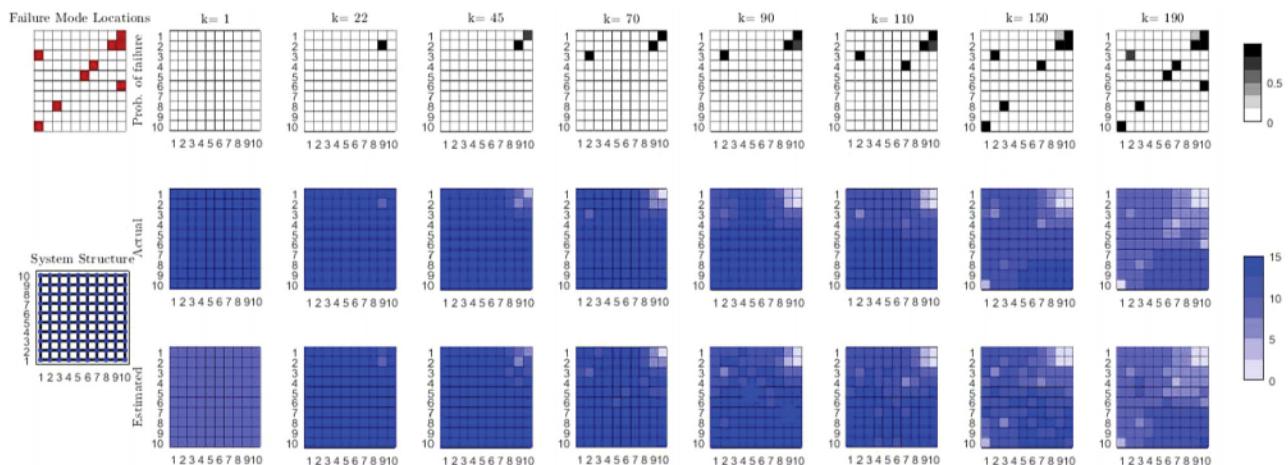


Fig. 11 Second case study: probabilities of failure (first row), actual (second row), and estimated (third row) values of the tanks water levels at different time instances. Each pixel represents the water level of a tank in the lattice.

6 Conclusions

We have presented a reduced-order distributed implementation of a fault detection and isolation algorithm for nonlinear large-scale systems. A network of interconnected DNs is employed to monitor the entire process. Each node monitors lower-order subdivisions of the monolithic system. The DNs have access to partial local measurements and can communicate with adjacent nodes of the monitoring network. The layout of the scheme is driven by the two main constraints of networked systems: the available communication bandwidth and processing capabilities of the nodes. An on-line hypothesis testing module is embedded at each failure sensitive filter that triggers alarm indicators in the presence of a fault. This inference component eliminates the need for the entire system's state at each DN and the necessity of a bank of estimators to isolate the occurring faults. A simplistic state fusion step takes place between nodes that monitor common states. This approach relieves the filter design analysis by substituting the complex stability proofs that are required by observed-based methods, with Monte Carlo simulations that are conveniently applicable to real-life sensor networks.

Funding Data

- National Science Foundation (NSF) (Award No. CMMI-1662742).

Nomenclature

b	= binary vector of S
B	= set of time profile functions
b_I	= binary vector of subsystem S_I^f
d_I	= interconnection variables (forcing terms) of subsystem S_I
e^i	= prediction error of particle i
$f(\cdot)$	= state transition function
$\mathcal{F}(\cdot)$	= compact state transition function
$f_I(\cdot)$	= state transition function of subsystem S_I
$\mathcal{F}_I(\cdot)$	= compact state transition function of subsystem S_I^f
\mathcal{G}_s	= structural graph (digraph) of system S
$g^j(\cdot)$	= function of fault mode j
$g_I^j(k)$	= fault function of failure mode j at subsystem S_I
$h(\cdot)$	= observation function
$h_I(\cdot)$	= observation function of subsystem S_I
$\mathcal{H}_I(\cdot)$	= compact local observation function
H_I	= local state matrix
k	= time index
k_0^j	= time occurrence of failure mode j
M	= number of failure modes
n	= uniform white noise
N	= number of subsystems
n_{d_I}	= dimension of subsystem's S_I forcing vector d_I
n_u	= dimension of input vector
n_{u_I}	= dimension of subsystem's S_I input vector u_I
n_v	= dimension of process noise vector
n_{v_I}	= dimension of subsystem's S_I system noise vector v_I
n_x	= dimension of state vector
n_{x_I}	= dimension of subsystem's S_I state vector x_I
n_z	= dimension of observation vector
n_{z_I}	= dimension of subsystem's S_I measurement vector z_I
n_ω	= dimension of measurement noise vector
n_{ω_I}	= dimension of subsystem's S_I measurement noise vector ω_I
N_s	= number of particles
$\mathcal{N}(\cdot, \cdot)$	= normal distribution
$p(x(k) z(1:k))$	= posterior density function

$p(x(k) x(k-1))$	= state transition density function
$p(z(k) x(k))$	= likelihood density function
$q(x(k) z(1:k))$	= proposal distribution function
\mathbb{R}	= set of real numbers
S	= monolithic system
S_I	= subsystem I of monolithic system S
S^f	= failure sensitive filter
S_I^f	= subsystem I of failure sensitive filter S^f
T	= StS matrix
u	= input vector
u_I	= input vector of subsystem S_I
U_I	= compact input vector of subsystem S_I^f (combination of u_I and d_I)
$\mathcal{U}(\cdot, \cdot)$	= uniform distribution
v	= process noise vector
\tilde{v}	= approximate process noise vector
V	= set of noise inputs
\mathcal{V}	= compact noise vector of system S^f (combination of \tilde{v} and n)
v_I	= process noise vector of subsystem S_I
\mathcal{V}_I	= compact noise vector of subsystem S_I^f (combination of v_I and n_I)
\mathcal{V}_m	= neighborhood set of vertex m
\mathcal{V}_s^I	= set of vertices of the graph \mathcal{G}_s
w^i	= particles' weight
w_I^i	= particles' weight of DN I
x	= state vector of monolithic system S
\tilde{x}	= global fused state vector
\mathcal{X}	= compact state vector of S^f (combination of x^c and $b_I^j _{j=1,\dots,M}$)
x_I	= state vector of subsystem S_I
X_i	= state variable i
\mathcal{X}_I	= compact state vector of subsystem S_I^f (combination of x_I^c and $b_I^j _{j=1,\dots,M}$)
x^i	= particles of x
x^c	= continuous valued state vector of S^f
\mathcal{X}_I^i	= particles generated by DN I
\hat{x}_I^g	= n_x order estimate of state x_I
$\hat{X}_j^{S_I^f}$	= estimate of X_j by DN I
z	= measurement vector
\mathcal{Z}	= compact measurement vector of system S^f
z_I	= observation vector of subsystem S_I
\mathcal{Z}_I	= compact observation vector of subsystem S_I^f
\mathbb{Z}^+	= set of positive integers
β	= time profile function of a fault's occurrence
$\delta(\cdot)$	= Dirac function
ε_s	= edges of the graph \mathcal{G}_s
Σ_ω	= covariance matrix of measurement noise
$\Phi(\cdot)$	= update function of the binary states
χ	= set of system states
ω	= measurement noise vector
$\tilde{\omega}$	= approximate measurement noise vector
ω_I	= measurement noise vector of subsystem S_I

References

- [1] Willsky, A. S., 1976, "A Survey of Design Methods for Failure Detection in Dynamic Systems," *Automatica*, 12(6), pp. 601–611.
- [2] Gertler, J. J., 1988, "Survey of Model-Based Failure Detection and Isolation in Complex Plants," *IEEE Control Syst. Mag.*, 8(6), pp. 3–11.
- [3] Frank, P. M., 1990, "Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge-Based Redundancy: A Survey and Some New Results," *Automatica*, 26(3), pp. 459–474.
- [4] Chen, J., and Patton, R. J., 1999, *Robust Model-Based Fault Diagnosis for Dynamic Systems*, Kluwer Academic Publishers, Dordrecht, The Netherlands.
- [5] Venkatasubramanian, V., Rengaswamy, R., Yin, K., and Kavuri, S. N., 2003, "A Review of Process Fault Detection and Diagnosis—Part I: Quantitative Model-Based Methods," *Comput. Chem. Eng.*, 27(3), pp. 293–311.
- [6] Ding, S. X., 2008, *Model-Based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*, Springer Science & Business Media, Berlin.
- [7] Baroni, P., Lamperti, G., Pogliano, P., and Zanella, M., 1999, "Diagnosis of Large Active Systems," *Artif. Intell.*, 110(1), pp. 135–183.

[8] Rish, I., Brodie, M., Ma, S., Odintsova, N., Beygelzimer, A., Grabarnik, G., and Hernandez, K., 2005, "Adaptive Diagnosis in Distributed Systems," *IEEE Trans. Neural Networks*, **16**(5), pp. 1088–1109.

[9] Le, T., and Hadjicostis, C. N., 2006, "Graphical Inference Methods for Fault Diagnosis Based on Information From Unreliable Sensors," Ninth International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, Dec. 5–8, pp. 1–6.

[10] Dong, H., Wang, Z., Ding, S. X., and Gao, H., 2014, "A Survey on Distributed Filtering and Fault Detection for Sensor Networks," *Math. Probl. Eng.*, **2014**, p. 858624.

[11] Ferrari, R. M., Parisini, T., and Polycarpou, M. M., 2012, "Distributed Fault Detection and Isolation of Large-Scale Discrete-Time Nonlinear Systems: An Adaptive Approximation Approach," *IEEE Trans. Autom. Control*, **57**(2), pp. 275–290.

[12] Boem, F., Ferrari, R. M., Parisini, T., and Polycarpou, M. M., 2013, "Distributed Fault Diagnosis for Continuous-Time Nonlinear Systems: The Input–Output Case," *Annu. Rev. Control*, **37**(1), pp. 163–169.

[13] Shames, I., Teixeira, A. M., Sandberg, H., and Johansson, K. H., 2011, "Distributed Fault Detection for Interconnected Second-Order Systems," *Automatica*, **47**(12), pp. 2757–2764.

[14] Yan, X.-G., and Edwards, C., 2008, "Robust Decentralized Actuator Fault Detection and Estimation for Large-Scale Systems Using a Sliding Mode Observer," *Int. J. Control.*, **81**(4), pp. 591–606.

[15] Zhang, X., and Zhang, Q., 2012, "Distributed Fault Diagnosis in a Class of Interconnected Nonlinear Uncertain Systems," *Int. J. Control.*, **85**(11), pp. 1644–1662.

[16] Daigle, M. J., Koutsoukos, X. D., and Biswas, G., 2007, "Distributed Diagnosis in Formations of Mobile Robots," *IEEE Trans. Rob.*, **23**(2), pp. 353–369.

[17] Boem, F., Ferrari, R. M., Parisini, T., and Polycarpou, M. M., 2011, "A Distributed Fault Detection Methodology for a Class of Large-Scale Uncertain Input–Output Discrete-Time Nonlinear Systems," 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), Orlando, FL, Dec. 12–15, pp. 897–902.

[18] Franco, E., Olfati-Saber, R., Parisini, T., and Polycarpou, M. M., 2006, "Distributed Fault Diagnosis Using Sensor Networks and Consensus-Based Filters," 45th IEEE Conference on Decision and Control (CDC), San Diego, CA, Dec. 13–15, pp. 386–391.

[19] Olfati-Saber, R., 2009, "Kalman-Consensus Filter: Optimality, Stability, and Performance," 48th IEEE Conference on Decision and Control (CDC), Shanghai, China, Dec. 15–18, pp. 7036–7042.

[20] Olfati-Saber, R., 2007, "Distributed Kalman Filtering for Sensor Networks," 46th IEEE Conference on Decision and Control (CDC), New Orleans, LA, Dec. 12–14, pp. 5492–5498.

[21] Kadirkamanathan, V., Li, P., Jaward, M. H., and Fabri, S. G., 2002, "Particle Filtering-Based Fault Detection in Non-Linear Stochastic Systems," *Int. J. Syst. Sci.*, **33**(4), pp. 259–265.

[22] Cheng, Q., Varshney, P. K., Michels, J. H., and Belcastro, C. M., 2009, "Distributed Fault Detection With Correlated Decision Fusion," *IEEE Trans. Aerosp. Electron. Syst.*, **45**(4), pp. 1448–1465.

[23] Cheng, Q., Varshney, P. K., Michels, J., and Belcastro, C. M., 2005, "Distributed Fault Detection Via Particle Filtering and Decision Fusion," Eighth International Conference on Information Fusion (ICIF), Philadelphia, PA, July 25–28, p. 8.

[24] Noursadeghi, E., and Raptis, I., 2015, "Full-Order Distributed Fault Diagnosis for Large-Scale Nonlinear Stochastic Systems," *ASME Paper No. DSCC2015-9927*.

[25] Noursadeghi, E., and Raptis, I., 2015, "Distributed Fault Detection of Nonlinear Large-Scale Dynamic Systems," *ACM/IEEE Sixth International Conference on Cyber-Physical Systems*, Seattle, WA, Apr. 14–16, pp. 51–59.

[26] Hlinka, O., Hlawatsch, F., and Djuric, P. M., 2013, "Distributed Particle Filtering in Agent Networks: A Survey, Classification, and Comparison," *IEEE Signal Process. Mag.*, **30**(1), pp. 61–81.

[27] Hlinka, O., 2012, "Distributed Particle Filtering in Networks of Agents," *Ph.D. dissertation*, Vienna University of Technology, Vienna, Austria.

[28] Mohammadi, A., 2013, "Distributed Implementations of the Particle Filter With Performance Bounds," *Ph.D. dissertation*, York University, Toronto, ON, Canada.

[29] Mohammadi, A., and Asif, A., 2011, "A Consensus/Fusion Based Distributed Implementation of the Particle Filter," Fourth IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), San Juan, Puerto Rico, Dec. 13–16, pp. 285–288.

[30] Sadeghzadeh Nokhodberiz, N., and Poshtan, J., 2014, "Belief Consensus-Based Distributed Particle Filters for Fault Diagnosis of Non-Linear Distributed Systems," *Proc. Inst. Mech. Eng., Part I*, **228**(3), pp. 123–137.

[31] Ristic, B., Arulampalam, S., and Gordon, N., 2004, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Vol. 830, Artech House, London.

[32] Gordon, N. J., Salmond, D. J., and Smith, A. F., 1993, "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation," *IEE Proc. F Radar Signal Process.*, **140**(2), pp. 107–113.

[33] Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T., 2002, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Trans. Signal Process.*, **50**(2), pp. 174–188.

[34] Orchard, M., 2006, "A Particle Filtering-Based Framework for Online Fault Diagnosis and Failure Prognosis," *Ph.D. dissertation*, Georgia Institute of Technology, Atlanta, GA.

[35] Raptis, I. A., and Vachtsevanos, G., 2011, "An Adaptive Particle Filtering-Based Framework for Real-Time Fault Diagnosis and Failure Prognosis of Environmental Control Systems," *Annual Conference of the Prognostics and Health Management Society*, Montreal, QC, Canada, Sept. 25–29.

[36] Raptis, I. A., Sconyers, C., Martin, R., Mah, R., Oza, N., Mavris, D., and Vachtsevanos, G. J., 2013, "A Particle Filtering-Based Framework for Real-Time Fault Diagnosis of Autonomous Vehicles," *Annual Conference of the Prognostics and Health Management Society*, New Orleans, LA, Oct. 14–17.

[37] Ferrari, R. M., 2009, "Distributed Fault Detection and Isolation of Large-Scale Nonlinear Systems: An Adaptive Approximation Approach," *Ph.D. dissertation*, Universita degli studi di Trieste, Trieste, Italy.

[38] Siljak, D. D., 1978, *Large-Scale Dynamic Systems: Stability and Structure*, Vol. 2, North Holland, Amsterdam, The Netherlands.

[39] Vidyasagar, M., 1980, "Decomposition Techniques for Large-Scale Systems With Nonadditive Interactions: Stability and Stabilizability," *IEEE Trans. Autom. Control*, **25**(4), pp. 773–779.

[40] Mesbahi, M., and Egerstedt, M., 2010, *Graph Theoretic Methods in Multiagent Networks*, Princeton University Press, Princeton, NJ.

[41] Khan, U. A., and Moura, J. M., 2008, "Distributing the Kalman Filter for Large-Scale Systems," *IEEE Trans. Signal Process.*, **56**(10), pp. 4919–4935.

[42] Karlsson, R., Schon, T., and Gustafsson, F., 2005, "Complexity Analysis of the Marginalized Particle Filter," *IEEE Trans. Signal Process.*, **53**(11), pp. 4408–4411.

[43] Zhang, X., Polycarpou, M. M., and Parisini, T., 2002, "A Robust Detection and Isolation Scheme for Abrupt and Incipient Faults in Nonlinear Systems," *IEEE Trans. Autom. Control*, **47**(4), pp. 576–593.