A Particle-Filtering Based Approach for Distributed Fault Diagnosis of Large-Scale Interconnected Nonlinear Systems

Elaheh Noursadeghi, Ioannis Raptis
Mechanical Engineering Department, University of Massachusetts Lowell
Email: {Elaheh_Noursadeghi, Ioannis_Raptis}@uml.edu

Abstract—This paper deals with the problem of designing a distributed fault detection and isolation algorithm for nonlinear large-scale systems that are subjected to multiple fault modes. To solve this problem, a network of communicating detection nodes is deployed to monitor the monolithic process. Each node consists of an estimator with partial observation of the system’s state. The local estimator executes a distributed variation of the particle filtering algorithm using the partial sensor measurements and the fault progression model of the process. During the implementation of the algorithm, each node communicates with its neighbors by sharing pre-processed information. The communication topology is defined using graph theoretic tools. The information fusion between the neighboring nodes is performed by means of a distributed average consensus algorithm to ensure the agreement over the value of the local likelihood functions. The proposed method enables online hypothesis testing without the need of a bank of estimators. Numerical simulations demonstrate the efficiency of the proposed approach.

Index Terms—distributed fault diagnosis, large-scale systems, particle filtering, networked control systems, sensor networks, information fusion.

I. INTRODUCTION

Every system is susceptible to faults that may lead to catastrophic failures. Complex processes are significantly more vulnerable to faults, since a malfunction in a single component may have a major effect to the entire system. Availability, dependability and resiliency are becoming major design goals for large-scale technological systems due to stringent economic, ecological and safety demands [1]. 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 downtimes, breakdowns and catastrophic failures.

The vast majority of the existing fault diagnosis methods have a centralized architecture [2], [3], where all the sensors measurements are transmitted to a single computing node that process the information and determines the condition of the system. Among the existing centralized FD methods, model-based approaches are typically preferred since they are compliant to performance analysis [4]–[9]. Model-based FD makes explicit use of a mathematical model of the process. In this framework, the actual measurements from the system are compared with the information derived by the analytical model. A fault is detected when there is a change between the actual measurements and the output of the model.

The existing model-based FD methods can be roughly classified into four main categories: Parameter identification, parity relations, observer-based and estimation-based. Identification methods involve estimating one or more process parameters in order to infer about the health of the system. Parity relations algorithms depend on the design of algebraic residuals between the actual and the modeled behavior of the system [3], [7], [8], [10]. Observer-based and estimation-based schemes essentially operate in a similar manner. Both the process’s states and outputs are estimated using state-space models. The main difference between these two model-based methods lies in the consideration of model and measurement uncertainty. Estimation-based methods apply a probabilistic treatment of uncertainty while in observer-based methods uncertain parameters and signals are treated as unknown, deterministic disturbances with known bounds. The convergence of fault sensitive observers is established through Lyapunov arguments, while estimation theory approaches employ Bayes theory.

State estimation for Linear Time Invariant (LTI) systems was introduced by Luenberger [11], [12]. The implementation of state observers for FD of LTI systems is attributed to the foundational work of Beard, Jones and Clark [13]–[15]. Precursory research on observer-based FD for nonlinear systems is reported in [16]–[18]. Significant milestone for the estimation-based methods was the work of Wilsky [19] and Basseville [20] that established likelihood ratio tests as a means for on-line hypothesis testing. The significant majority of estimation-based methods for FD relies on Kalman Filter (KF) and its derivatives such as the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF) [7], [21]–[25]. Early works on the application of KF for FD are reported in [19]–[21]. Since then many variations of the KF for FD have appeared in the scientific literature [8]. Their main limitation is that Kalman filtering methods are restricted to linear systems driven by Gaussian noise. For the case of nonlinear systems, the implementation of the EKF [26] requires the linearization of the system’s dynamics. This approximation is deteriorating the fidelity of the model and subsequently leading to high false alarm rates [27]. This limitation is evaded by introducing the Particle Filter (PF) [28], [29]. The particle filtering technique is a highly suitable estimator for fault diagnosis since it
avoids linearity and Gaussian noise assumptions. The first implementation of the PF for FD is reported in [27], [30]. A comprehensive PF method for combined FD and prognosis has been successfully presented in [31], [32] for a wide variety of engineering processes.

While the centralized architecture is appropriate for small and centralized systems, it is ill-suited for large-scale systems with spatially distributed components. The main reasons are: i) the spatial coverage of the region, over which the dynamic process takes place, requires the remote deployment of a large number of sensors that are connected to a single processing center. This leads to an overhead to the communication bandwidth as the number of sensors increase; ii) in some applications the transmission of data to the processing center is practically difficult or even impossible; and iii) malfunction of the central monitoring unit will unavoidably result to the termination of the diagnosis task.

A distributed derivation of the problem would be an excellent alternative that can meet both the computational and communication constraints of the original process. The design challenge is to generate in a distributive manner a reliable global inference of the process’s health using a network of diagnostic nodes that have access to only local information. Despite the obvious benefits, there are very few algorithms available in the literature. These limited attempts are i) suitable only for linear systems; ii) deterministic, thus, they do not handle the stochastic nature of uncertainty; and iii) computationally deficient since they require a bank of estimators equal to the number of fault modes, increasing the dimensionality of the problem.

The significant majority of the proposed distributed FD methods are tailored to discrete-event systems for multi-processor computing applications [33]–[35]. A primary taxonomy of the existing distributed FD methodologies for dynamic systems is based on the type of exchanged data between the nodes of the diagnostic network. The diagnostic nodes can interchange: state estimates [36], [37], raw measurements of the interconnected states [38]–[40], or fault signatures [41]. The most prominent work on distributed, observed-based FD for nonlinear dynamic systems has been reported by Ferrari et al. [38] and Boem et al. [42]. 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 [43]. The algorithm is based on the distributed version of KF established by Olfati-Saber in [44], [45]. Foundational work on estimation-based FD for nonlinear systems that employ large number of correlated sensors is introduced in [46], [47]. The author combines a distributed PF algorithm for state estimation with fused hypothesis testing through likelihood tests, to determine the occurrence of fault modes. A comprehensive analysis on distributed PF algorithms is given by Hlinka [48], [49] and Mohammadi [50], [51].

In this work, we present a full-order distributed fault diagnosis algorithm for large-scale nonlinear systems. The design objective is to develop a network of interconnected fault sensitive filters for monitoring a stochastic nonlinear process that is subjected to a set of active fault modes. Each node has access to a partial and noisy measurement of the system’s state and to processed statistical information from its neighboring nodes. The network can infer about the health of the entire system based on individual and partial observations obtained by the diagnostic nodes. Every node consists of an embedded processing unit that computes a local PF algorithm and a consensus filter that fuses the processed information of neighboring nodes such that the entire diagnostic system reaches an agreement of its estimates. Graph-based abstractions will be used to represent the active communication channels between the nodes.

The proposed approach is suitable for distributed large-scale systems. Global inference of the process’s health is obtained through local measurements and processing. The PF is an ideal estimator for complex nonlinear systems that relieves the analysis from complex Lyapunov convergence arguments that are required by observer-based methods. Furthermore, the proposed distributed diagnostic filter had embedded an on-line hypothesis testing module that eliminates the need for a collection of estimators to conduct fault isolation. The execution of the diagnostic algorithm is completely distributed without requiring a priori knowledge of the networks topology. The numerical simulation results illustrate the correlation between the detection network’s topology and the performance of the diagnostic algorithm.

This paper is organized as follows: A brief description of the PF algorithm is presented in Section II. The synthesis of a centralized PF fault diagnosis algorithm is outlined in Section III. The centralized algorithm serves as a benchmark framework for its distributed counterpart. The distributed version of the failure sensitive filter is presented in Section IV. The performance of the proposed methodology is evaluated in Section V via numerical simulations. Finally, concluding remarks are given in Section VI.

II. 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:

\[
x(k) = f(x(k-1), v(k-1))
\]

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:

\[
z(k) = h(x(k)) + \omega(k)
\]

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 \( \omega(k) \in \mathbb{R}^{n_\omega} \) 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 (pdfs). 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 (MMSE):

\[
\hat{x}(k) \triangleq E\{x(k)|z(1:k)\} = \int x(k) \cdot p(x(k)|z(1:k)) dx(k) 
\]

(3)

The PF is a Sequential Monte Carlo (SMC) method that uses a finite set of weights (referred as “particles”) to represent probability densities [52]. The basic idea is to represent the non-Gaussian posterior pdf \( p(x(k)|z(k)) \) by a set of randomly drawn samples of 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:

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

(4)

where \( \delta(\cdot) \) denotes the multivariate delta Dirac function. As the number of particles becomes very large, the SMC representation is closer to the analytical description of the posterior. Using the representation in (4), one can obtain various estimates of \( x(k) \). For example the MMSE estimate is approximated as:

\[
\hat{x}(k) = \sum_{i=1}^{N_s} w^i(k) \int x(k) \delta(x(k) - x^i(k)) dx(k) 
\]

(5)

\[
= \sum_{i=1}^{N_s} w^i(k)x^i(k)
\]

A way to mitigate the inability to directly sample from a posterior distribution, is to apply the method of Sequential Importance Sampling (SIS). This technique allows the computation of a distribution using random samples that are drawn from another. The particles \( x^i(k) \) are drawn from an importance density (proposal pdf) \( q(x(k)|x(k-1), z(k)) \) that depends on the state-transition pdf \( p(x(k)|x(k-1)) \). Then, the weights \( w^i(k) \) are sequentially calculated involving the likelihood \( p(z(k)|x(k)) \) according to:

\[
w^i(k) \propto w^i(k-1) \frac{p(z(k)|x^i(k))p(x^i(k)|x^i(k-1))}{q(x^i(k)|x^i(k-1), z(k))} 
\]

(6)

Filtering via SIS takes place by recursive updates of the importance weights \( w^i(k) \) and the support points \( x^i(k) \) as a new measurement \( z(k) \) becomes available. These steps form the basis algorithms of most PFs. Obviously, the performance of the filter is primarily determined based on the selection of the importance density \( q(x(k)|x(k-1), z(k)) \). In general, the accuracy of the PF improves when the proposal pdf \( q(x(k)|x(k-1), z(k)) \) is closer to the posterior pdf \( p(x(k)|z(k)) \).

From standard results [53], the optimal importance density function that minimizes the variance of particles weights, calculated in (5), is:

\[
q(x(k)|x^i(k-1), z(k))_{\text{opt}} = \frac{p(z(k)|x^i(k))p(x(k)|x^i(k-1))}{p(z(k)|x^i(k-1))} 
\]

(7)

that yields the following update equation for the weights:

\[
w^i(k) \propto w^i(k-1)p(z(k)|x^i(k-1)) 
\]

(8)

The optimal choice is not always straightforward since one needs to sample from \( p(x(k)|x^i(k-1), z(k)) \) and calculate:

\[
p(z(k)|x^i(k-1)) = \int p(z(k)|x(k))p(x(k)|x^i(k-1)) dx(k) 
\]

(9)

up to a normalizing constant [52]. Different variations of the PF algorithm exist depending on the choice of the importance density 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 [28], the auxiliary PF [54], and the regularized PF [54]. These PFs are derived using a suboptimal choice of the proposal pdf \( q(x(k)|z(k-1), z(k)) \).

The most popular SIR algorithm is the bootstrap filter that uses the state-transition density \( p(x(k)|x(k-1)) \) as the importance pdf. In this case, the update weight equation simplifies to:

\[
w^i(k) = w^i(k-1)p(z(k)|x^i(k)) 
\]

(10)

Compared to the optimal case of (9), in the bootstrap filter, the proposal distribution does not use the most recent observation \( z(k) \). If we consider the measurement noise \( \omega \) as a Gaussian distribution with zero mean and covariance matrix \( \Sigma_\omega \), the likelihood function for each particle is calculated by:

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

(11)

where \( \mathcal{N}(\cdot, 0, \Sigma) \) refers to the pdf of the normal distribution, and \( e^i(k) = z(k) - h(x^i(k)) \) represents the prediction error of the \( i \)-th particle.

The major problem of the SIS algorithms is the degeneration of samples over time. Degeneracy implies that gradually an increasing number of samples will have negligible contribution to the posterior. This is due to the fact that the transition prior \( p(x(k)|x(k-1)) \) is not conditioned on the measurement data. The performance of the algorithm will deteriorate rapidly, especially when the measurement noise is small.

To make the SIS simulation-based techniques viable, the resampling method of particles is adopted. The fundamental idea behind resampling is the preservation of particles with
large weights while discarding those with small weights. This action prevents diversity by replicating particles with the highest weights [53], [54]. The resampling can be performed in different ways. The most common method is systematic resampling. For a detailed description of the available resampling techniques the reader is referred to [53]. The pseudocode of the bootstrap filter and resampling are outlined in Table II and [11] respectively. The block diagram of the bootstrap algorithm is shown in Figure 1.

III. CENTRALIZED PARTICLE FILTERING FAULT DETECTION AND ISOLATION

In this work, we extend the methodology reported in [31] from simple one dimensional fault-growth models to state-space representations of generic nonlinear processes introducing the Centralized Particle Filtering Fault Diagnosis (CPFFD) algorithm. The detection objective of the CPFFD algorithm is twofold: to provide an estimate of the system’s state from a noisy measurements sequence (filtering), and generate a statistical characterization 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:

$$x(k) = f(x(k-1), u(k-1)) + \ldots$$

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

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

(12)

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 subject to $M$ potential fault modes described by the nonlinear functions $\left\{ g^j(x(k-1), u(k-1)) \right\}_{j=1}^{M}$ with $g^j : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$. The term $\beta(k - k_0)$ is a scalar function representing the time profile of the fault occurrence that takes place at some unknown time $k_0$. We can consider both abrupt (step-like) or incipient (exponential-like) fault modes, defined as:

$$\beta(k - k_0) = \begin{cases} 0 & k < k_0 \\ 1 - e^{-(k-k_0)/\Delta} & k > k_0 \end{cases}$$

(13)

The system is assumed to be healthy at the initial time $k = 0$ ($\beta(0) = 0$). The algorithm incorporates the process model of (12), as well as a binary state vector, that estimate changes in the process dynamics expressed by the terms $\beta(k - k_0^j)$. To this end, a vector of binary states $b^j(k) = [b_1^j(k) \ b_2^j(k) \ \cdots]^T$ with $b_1, b_2 \in \{0, 1\}$ and $j = 1, \ldots, M$, is used to signal the occurrence of each fault mode. More specifically, $b_1^j(k) = 1$ indicates that the system is operating normally, while $b_2^j(k) = 1$ denotes that the fault mode $j$ is detected to the system. This technique provides an elegant way to calculate on-line probabilistic measures related to the existence of a fault in the system.

To determine the operating condition of the system (normal or faulty operating condition) and make a decision based on the occurrence of faults, a particle filtering approach is employed for the statistical characterization of both the binary and continuous-valued states, as new measurements are received. The binary states update sequence is embedded to the state-transition model $S$ given in (14). This way, the process model $S$ is converted to a failure sensitive filter that signals the presence of changes to the original healthy dynamics. Hence, the state vector that is used by the PF algorithm is $X^T(k) = [x^e(k)]^T \ b^1(k) \ b^2(k) \ \cdots \ b^M(k)]^T \in \mathbb{R}^{n_x+2M}$ where $x^e(k) \triangleq x(k)$. The state-space model of the system that is implemented to the PF algorithm is expressed in terms of the dynamics of two interconnected subsystems: The dynamics of the continuous and binary states, respectively.

---

Table II

| Function RESAMPLE |
|-------------------|
| **Inputs:** $x^e(k), w_i(k)$ |
| **Outputs:** $x^e(k), w_i(k)$ |
| 1: $c(1) = 0$  |
| 2: for $i = 2 : N_s$ do |
| 3: $c(i) = c(i-1) + w_i(k)$ |
| 4: end for |
| 5: $i = 1$  |
| 6: $u(1) = U[0, N_s^{-1}]$  |
| 7: for $j = 1 : N_s$ do |
| 8: $u(j) = u(1) + N_s^{-1}(j - 1)$ |
| 9: while $u(j) > c(i)$ do |
| 10: $i = i + 1$ |
| 11: end while |
| 12: $x^{res}(k) = x^e(k)$ |
| 13: $w^*(k) = N_s^{-1}$ |
| 14: end for |

---

Figure 1. Block diagram of the bootstrap PF.
The state-transition dynamics of $\mathcal{X}$ are described by:

$$
\begin{align*}
x^c(k) &= f(x^c(k-1), u(k-1)) + \ldots \\
M \sum_{j=1} g^j(x^c(k-1), u(k-1)) \cdot b^j_k(k-1) + \tilde{v}(k-1) \\
b^j_k(k) &= \Phi(b^j_k(k-1) + n^j(k-1)), \quad j = 1, \ldots, M \\
z(k) &= h(x^c(k)) + \tilde{w}(k)
\end{align*}
$$

(14)

where $\tilde{v}(k) \in \mathbb{R}^{n_x}$, and $\tilde{w}(k) \in \mathbb{R}^{n_z}$ are 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 evolution of the binary states $\Phi : \mathbb{R}^2 \rightarrow \{[0,1]^T, [1,0]^T\}$ is a nonlinear function 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 $b^j_k(k)$ is randomly excited at each time step by $n^j(k)$. This random vector of $\mathbb{R}^2$ is assigned to one of the binary states (healthy/faulty) based on the distance metric of the perturbed vector $b^j_k(k-1) + n^j(k-1)$ to the coordinates $[0,1]^T$ and $[1,0]^T$. With this technique, when a fault occurs, the weights will gradually converge the binary variable $b^j_k(k)$ to one ($b^j_k(k) \rightarrow 1$). This is due to the fact that the likelihood of the measurements will diminish the weights of particles associated with the healthy condition. A choice of $\Phi(\cdot)$ that has been successfully used in $[31]$, $[55]$, $[56]$ is:

$$
\Phi(x) = \begin{cases} 
eq 1 & \text{if } \|x - e_1\| \leq \|x - e_2\| \\ e_2 & \text{else} \end{cases}
$$

(15)

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:

$$
\begin{align*}
\mathcal{X}(k) &= \mathcal{F}(\mathcal{X}(k-1), u(k-1), \mathcal{V}(k-1)) \\
\mathcal{Z}(k) &= \mathcal{H}(\mathcal{X}(k)) + \tilde{w}(k)
\end{align*}
$$

(16)

where $\mathcal{Z}(k) = z(k)$, $\mathcal{V}(k) = [\tilde{v}(k) \ n^1(k) \ldots n^M(k)]$ and $\mathcal{F}(\cdot), \mathcal{H}(\cdot)$ are nonlinear functions of appropriate dimensions and structure. The above definition will be used to ease the notation in subsequent parts of the analysis. The outputs of the CPFFD module are the probabilities of failure of each fault mode. These are the expectations of the binary states $b^j_k(k) = E[b^j_k(k)|\mathcal{Z}(k)]$. 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. $b^j_k(k) < \alpha$ indicates normal operation). With this layout two or more different co-existing fault modes can be simultaneously detected.

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 require 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 one ($b^j_k(k)$) 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 require the execution of a bank of estimators that is equal to the fault modes. For large-scale systems this computational load is prohibited.

### IV. DISTRIBUTED PARTICLE FILTERING FAULT DIAGNOSIS

In this Section we introduce a full-order Distributed Particle Filtering Fault Diagnosis (DPFFD) algorithm for large-scale non-linear/non-Gaussian systems. The main objective is to develop a network of interconnected Diagnostic Nodes (DNs) to monitor a stochastic nonlinear process that is subjected to a set of active fault modes. The DN's should detect distributively the occurrence of the fault modes.

Each DN has partial access to the system’s state and can exchange information with its neighboring nodes. Every node executes a localized PF for estimating the state of the entire system, and a consensus protocol that fuses locally the likelihood function of the node with its neighbors. The output of the detection network is a filtered estimate of the process’s state and the probability of failure of each fault mode.

The CPFFD algorithm can be conveniently executed in a distributed manner using a Distributed Particle Filtering (DPF) formulation and overcome the limitations of the centralized PF algorithm. The main challenge is to guarantee that all DN's consent to a common reliable estimate of the state variables $[50]$. In this work, we employ a statistic dissemination consensus-based DPF algorithm, presented first in $[48]$, to serve as the basis of the DPFFD algorithm.

Graph-based abstractions will be used to represent the active communication channels between the nodes. The communication network is described by the graph $G$, defined as the pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ is the vertex set or DN's set, and $\mathcal{E} = \{(v_i, v_j) \in \mathcal{V} \times \mathcal{V}\}$ represents the set of edges of $\mathcal{G}$. Each element of $\mathcal{E}$ represents an undirected communication link between two DNs. The neighborhood $N_i \subseteq \mathcal{V}$ of the vertex $v_i$ is defined as the set of all vertices that are adjacent to $v_i$, $\{v_j \in \mathcal{V} : (v_i, v_j) \in \mathcal{E}\}$. If $v_j \in N_i$, it follows that $v_i \in N_j$, since an undirected edge exists between them $[51]$. The degree of the node $i$ is defined as $d_i = |N_i|$. Similar to the derivation of Section $[12]$ we assume that the monolithic system $S$ of $[12]$ is subjected to $M$ potential fault modes. A network of $N$ DNs is deployed to monitor the process’s dynamics. The interconnection of the DNs is defined by

---

**Table III**

| PSEUDOCODE OF THE CPFFD ALGORITHM |
|-----------------------------------|
| **function** CPFFD |
| **Inputs**: $\mathcal{X}(k-1), w^i(k-1), Z(k)$ |
| **Outputs**: $\mathcal{X}(k), w^i(k)$, $\{b^j_k(k)\}_{j=1}^M$ |
| **Required**: RESAMPLE |
| 1: for $i = 1 : N$, do |
| 2: $\mathcal{X}(k) = \mathcal{F}(\mathcal{X}(k-1), w^i(k-1), \mathcal{V}(k-1))$ -> Particle update |
| 3: $w^i(k) = w^i(k-1) \cdot p[Z|\mathcal{X}(k)]$ -> Weights update |
| 4: end for |
| 5: $w^i(k) = \frac{w^i(k)}{\sum_{j=1}^N w^j(k)}$ -> Weight normalization |
| 6: if $[\mathcal{X}(k), w^i(k)] \in RESAMPLE[\{\mathcal{X}(k), w^i(k)\}_{i=1}^N]$ -> Resampling |
| 7: $\mathcal{X}(k) = \sum_{i=1}^N w^i(k)\mathcal{X}(k)$ -> Minimum Mean-Square Error |
| 8: for $j = 1 : M$, do |
| 9: $b^j_k(k) = E[b^j_k(k)|Z(k)]$ -> Probabilities of failure |
| 10: end for |
the communication graph $\mathcal{G}$. The local measurement equation of each DN is expressed by:

$$z_I(k) = h_I(x(k)) + \omega_I(k) \quad \text{with} \quad I = 1, 2, \ldots, N$$

(17)

The global measurement vector can be defined as the union of the local observation vectors, $z(k) = \bigcup_{I=1}^{N} z_I(k)$. Analogously to (16), a compact formulation of the state-transition model that is executed by the local PF module of each DN is given by:

$$X(k) = \mathcal{F}(X(k-1), u(k-1), \nu(k-1))$$

$$Z_I(k) = H_I(X(k)) + \tilde{\omega}_I(k)$$

(18)

where $\mathcal{F}()$ is given in (14), and $H_I()$ is a nonlinear function that represents the local measurement model based on (17). As a reminder, the state $X$ includes the monolithic system’s state variable $x^e$, and the binary states $b^i(k)$ that signify the healthy/faulty condition of each fault mode. An illustration of the DPFFD algorithm’s architecture is depicted in Figure 2.

Assuming that the measurement noise sequences of the DNs are independent, the global likelihood function can be factorized as the product of the local likelihood distributions (48), (58)–(61) as:

$$p(Z(k)|X(k)) = \prod_{I=1}^{N} p(Z_I(k)|X_I(k))$$

(19)

The goal of the DPFFD algorithm is to estimate the full system’s state $X(k)$ sequentially, based on the local measurement vectors of each DN. This algorithm requires that the same set of particles $\{X_I^j(k)\}_{j=1}^{N_I} = \{X^i(k)\}_{i=1}^{N}$ are sampled at each iteration, hence, the synchronization of the DNs’ local random number generators is necessary. The synchronous operation of the DNs is relative limiting, however, it can be easily implemented by employing of random number generators with identical seeds that are initialized at the same points. Each DN calculates its local likelihood function based on its observation vector. Following (16) and the factorization of the global likelihood distribution given in (19), the particle weights are calculated as follows:

$$w_I^j(k) = w_I^j(k-1) \cdot \left( \prod_{J \in \{I,N\} \setminus \{I\}} p(Z_J(k)|X^i(k)) \right)$$

(20)

where $\prod_{J \in \{I,N\} \setminus \{I\}} p(Z_J(k)|X^i(k))$ represents the product of the local likelihood functions over node $I$ and its neighbors.

Applying the natural logarithm to the above product yields:

$$\log \left( \prod_{J \in \{I,N\} \setminus \{I\}} p(Z_J(k)|X^i(k)) \right) = \sum_{J \in \{I,N\} \setminus \{I\}} \log \left( p(Z_J(k)|X^i(k)) \right)$$

(21)

The above sum can be calculated in a distributed way by means of a distributed average consensus algorithm (62). This selection is computationally simple and guarantee the asymptotic convergence of the algorithm. The Metropolis weights for the graph $\mathcal{G}$ are then defined as follows:

$$W_{IJ} = \begin{cases} 1 & I \neq J \quad J \in \mathcal{N}_I \\ \max\{\eta_J(t), 1\} - \sum_{I' \in \mathcal{N}_J} W_{I'I'} & I = J \end{cases}$$

(23)

Table IV: Pseudo-code of the DPFFD Algorithm

| Function DPFFD |
|-----------------|
| **Inputs:** $X^i(k-1), w_I^j(k-1), \{Z_I(t)\}_{I=1}^{N}$ |
| **Outputs:** $X^i(k), w_I^j(k), \{b^i_J(k)\}_{J=1}^{M}$ |
| **Required:** i) RESAMPLE, ii) CONSENSUS |
| 1: for $I = 1 : N$ do |
| 2: for $i = 1 : N_I$ do |
| 3: $X^i(k) = \mathcal{F}(X^i(k-1), u^i(k-1), \nu^i(k-1))$ do Particle update |
| 4: $w_I^j(k) = w_I^j(k-1) \cdot \exp \left( \text{CONSENSUS} [\eta_I(t)] \right)$ do Weights update |
| 5: $w_I^j(k) = \frac{w_I^j(k)}{\sum_{I=1}^{N} w_I^j(k)}$ do Weight normalization |
| 6: end for |
| 7: $[X^i(k), w_I^j(k)]_{I=1}^{N}$ = RESAMPLE $[X^i(k), w_I^j(k)]_{I=1}^{N}$ |
| 8: $X^i(k) = \sum_{i=1}^{N} w_I^j(k)X^i(k)$ do Minimum Mean-Square Error |
| 9: for $j = 1 : M$ do |
| 10: $b^i_J(k) = E[b^i_J(k)|Z_J(k)]$ do Probabilities of failure |
| 11: end for |
| 12: end for |

Figure 2. Illustration of the DPFFD algorithm’s structure.
$N_i$ iterations. In [62] it has been shown that consensus can be reached when $N_i$ is greater or equal to the diameter of the graph $G$.

The consensus filter plays the role of a local data fusion center. Its output relies on two factors: i) the type of the consensus algorithm; and ii) the communication graph topology between the DNs. Finally the particles’ weights at each DN are updated according to:

$$w_i^j(k) = w_i^j(k-1) \cdot \exp(\eta_i)$$

(24)

The rest of the algorithm’s step include weight normalization, resampling, and calculation of the MMSE. These steps are executed the same way as the the CPFFD algorithm. The block diagram of the DPFFFD algorithm is illustrated in Figure 3.

The outputs of the DNs are the probabilities of failure modes $b_Z^j(k) = E [ b_Z^j(k) | Z_i(k) ]$. It is important to note that due to the consensus filter that is embedded in each DN, we have $E [ b_Z^j(k) | Z_i(k) ] = E [ b_Z^j(k) | Z(k) ]$. This equality states that the probability of failure of each fault mode is the same at every DN since all the particles are synchronized at every node. Like the centralized case, this value is used to trigger alarm indicators if it exceeds a certain threshold $\alpha \in \{0, 1\}$ that marks the probability of detection. The pseudocodes of the DPFFFD and the consensus filter algorithms are listed in Tables IV and V respectively.

In terms of the computation load, the DPFFFD does not have any advantages compared to its centralized counterpart. The layout of the algorithm does not reduce the dimension of the data process. The main advantage of the DPFFFD algorithm is its distributed architecture and its ability to derive distributively global estimates based on local observations. A way to reduce the computation load is the use of DPF with a set membership [63]. Other techniques for reducing the computational and communication load of the DPFF algorithm are presented in [64–66].

V. NUMERICAL RESULTS

This Section provides an evaluation of the proposed DPFFFD algorithm via extensive numerical simulations. Two distinct cases are investigated that test the efficiency of the algorithm for different physical interconnections of the monolithic system and communication topologies of the DNs. 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 Figure 4. This process is consisted of ten identical cylindrical tanks of cross section area $S$. The tanks are connected to each other in a cascade configuration with pipes of cross section area $S_p$. A pump serves as a supply source to the first tank with a flow rate $Q_{in}$. The flow rate $Q_{i,j}$ between tank $i$ and tank $j$ is defined by means of Torricelli’s rule as:

$$Q_{i,j} = \mu_i \cdot S_p \cdot \text{sign}(x_i - x_j) \cdot \sqrt{2g|x_i - x_j|}$$

(25)

where $\mu_i$ 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 level of tank $i$ is determined by means of the mass balance equation as:

$$\dot{x}_i = \frac{1}{S_c} \left( \sum Q_{in}^i - \sum Q_{out}^i \right)$$

(26)

where $Q_{in}^i, Q_{out}^i$ are the inflow and outflow rates to tank $i$, respectively. The last tank of the sequence has a water drain outlet with flow rate $Q_{out}$ calculated by:

$$Q_{out} = \mu_{10} \cdot S_p \cdot \text{sign}(x_{10}) \cdot \sqrt{2g|x_{10}|}$$

(27)

The nominal values of the process’s parameters are given in Table VI and are based on a similar benchmark described in [67]. The inflow ($Q_{in}$) and outflow ($Q_{out}$) rate variables control the water level of the tanks. The fault modes under consideration are abrupt leaks to the water tanks. The dynamic
of a leakage at tank $j$ can be expressed as:

$$ g^j(x_j(k)) = \left( \frac{\mu_j \cdot S_j}{S} \right) \, \text{sign}(x_j(k)) \sqrt{2g \vert x_j(k) \vert} \tag{28} $$

From the mass balance equation and discretizing the continuous dynamics, the state-transition model of the ten tank system is described by:

$$ S : \begin{align*}
    x_1(k) &= x_1(k-1) + \frac{T_s}{S} \left( Q_{in} - Q_{1,2} \right) - b_2^1(k-1) \cdot g^1(x_1(k-1)) + v_1(k-1) \\
    x_i(k) &= x_i(k-1) + \frac{T_s}{S} \left( Q_{i-1,i} - Q_{i,i+1} \right) - b_2^i(k-1) \cdot g^i(x_i(k-1)) + v_1(k-1) \\
    x_{10}(k) &= x_{10}(k-1) + \frac{T_s}{S} \left( Q_{10} - Q_{out} \right) - b_2^{10}(k-1) \cdot g^{10}(x_{10}(k-1)) + v_{10}(k-1)
\end{align*} \tag{29} $$

where $T_s = 0.1$ sec is the sampling period, and the process noise $v_i \in \mathbb{R}$ with $i = 1, \ldots, 10$ is set to $\mathcal{N}(0,0.1)$. A diagnostic network of ten DNs, with one node for each tank, is deployed to monitor the process. The local observation of each DN is expressed by:

$$ z_I(k) = x_I(k) + \omega_I(k) \quad \text{with} \ I = 1, \ldots, 10 \tag{30} $$

where the stochastic noise $\omega_I$ is generated by the normal distribution $\mathcal{N}(0,0.1)$. A single fault is seeded at tank 5 when $k = 100$. The time horizon of the simulation is set to 200 time steps. The number of particles at each DN is set to $N_s = 100$. The initial conditions are configured such that the system starts at steady state. During the execution of the DPFFD algorithm the i.i.d noise, that drives the binary state, is generated by $\mathcal{U}(-0.6,0.6)$. Figure 5 shows the concentration of the particles on the $b_1$-$b_2$ plane during the healthy and faulty operating condition of the system at a given time instant. The selection of the 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 Figure 6. When the noise is $\mathcal{U}(-a,a)$ with $a = 0.5$ 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 function $\Phi(\cdot)$ of (14) serves as a "stochastic feedback" for the failure sensitive filter. When the system is healthy, the particles that indicate a fault would be diminished indirectly through the measurement likelihood function $\mathcal{L}(\cdot)$. A compromising value that ensures the optimal operation of the diagnostic filter was shown to be $a = 0.6$.

The second factor with a dominant impact to the performance of the algorithm is the topology of the communication graph $\mathcal{G}$. The communication topology between the DNs is directly effecting the output of the consensus algorithm. The impact of the diagnostic network’s topology is demonstrated by inspecting two distinct cases: a complete graph (highest possible connectivity) and a path graph (lowest possible connectivity). A complete graph is an undirected graph where every pair of individual DNs is connected by a single link. In the path graph only two terminal nodes have a degree of one, while all others have a degree of two. More information about graph theoretic concepts can be found in [57]. The detection threshold $\alpha$ is set to 0.8.

The probability of detection and the estimated state of the system, for the path graph topology $\mathcal{G}$, are illustrated in Figures 7 and 8 respectively. Since all DNs are synchronized by the consensus protocol, their outputs are identical (see Section IV). Hence, only the output of DN 1 is displayed in Figures 7 and 8. As it can be seen from Figure 9, the fully-connected network exhibits a faster diagnostic response compared to the path graph case. This figure shows a snapshot of the probability of detection for tank 5, limiting the time interval to $[90,150]$.
Figure 7. *First case study*: Probabilities of failure for each fault mode. The dashed black, the solid horizontal red, and the solid blue lines refer to the probability of failure, the detection threshold, and the occurrence of the abrupt fault, respectively.

Figure 8. *First case study*: Actual (solid blue line) and estimated (dashed red line) tank levels.

![Table VII](image)

| Approach               | RMSE        |
|------------------------|-------------|
| CPFFD                  | 0.0321      |
| DPFFD: Fully-connected graph | 0.03379    |
| DPFFD: Path graph      | 0.03630     |

Obviously, higher connectivity of the communication graph $\mathcal{G}$ results in faster convergence of the consensus algorithm. In terms of performance, the CPFFD constitutes an upper threshold. Table VII summarizes the RMSE values of the CPFFD and DPFFD algorithms, for two different topologies of the latter. As expected, the performance deteriorates as the connectivity of $\mathcal{G}$ is reduced.

The second simulation involves the six water tanks system of Figure 10. In this scenario, the subcomponents (water tanks) have more physical interconnections compared to the first case. The description of the discrete-time state-transition model is omitted due to space confinement. A DN is assigned to each water tank. Every DN can observe the water level of its assigned tank and exchange information with each neighboring nodes. The observation equation of each DN is same as before (30). Also, the communication topology of the diagnostic network matches the physical interconnections of the process.

An abrupt leak is seeded to every tank at the time instances listed in Table VIII. The nominal values of the system parameters and model/process noise models are identical to the first scenario (Table VI). The time horizon of the simulation is set to 280 time steps. The same tuning guidelines with the first example hold for this case study as well. Contrary to the first example, the second system is initiated from a transient state.

The probabilities of failure for each fault mode and the estimates states of the system are illustrated in Figures 11 and 12 respectively. These Figures indicate the outputs of DN 1. As mentioned earlier, the diagnostic network is synchronized so every node concludes to the same outlets. The results are deemed satisfactory since each DN can promptly detect and isolate every fault mode in spite of having access to local information. It is important to highlight that each DN can detect the faults that occur in other subcomponents as well.
as the faults corresponding to its monitored parts.

VI. CONCLUSION

We have presented a distributed implementation of a fault detection and isolation algorithm for nonlinear large-scale systems. A network of interconnected diagnostic nodes is employed to monitor the monolithic process. Each node has partial access of the process’s state. The information exchange between the diagnostic nodes allows the global inference of the system’s health by relying exclusively on local measurements. A distributed version of the particle filtering method is applied for both state estimation as well as statistical characterization of each fault mode. An online hypothesis testing module is embedded at every failure sensitive filter that triggers alarm indicators in the presence of a fault. The information fusion between the nodes take place using an average consensus protocol that synchronizes the local likelihood functions of the entire network. This inference component eliminates the need for a bank of estimators to isolate the occurring faults. This approach relieves the filter design analysis by substituting the complex stability proofs, that are required by observed-based methods, with a Monte Carlo simulations method that is conveniently applicable to real-life sensor networks.

REFERENCES

[1] M. Blanke, “Diagnosis and fault-tolerant control,” Springer, 2003.
[2] J. Gertler, “Fault detection and diagnosis in engineering systems,” CRC press, 1998.
[3] R. Isermann, “Fault-diagnosis systems,” Springer, 2006.
[4] A.S. Willsky, “A survey of design methods for failure detection in dynamic systems,” Automatica, vol. 12, no. 6, pp. 601–611, 1976.
[5] J.J. Gertler, “Survey of model-based failure detection and isolation in complex plants,” IEEE Control System Magazine, vol. 8, no. 6, pp. 3–11, 1988.
[6] P.M. Frank, “Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results,” Automatica, vol. 26, no. 3, pp. 459–474, 1990.
[7] J. Chen and R.J. Patton, “Robust model-based fault diagnosis for dynamic systems,” Kluwer academic publishers, 1999.
[8] V. Venkatasesubramanian, R. Rengaswamy, K. Yin, and S.N. Kavuri, “A review of process fault detection and diagnosis part i: Quantitative model-based methods,” Computers and Chemical Engineering, vol. 27, no. 3, pp. 293–311, 2003.
[9] S.X. Ding, “Model-based fault diagnosis techniques: design schemes, algorithms, and tools,” Springer Science & Business Media, 2008.
[10] E. Chow and A.S. Willsky, “Analytical redundancy and the design of robust failure detection systems,” IEEE Transactions on Automatic Control, vol. 29, no. 7, pp. 603–614, 1984.
[11] D.G. Luenberger, “Observing the state of a linear system,” IEEE Transactions on Military Electronics, pp. 74–80, 1964.
D.G. Luenberger, “Observers for multivariable systems,” IEEE Transactions on Automatic Control, vol. 11, pp. 190–197, 1966.

R.V. Beard, “Failure accommodation in linear systems through self-reorganization,” Ph.D. dissertation, Massachusetts Institute of Technology, 1971.

H.L. Jones, “Failure detection in linear systems,” Ph.D. dissertation, Massachusetts Institute of Technology, 1973.

R. Clark, “Instrument fault detection,” IEEE Transactions on Aerospace Electronic Systems, vol. 14, pp. 456–465, 1978.

J. Wünnenberg, “Observer-based fault detection in dynamic systems,” VDI-Verlag, 1990.

M. Kinnairt, “Robust fault detection based on observers for bilinear systems,” Automatica, vol. 35, no. 11, pp. 1829–1842, 1999.

C. De Persis and A. Isidori, “A geometric approach to nonlinear fault detection and isolation,” IEEE Transactions on Automatic Control, vol. 46, no. 6, pp. 853–865, 2001.

A.S. Willsky and H.L. Jones, “A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems,” Automatic Control, IEEE Transactions on, vol. 21, no. 1, pp. 108–112, 1976.

M. Bassesteve and A. Benveniste, “Detection of abrupt changes in signals and dynamic systems,” Lecture Notes in Control and Information Sciences, vol. 77, 1986.

R.K. Mehra and J. Peschon, “An innovations approach to fault detection and diagnosis in dynamic systems,” Automatica, vol. 7, no. 5, pp. 637–640, 1971.

A.J. Volponi, H. DePold, R. Ganguli, and C. Daguang, “The use of kalman filter and neural network methodologies in gas turbine performance diagnostics: a comparative study,” Journal of Engineering for Gas Turbines and Power, vol. 125, no. 4, pp. 917–924, 2003.

T. Kobayashi and D.L. Simon, “Application of a bank of Kalman filters for aircraft engine fault diagnostics,” in Proceedings of the ASME International Joint Power Generation Conference, 2003, pp. 461–470.

K. Salahshoor, M. Mosallaei, and M. Bayat, “Centralized and decentralized process and sensor fault monitoring using data fusion based on adaptive extended kalman filter algorithm,” Measurement, vol. 41, no. 10, pp. 1059–1076, 2008.

A. Serrao and K. Salahshoor, “Fault diagnosis and accommodation of nonlinear systems based on multiple-model adaptive unscented kalman filter and switched mpc and h-infinity loop-shaping controller,” Journal of Process Control, vol. 22, no. 3, pp. 626–634, 2012.

B.D. Anderson and J.B. Moore, “Optimal filtering,” Courier Dover Publications, 2012.

V. Kadirkamanathan, P. Li, M. Jaward, and S. Fabri, “A sequential monte carlo filtering approach to fault detection and isolation in nonlinear systems,” in Proceedings of the 39th IEEE Conference on Decision and Control, 2000, pp. 4331–4346.

N.J. Gordon, D.J. Salmond, and A.F. Smith, “Novel approach to nonlinear/non-gaussian bayesian state estimation,” in IEE Proceedings of Radar and Signal Processing, 1993, pp. 107–113.

P.M. Djurić, J.H. Kotecha, I. Zhang, Y. Huang, T. Ghirmai, M.F. Bugallo, and J. Míguez, “Particle Filtering,” Signal Processing Magazine, vol. 20, no. 5, pp. 19–38, 2003.

V. Kadirkamanathan, P. Li, M.H. Jaward, and S.G. Fabri, “Particle filtering-based fault detection in non-linear stochastic systems,” International Journal of Systems Science, pp. 259–265, 2002.

M.E. Orchard, “A particle filtering-based framework for on-line fault diagnosis and failure prognosis,” Ph.D. dissertation, Georgia Institute of Technology, 2007.

M.E. Orchard and G.J. Vachtsevanos, “A particle-filtering approach for on-line fault diagnosis and failure prognosis,” Transactions of the Institute of Measurement and Control, vol. 31, no. 3-4, pp. 221–246, 2009.

P. Baroni, G. Lamperti, P. Pogliano, and M. Zanella, “Diagnosis of large active systems,” Artificial Intelligence, vol. 110, no. 1, pp. 135–183, 1999.

I. Rish, M. Brodie, S. Ma, N. Odintsova, A. Beygelzimer, G. Grabarnik, and K. Hernandez, “Adaptive diagnosis in distributed systems,” IEEE Transactions on Neural Networks, vol. 16, no. 5, pp. 1088–1109, 2005.

T. Le and C.N. Hadjicostis, “Graphical inference methods for fault diagnosis based on information from unreliable sensors,” in Proceedings of the 9th International IEEE Conference on Control, Automation, Robotics, and Vision, 2006, pp. 1–6.

X.G. Yan and C. Edwards, “Robust decentralized actuator fault detection and estimation for large-scale systems using a sliding mode observer,” International Journal of Control, vol. 81, no. 4, pp. 591–606, 2008.

X. Zhang and Q. Zhang, “Distributed fault diagnosis in a class of interconnected nonlinear uncertain systems,” International Journal of Control, vol. 85, no. 11, pp. 1644–1662, 2012.

R.M.G. Ferrari, T. Parisini, and M.M. Polycarpou, “Distributed fault detection and isolation of large-scale discrete-time nonlinear systems: An adaptive approximation approach,” IEEE Transactions on Automatic Control, vol. 57, no. 2, pp. 275–290, 2012.

F. Boem, R.M. Ferrari, T. Parisini, and M.M. Polycarpou, “Distributed fault diagnosis for continuous-time nonlinear systems: The input–output case,” Annual Reviews in Control, vol. 37, no. 1, pp. 163–169, 2013.

I. Shames, A.M. Teixeira, H. Sandberg, and K.H. Johansson, “Distributed fault detection for interconnected second-order systems,” Automatica, vol. 47, no. 12, pp. 2757–2764, 2011.

M. J. Daigle, X.D. Koutsoukos, and G.Bovas, “Distributed diagnosis in formations of mobile robots,” IEEE Transactions on Robotics, vol. 23, no. 2, pp. 353–369, 2007.

F. Boem, R.M. Ferrari, T. Parisini, and M.M. Polycarpou, “A distributed fault detection methodology for a class of large-scale uncertain input-output discrete-time nonlinear systems,” in Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference, 2011, pp. 897–902.

E. Franco, R. Olfati-Saber, T. Parisini, and M.M. Polycarpou, “Distributed fault diagnosis using sensor networks and consensus-based filters,” in Proceedings of the 45th IEEE Conference on Decision and Control, 2006, pp. 386–391.

R. Olfati-Saber, “Kalman-consensus filter: Optimality, stability, and performance,” in Proceedings of the 48th IEEE Conference on Decision and Control, 2009, pp. 7036–7042.

R. Olfati-Saber, “Distributed Kalman filtering for sensor networks,” in Proceedings of the 46th IEEE Conference on Decision and Control, 2007, pp. 5492–5498.

Q. Cheng, P.K. Varshney, J. Michels, and C.M. Belcastro, “Distributed fault detection via particle filtering and decision fusion,” in Proceedings of the 8th IEEE International Conference on Information Fusion, 2005, pp. 8–pp.

Q. Cheng, P.K. Varshney, J.H. Michels, and C.M. Belcastro, “Distributed fault detection with correlated decision fusion,” IEEE Transactions on Aerospace and Electronic Systems, vol. 45, no. 4, pp. 1448–1465, 2009.

O. Hlinka, F. Hlawatsch, and P.M. Djuric, “Distributed particle filtering in agent networks: A survey, classification, and comparison,” IEEE Signal Processing Magazine, vol. 30, no. 1, pp. 61–81, 2013.

O. Hlinka, “Distributed particle filtering in networks of agents,” Ph.D. dissertation, Vienna University of Technology, 2012.

A. Mohammadi, “Distributed implementation of the particle filter with performance bounds,” Ph.D. dissertation, York University, 2013.

A. Mohammadi and A. Asif, “A consensus/fusion based distributed implementation of the particle filter,” in 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, 2011, pp. 285–288.

B. Ristic, S. Arulampalam, and N. Gordon, “Beyond the Kalman filter: Particle filters for tracking applications,” Artech house, 2004.

A. Doucet, S. Godsill, and C. Andrieu, “On sequential monte carlo sampling methods for bayesian filtering,” Statistics and computing, vol. 10, no. 3, pp. 197–208, 2000.

M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” IEEE Transactions on Signal Processing, vol. 50, no. 2, pp. 174–188, 2002.

I.A. Raptis and G. Vachtsevanos, “An adaptive particle filtering-based framework for real-time fault diagnosis and failure prognosis of environmental control systems,” in Proceedings of the Prognostics and Health Management, 2011.

I.A. Raptis, C. Sconyers, R. Martin, R. Mah, N. Oza, D. Mavris, and G.J. Vachtsevanos, “A particle filtering-based framework for real-time fault diagnosis of autonomous vehicles,” in Annual Conference of the Prognostics and Health Management Society, 2013.

M. Mesbahi and M. Egerstedt, Graph theoretic methods in multiagent networks,” Princeton University Press, 2010.

M. Coates, “Distributed particle filters for sensor networks,” in Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks, IEEE Press, 2004, p. 24.
[60] D. Gu. “Distributed particle filter for target tracking,” in Proceedings of the IEEE International Conference on Robotics and Automation, 2007, pp. 3856–3861.

[61] D. Gu, J. Sun, Z. Hu, and H. Li, “Consensus based distributed particle filter in sensor networks,” in Proceedings of the International IEEE Conference on Information and Automation, 2008, pp. 302–307.

[62] L. Xiao, S. Boyd, and S. Lall, “A scheme for robust distributed sensor fusion based on average consensus,” in Fourth International IEEE Symposium on Information Processing in Sensor Networks, 2005, pp. 63–70.

[63] S. Farahmand, S.I. Roumeliotis, and G.B. Giannakis, “Set-membership constrained particle filter: Distributed adaptation for sensor networks,” IEEE Transactions on Signal Processing, vol. 59, no. 9, pp. 4122–4138, 2011.

[64] D. Ustebay, M. Coates, and M. Rabbat, “Distributed auxiliary particle filters using selective gossip,” in Proceedings of the IEEE/SP International Conference on Acoustics, Speech and Signal Processing, 2011, pp. 3296–3299.

[65] O. Hlinka, O. Sluciak, F. Hlawatsch, P.M. Djuric, and M. Rupp, “Likelihood consensus: Principles and application to distributed particle filtering,” in Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers, 2010, pp. 349–353.

[66] O. Hlinka, O. Sluciak, F. Hlawatsch, P.M. Djuric, and M.Rupp, “Likelihood consensus and its application to distributed particle filtering,” IEEE Transactions on Signal Processing, vol. 60, no. 8, pp. 4334–4349, 2012.

[67] X. Zhang, M.M. Polycarpou, and T. Parisini, “A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems.” IEEE Transactions on Automatic Control, vol. 47, no. 4, pp. 576–593, 2002.