Modeling, Identification and Control, Vol. 35, No. 4, 2014, pp. 303–315, ISSN 1890–1328

A Unified Framework for Fault Detection and Diagnosis Using Particle Filter

Bo Zhao [1] Roger Skjetne [1]

[1] *Department of Marine Technology, Norwegian University of Science and Technology, N-7491 Trondheim, Norway.*
E-mails: bo.max.zhao@gmail.com, roger.skjetne@ntnu.no

Abstract

In this paper, a particle filter (PF) based fault detection and diagnosis framework is proposed. A system with possible faults is modeled as a group of hidden Markov models representing the system in fault-free mode and different failure modes, and a first order Markov chain is modeling the system mode transitions. A modified particle filter algorithm is developed to estimate the system states and mode. By doing this, system faults are detected when estimating the system mode, and the size of the fault is diagnosed by estimating the system state. A new resampling method is also developed for running the modified PF efficiently.

Two introductory examples and a case study are given in detail. The introduction examples demonstrate the manner to model a system with possible faults into hidden Markov model and Markov chain. The case study considers a numerical model with common measurement failure modes. It focuses on the verification of the proposed fault diagnosis and detection algorithm and shows the behavior of the particle filter.

Keywords: Fault detection and diagnosis, Particle filter, Hidden Markov model, Markov chain

1 Introduction

1.1 Fault Detection and Diagnosis

Due to the increasing requirements of safety, reliability, and performance of control systems, a conventional feedback control design for a sophisticated system may result in an unsatisfactory performance, or even instability, in the event of failures in actuators, sensors, or other system components. Thus, to maintain the performance of the system, or to let the system work in degraded but safe condition when failures occur is particularly important for safety-critical systems such as aircrafts, submarines, nuclear power stations, chemical plants, and so on. In such systems, the consequences of a minor fault in a component or any loss of system functionality can be catastrophic. Therefore, the demand on reliability, safety, and fault tolerance is high.

Towards realizing a fault tolerant system, the first step is to diagnose and detect faults. Hence, the control engineers and system designers started to add fault diagnosis and detection (FDD) functions to the control system. There are two approaches to achieve FDD, analytical approach and heuristic approach. The analytical approach further consists of two methods – signal- processing-based and model-based (Isermann, 2006). Signal-processing- based method considers the time domain (statistical) and frequency domain features of output signals. Basically, the threshold, amplitude, mathematical expectation, variance, correlation, and frequency spectrum (Fourier analysis) must be inspected in the signal-processing-based method. In addition, advanced analytical methods such as wavelet analysis, intelligent analysis (neural networks and fuzzy logic), cluster analysis, and some other methods from pattern recognition may also be helpful.

Model-based analytical FDD method is discussed in

doi:10.4173/mic.2014.4.7

detail in Blanke et al. (2006). A system model, usually a state estimator such as a Kalman filter, is introduced into this scheme. Then the system outputs is compared with the estimated outputs from the observer. FDD results can be deduced from the comparison residual.

This paper follows the model-based analytical FDD idea, and it makes efforts to extend the existing model-based methods into a more general framework by using the advantages of particle filters.

Particle filter (PF), which is derived from Bayesian estimation and Monte Carlo methods, has been studied for about two decades. The idea of a particle filter was first introduced by Gordon et al. (1993), then enriched by Kitagawa (1996), Liu and Chen (1998), and many other researchers. A good early tutorial is by Arulampalam et al. (2002), and recently summarized by Doucet and Johansen (2009). The PF has drawn great attention since it was proposed because it is a powerful tool to solve the optimal estimation problems in nonlinear non-Gaussian systems. The PF has been applied in target tracking, computer vision, digital communications, speech recognition, machine learning, and other areas; see Chen (2003) for more detail. The PF has also shown its strong capability to solve state estimation problem in nonlinear systems with non-Gaussian noise. This article trends to extend the usage of PF to solve FDD problems.

1.2 An Introductory Idea - Fault Detection and Diagnosis with Filter Bank

In Gustafsson (2001), a linear system with possible faults can be modeled as a time varying linear system, whose dynamic is described by[1]

$$\begin{aligned}
\boldsymbol{x}_{k+1} &= \boldsymbol{A}_k(\delta_k)\boldsymbol{x}_k + \boldsymbol{B}_{u,k}(\delta_k)\boldsymbol{u}_k \\
&\quad + \boldsymbol{B}_{w,k}(\delta_k)\boldsymbol{w}_k && (1)\\
\boldsymbol{y}_k &= \boldsymbol{C}_k(\delta_k)\boldsymbol{x}_k + \boldsymbol{D}_{u,k}(\delta_k)\boldsymbol{u}_k + \boldsymbol{v}_k && (2)\\
\boldsymbol{w}_k &\sim \mathcal{N}\left(\boldsymbol{\mu}_{w,k}(\delta_k), \boldsymbol{Q}_k(\delta_k)\right) && (3)\\
\boldsymbol{v}_k &\sim \mathcal{N}\left(\boldsymbol{\mu}_{v,k}(\delta_k), \boldsymbol{R}_k(\delta_k)\right). && (4)
\end{aligned}$$

In this model, $\boldsymbol{x}$ is the state vector, $\boldsymbol{y}$ is the measurement, $\boldsymbol{A}$ is the state transition matrix, $\boldsymbol{B}_u$ is the input matrix, $\boldsymbol{B}_w$ is the noise input matrix, $\boldsymbol{C}$ is the measurement matrix, $\boldsymbol{D}_u$ is the direct input to measurement transition matrix, and k always denotes the current time instance. $\boldsymbol{v}$ and $\boldsymbol{w}$ are the measurement noise and system noise vectors, respectively. These noise terms are assumed to be white and Gaussian, but there can be a DC component embedded. $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$ is a multivariate Gaussian probability density function, and $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ is the mean vector and covariance matrix, respectively. The sign "$\sim$" denotes a random variable "subject to" a probability density function (PDF). All the parameter matrices can be time- varying and conditioned on the system mode parameter δ.

The parameter δ [2] is important in this model. It is called the system mode parameter because the system switches its behavior as δ changes its value if the parameter matrices are depending on δ nontrivially. This system mode parameter is used to distinguish the fault-free and faulty behavior of the system in the following FDD framework.

The model (1), (2) is linear Gaussian, so that it coincides with most of the assumptions in a Kalman filter, except for the system mode parameter δ. Then, the Kalman filter is still valid in this case, if assuming that δ is known. However, in FDD δ is the objective of the detection and diagnosis, so it is unknown to us. For these features of the model, a Kalman filter bank based FDD scheme is given in Gustafsson (2001). In this scheme, the FDD problem is reformed as an estimating problem of the system mode $\delta_{1:k}$, to obtain the maximum a posteriori (MAP) estimator

$$\left\{\hat{\delta}_{1:k}\right\}_{MAP} = \arg\max_{\delta_{1:k}} p\left(\delta_{1:k} \left| \boldsymbol{y}_{1:k}\right.\right). \tag{5}$$

However, this model has its limitations:

1. The model is linear, but the real systems are generally nonlinear. On the other hand, as shown in the previous work by Zhao et al. (2012b), sometimes we may have to think about nonlinear measurement problem. So it is necessary to extend this model to the nonlinear case.

2. The noise terms are assumed to be Gaussian in the model, but it does not always fit the real world. Also refer to Zhao et al. (2012a) and Zhao et al. (2014), although we can whiten a colored noise by state augmentation, we still have a Rice distributed measurement noise and a Gaussian mixture driving noise.

Focusing on these limitations in the model (1), (2), a generalized system model is defined as a combination of a hidden Markov model and a Markov chain. Several modeling examples are given in Section 3. Then a modified particle filter algorithm is proposed to estimate the system states and system mode in the generalized model in Section 4. Two case studies in different scenarios are given in Section 5 as validation of the proposed algorithm. Finally the conclusion of this paper is given.

[1] We modified the notations in Gustafsson (2001) in compliance with this paper.

[2] Regarding the notation in this paper, we use $a_{m:n}$ representing the sequence $a_m, a_{m+1}, \cdots, a_n$. And a single a is a generic reference to this time sequence, whose meaning can be understood from the context.

2 A Generalized Model

In this paper, a system is described by a switching mode hidden Markov model (HMM)

$$\begin{aligned} &\boldsymbol{X}_{k+1} \left| (\boldsymbol{X}_k = \boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\delta}_k) \right. \\ \sim & p\left(\boldsymbol{x}_{k+1} \left| \boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\delta}_k \right.\right) = f_k\left(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\delta}_k\right) \qquad (6) \\ &\boldsymbol{Y}_k \left| (\boldsymbol{X}_k = \boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\delta}_k) \right. \\ \sim & p\left(\boldsymbol{y}_k \left| \boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\delta}_k \right.\right) = h_k\left(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\delta}_k\right), \qquad (7) \end{aligned}$$

where (6) is the *system equation*, defining how the states propagate depending on the input, and system mode. Moreover:

- $\boldsymbol{X}$ is the *state vector*, a random vector in $\mathbb{R}^{N_x}$.
- $\boldsymbol{x}_{k+1}$ and $\boldsymbol{x}_k$ are the realizations of the corresponding state vectors.
- $\boldsymbol{u}_k \in \mathbb{R}^{N_u}$ is the *known input* to the system at time k.
- $\boldsymbol{\delta}_k = \left[\left(\boldsymbol{\Delta}^0\right)^\top \left(\boldsymbol{\Delta}^1\right)^\top \cdots \left(\boldsymbol{\Delta}^{N_m}\right)^\top \right]^\top$ is a discrete parameter representing the *system mode*, where $\boldsymbol{\Delta}^i \in \{0,1\}$ is a binary representation of the occurrence of a fault, such that $\boldsymbol{\delta}_k \in \{0,1\}^{N_m}$. The detailed discussion about this system mode parameter is given in the next section.
- $f_k(\cdot) : \mathbb{R}^{N_x} \times \mathbb{R}^{N_u} \times \{0,1\}^{N_m} \mapsto \mathbb{R}$ is the *state transition mapping*, which maps the states, input, and system mode at current time to the PDF of the states at the next time instance. And $p(\cdot)$ is a probability measure on $\mathbb{R}^{N_x}$.

Equation (7) is the *measurement equation* . It defines the relation from the current states, input, and system mode to the observation, where:

- $\boldsymbol{Y}_k \in \mathbb{R}^{N_x}$ is a random vector, representing the observation.
- $\boldsymbol{y}_k \in \mathbb{R}^{N_y}$ is the measurement at time instance k.
- $h_k(\cdot) : \mathbb{R}^{N_x} \times \mathbb{R}^{N_u} \times \{0,1\}^{N_m} \mapsto \mathbb{R}$ is the *measurement mapping*, which maps the states, input, and system mode at current time to the PDF of the measurement. This mapping corresponds to the measurement equation in the state space model. And $p(\cdot)$ is a probability measure on $\mathbb{R}^{N_y}$.

This model is a generalization of (1), (2) since it can describe a nonlinear system subject to non-Gaussian noise conditions. For example, the model (1), (2) can be written in the HMM form (6) and (7) according to

$$\begin{aligned} & p\left(\boldsymbol{x}_{k+1} \left| \boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\delta}_k \right.\right) \\ = \ & \mathcal{N}\Big(\boldsymbol{A}_k\left(\boldsymbol{\delta}_k\right)\boldsymbol{x}_k + \boldsymbol{B}_{u,k}\left(\boldsymbol{\delta}_k\right)\boldsymbol{u}_k + \boldsymbol{B}_{w,k}\left(\boldsymbol{\delta}_k\right)\boldsymbol{\mu}_{w,k}, \\ & \boldsymbol{B}_{w,k}\left(\boldsymbol{\delta}_k\right)\boldsymbol{w}_k\boldsymbol{B}_{w,k}^\top\left(\boldsymbol{\delta}_k\right)\Big) \qquad (8) \\ & p\left(\boldsymbol{y}_k \left| \boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\delta}_k \right.\right) \\ = \ & \mathcal{N}\Big(\boldsymbol{C}_k\left(\boldsymbol{\delta}_k\right)\boldsymbol{x}_k + \boldsymbol{D}_{u,k}\left(\boldsymbol{\delta}_k\right)\boldsymbol{u}_k + \boldsymbol{B}_{v,k}\left(\boldsymbol{\delta}_k\right)\boldsymbol{\mu}_{v,k}, \\ & \boldsymbol{B}_{v,k}\left(\boldsymbol{\delta}_k\right)\boldsymbol{v}_k\boldsymbol{B}_{v,k}^\top\left(\boldsymbol{\delta}_k\right)\Big), \qquad (9) \end{aligned}$$

The manner to solve the FDD problem by an HMM is by estimating the system mode sequence $\boldsymbol{\delta}$. For instance, assuming that the sequence $\boldsymbol{\delta}$ is estimated, suppose it is found that $\boldsymbol{\delta}_i$ is nontrivial for $i = k-l, \cdots, k$. It can then be concluded that a fault has occurred, and the fault happened between time instance $k-l-1$ and $k-l$.

It should be highlighted that although this model is very general, it is not universal. For instance, while (6) and (7) are discrete, physical systems are continuous. Discretization is necessary for the physical systems to fit the model. The cost we then pay is the discretization error.

It can be a problem that a system with its failure modes may be modeled in different ways within this framework. This implies that the model (6) and (7) is too general and flexible and it loses its functionality to be a canonical form. Hence, the model should be more specialized. More discussion regarding the canonical form will be given in the next section after some further examples and analysis.

3 Modeling Examples and Discussion

Modeling a system into the form (6) and (7) is not difficult, because the knowledge and methods of modeling a system into state space form can be directly inherited. Regarding nonlinearity and non-Gaussian noise, it is possible to inherit the nonlinearity of the system directly rather than linearizing the system like the extended Kalman filter, and also the general model is compatible with non-Gaussian noise by manufacturing the mappings. We give some examples to show how system with their possible faults are modeled in the HMM form. These examples are motivated by the examples in (Gustafsson, 2001).

3.1 Example 1: Detection in Changing Mean Model

3.1.1 One-time Changing Mean Model

Consider the case of an unknown DC component embedded in white noise. Suppose that we want to test the hypothesis that the DC component has been changed at some unknown time instant. We can then model the signal by

$$y_k = \theta_1 - \sigma(k-l+1)\,\theta_1 + \sigma(k-l+1)\,\theta_2 + v_k \quad (10)$$

where $\sigma(\cdot)$ is the step function, θ_1 and θ_2 are the DC components embedded in the white noise before and after the change, l is the change time, and v_k is the white measurement noise. If all possible change instants are to be considered, the variable l takes its value from the set $\{0, 1, \cdots, k-1, k\}$, where $l = k$ should be interpreted as no change.

We can rewrite (10) as

$$y_k = \mathbf{\Delta}_k \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} + v_k, \quad (11)$$

where the system mode parameter $\mathbf{\Delta}_k = \left[\Delta^1\, \Delta^2\right] \in \{[0\,1], [1\,0]\}$. This model is in the form of (7) in the sense that

$$p(y_k \,|\mathbf{\Delta}_k) = \mathbf{\Delta}_k \begin{bmatrix} \mathcal{N}(\theta_1, \mathrm{Var}(v_k)) \\ \mathcal{N}(\theta_2, \mathrm{Var}(v_k)) \end{bmatrix}, \quad (12)$$

where $\mathcal{N}(\mu, \sigma)$ is the scalar Gaussian probability density function.

However, the new model (11) is not equivalent to (10), because the system mode in (11) can switch multiple times between $[1\,0]$ and $[0\,1]$, while it can change only once in (10). So some restriction must be assigned to the system mode parameter in (11), such that δ can only change from $[1\,0]$ to $[0\,1]$, but never opposite. This can mathematically be described by the following Markov chain (see also Figure 1),

$$\begin{aligned}
&\Pr(\mathbf{\Delta}_k = [1\,0] \,|\mathbf{\Delta}_{k-1} = [1\,0]) = p_{11} && (13a)\\
&\Pr(\mathbf{\Delta}_k = [1\,0] \,|\mathbf{\Delta}_{k-1} = [0\,1]) = 0 && (13b)\\
&\Pr(\mathbf{\Delta}_k = [0\,1] \,|\mathbf{\Delta}_{k-1} = [1\,0]) = p_{12} = 1 - p_{11} && (13c)\\
&\Pr(\mathbf{\Delta}_k = [0\,1] \,|\mathbf{\Delta}_{k-1} = [0\,1]) = 1, && (13d)
\end{aligned}$$

where p_{11} is the probability that the value of the DC component remains the same between the time steps, while p_{12} is the probability that the value of the DC component change from θ_1 to θ_2 between time steps. The transition probabilities 0 and 1 are set according to the assumption that the DC component can change only once in the whole sequence.

The combined model of (11) and (13) is now almost equivalent to the model (10). However, the model (11) and (13) is more precise than (10). This is because the combined model also defines the system mode transition probabilities, while in (10) the probability of the mode switching is undefined, meaning that the a prior information p_{11} and p_{12} cannot be used in the estimation.

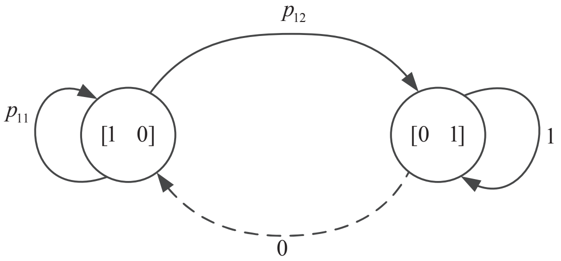

Figure 1: The Markov chain of mode transition.

3.1.2 Segmentation in Changing Mean Model

We can extend the above scenario to a more general case by relaxing the assumptions of previous knowledge of the DC values and the number of changes. In this case it is convenient to take the DC level as system state, and thereby model the signal as

$$\begin{aligned}
p(\theta_{k+1} \,|\theta_k, \mathbf{\Delta}_k) &= \mathbf{\Delta}_k \begin{bmatrix} \rho(\theta_{k+1} - \theta_k) \\ U(\theta_{\mathrm{inf}}, \theta_{\mathrm{sup}}) \end{bmatrix} && (14)\\
p(y_k \,|\theta_k) &= \mathcal{N}(\theta_k, \mathrm{Var}(v_k)), && (15)
\end{aligned}$$

where $\rho(\cdot)$ is the Dirac measure, $U(\theta_{\mathrm{inf}}, \theta_{\mathrm{sup}})$ is the uniform distribution $U(\theta_{\mathrm{inf}}, \theta_{\mathrm{sup}})$, and $\mathbf{\Delta}_k$ follows the definition in the previous example. $\mathbf{\Delta}_k = [1\,0]$ represents there is no change at time k, while $\mathbf{\Delta}_k = [0\,1]$ represents there is a change at time k.

The mode transition can be defined according to the frequency of the mode change, such as

$$\begin{cases} \Pr(\mathbf{\Delta}_k = [0\,1]) = p_1 \\ \Pr(\mathbf{\Delta}_k = [1\,0]) = p_2 = 1 - p_1. \end{cases} \quad (16)$$

3.2 Example 2: Decayed Input

In a control system, the output from actuators can be less than the required value, typically due to actuators' aging problem. This input reduction may happen suddenly, but changes mildly once occurred. We can model this fault as

$$\begin{aligned}
p(\boldsymbol{x}_{k+1} \,|\boldsymbol{x}_k, \boldsymbol{\alpha}_k, \boldsymbol{u}_k, \mathbf{\Delta}_k) &= f(\boldsymbol{x}_k, \boldsymbol{\alpha}_k, \boldsymbol{u}_k) && (17)\\
p(\boldsymbol{\alpha}_{k+1} \,|\boldsymbol{\alpha}_k, \mathbf{\Delta}_k) &= \mathbf{\Delta}_k \begin{bmatrix} \mathbf{1} \\ g(\boldsymbol{\alpha}_k) \end{bmatrix} && (18)\\
p(\boldsymbol{y}_k \,|\boldsymbol{x}_k, \boldsymbol{\alpha}_k, \boldsymbol{u}_k) &= h(\boldsymbol{x}_k, \boldsymbol{\alpha}_k, \boldsymbol{u}_k), && (19)
\end{aligned}$$

where $\boldsymbol{\alpha}$ is a vector, whose entries are between 0 and 1, indicating the reduction rate of each channel of the system input. The dynamic of $\boldsymbol{\alpha}$ is modeled in (18), where $g(\boldsymbol{\alpha}_k)$ can be adjusted in a manner such that the entries in $\boldsymbol{\alpha}_{k+1}$ is less than or equal to the entries in $\boldsymbol{\alpha}_k$. $\boldsymbol{\Delta}_k = \left[\Delta^1\,\Delta^2\right] \in \{[0\,1],[1\,0]\}$, where $\boldsymbol{\Delta}_k = [1\,0]$ represents that the input channels are fault-free, while $\boldsymbol{\Delta}_k = [0\,1]$ represents that there is an input reduction.

3.3 Discussion of the General Model

We recall the general model

$$\begin{aligned} p\left(\boldsymbol{x}_{k+1}\left|\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\delta}_k\right.\right) &= f_k\left(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\delta}_k\right) & (20)\\ p\left(\boldsymbol{y}_k\left|\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\delta}_k\right.\right) &= h_k\left(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\delta}_k\right). & (21)\end{aligned}$$

As shown in the previous examples, it is necessary and beneficial to model the transition of the system mode parameter to obtain a more accurate and more informative model. On the other hand, in both examples mentioned above, we are considering single failure cases. In practice, however, we may encounter multiple failure cases when different faults occurs at the same time. For this sake, we can assign a state $\Delta^{f^{(i)}} \in \{0,1\}$ denoting whether fault “$f^{(i)}$” occurs in the system, so that the combination of states $\boldsymbol{\Delta} = \left[\Delta^{f^{(i)}}\,\Delta^{f^{(2)}}\,\ldots\Delta^{f^{(N_m)}}\right] \in \{0,1\}^{N_m}$ describes the failures in the whole system. Note that here we are using capital characters $\boldsymbol{\Delta}$ and Δ for the discrete random vector/variable of the system mode, while their realizations will be denoted as $\boldsymbol{\delta}$ and δ. The Markov chain

$$\Pr\left(\boldsymbol{\Delta}_{k+1} = \boldsymbol{\delta}^i\left|\boldsymbol{\Delta}_k = \boldsymbol{\delta}^j\right.\right) = p_{ji,k} \qquad (22)$$

is suitable to model the characteristics of these faults.

It is not difficult to combine the Markov chain model of the mode transition into the HMM of the states. First we define the extended state vector, which consists of the system state and the system mode parameter,

$$\boldsymbol{\xi}_k = \left[\boldsymbol{x}_k^\top \quad \boldsymbol{\delta}_k^\top\right]^\top. \qquad (23)$$

Then the uniform system model can be obtained by reforming equations (20), (21), and (22) as

$$\begin{aligned} p\left(\boldsymbol{\xi}_{k+1}\left|\boldsymbol{\xi}_k, \boldsymbol{u}_k\right.\right) &= \begin{bmatrix} p\left(\boldsymbol{x}_{k+1}\left|\boldsymbol{\xi}_k, \boldsymbol{u}_k\right.\right) \\ \Pr\left(\boldsymbol{\Delta}_{k+1} = \boldsymbol{\delta}^i\left|\boldsymbol{\Delta}_k = \boldsymbol{\delta}^j\right.\right)\end{bmatrix} \\ &= \begin{bmatrix} f_k\left(\boldsymbol{\xi}_k, \boldsymbol{u}_k\right) \\ p_{ji,k}\end{bmatrix} & (24)\\ p\left(\boldsymbol{y}_k\left|\boldsymbol{\xi}_k, \boldsymbol{u}_k\right.\right) &= h_k\left(\boldsymbol{\xi}_k, \boldsymbol{u}_k\right), & (25)\end{aligned}$$

where $i, j \in \{1, \cdots, N_m\}$. The model (24)) and (25) will be used in the following derivation of the PF-based FDD algorithm.

4 Particle Filter for Fault Detection and Diagnosis

4.1 The Algorithm of the PF for FDD

The algorithm of the PF proposed here is adapted from the *sampling importance resampling* (SIR) PF in Arulampalam et al. (2002). Note, however, that different Monte Carlo based Bayesian estimation algorithms can be used to solve the estimation problem for (24) and (25). The SIR PF presented here is our prototype. Figure 2 shows the process of the PF algorithm.

4.1.1 Inherit from the Last Cycle

The PF works in a recursive manner. At time k, it inherits $p\left(\boldsymbol{\xi}_{k-1}|\boldsymbol{y}_{1:k-1}\right)$, which is the posterior approximation of the distribution of the extended state vector given the observation up to the previous time instance $k-1$. In this PF, because there are both continuous and discrete states in the system, the posterior distribution $p\left(\boldsymbol{\xi}_{k-1}|\boldsymbol{y}_{1:k-1}\right)$ is expressed by

$$p\left(\boldsymbol{\xi}_{k-1}|\boldsymbol{y}_{1:k-1}\right) \approx \sum_{i=1}^{N_{s,k-1}} w_{k-1}^{(i)} \cdot \rho\left(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{(i)}\right) \cdot \rho_{\boldsymbol{\delta}_{k-1}, \boldsymbol{\delta}_{k-1}^{(i)}}, \qquad (26)$$

where $N_{s,k-1}$ is the number of particles at time instance $k-1$, $\rho(\cdot)$ is the Dirac function, and $\rho_{i,j}$ is the Kronecker delta function. A conceptual illustration of the posterior can be seen in Figure 2.

4.1.2 Time Update

The *time update* process is to obtain the prior estimation of the states as $p\left(\boldsymbol{\xi}_k\left|\boldsymbol{y}_{1:k-1}\right.\right)$. In the PF context, this process is done by draw samples from a so-called *importance density*. The SIR PF uses the most convenient importance density $p\left(\boldsymbol{\xi}_k\left|\boldsymbol{\xi}_{k-1}^{(i)}\right.\right)$ defined by (24) to derive the prior density.

Since there is a dependence in (24), the time update process has to be divided into two steps - the *system mode time update* corresponding to Action 1 in Figure 2, and the *system states time update* as Action 2. Intuitively, the system mode should update first, and then the system states are updated since they are mode-dependent. That is, for each particle, we determine its mode at current time instance by drawing sample from the mode transition Markov Chain, and then we determine the system states by drawing sample from $p\left(\boldsymbol{x}_k\left|\boldsymbol{\xi}_{k-1}^{(i)}, \boldsymbol{u}_k\right.\right) = f_k\left(\boldsymbol{\xi}_{k-1}^{(i)}, \boldsymbol{u}_k\right)$. The distribution

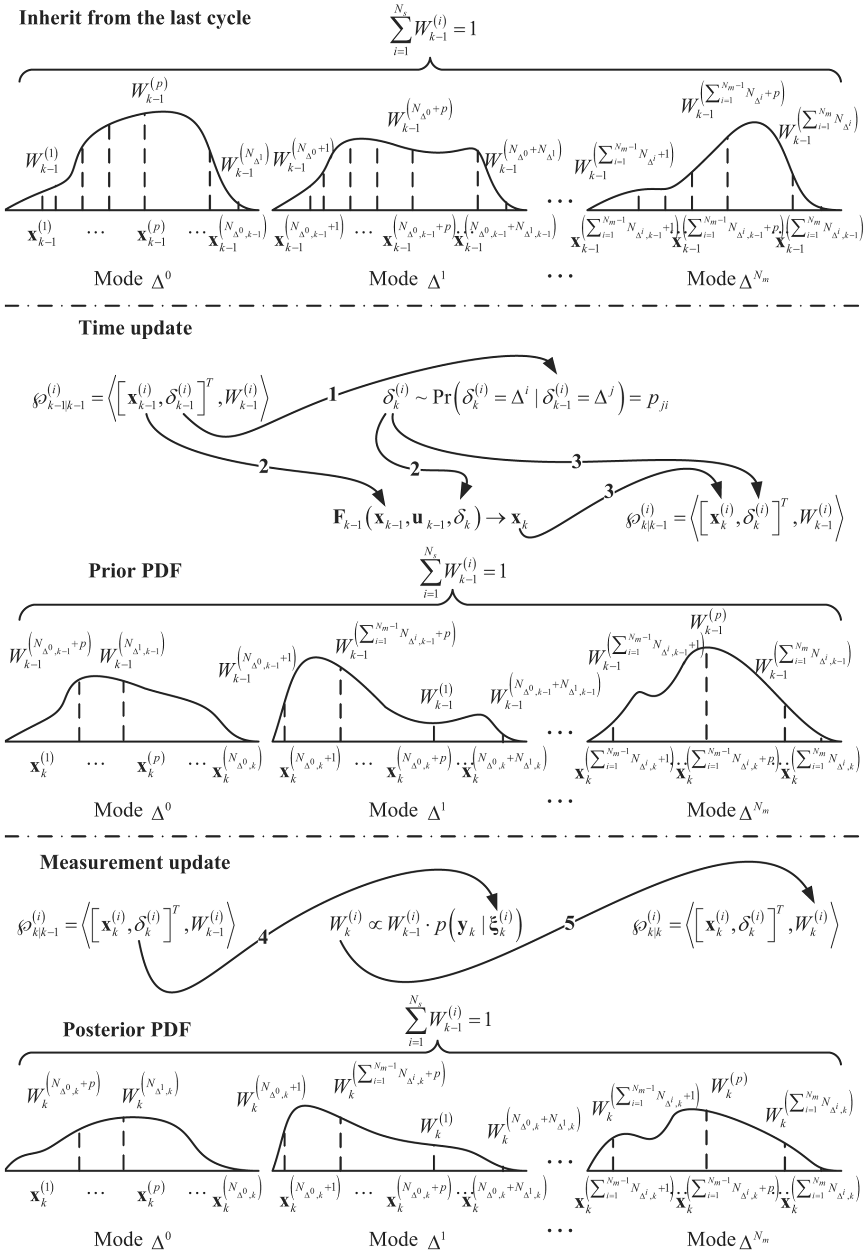

Figure 2: One cycle of the PF, divided into 3 steps by chain lines. These steps correspond to the inherit, time update, and measurement update, respectively. The resampling step is not included in this figure.

that we draw sample from is then equivalent to

$$\begin{aligned} & p\left(\boldsymbol{\xi}_k \left| \boldsymbol{\xi}_{k-1}^{(i)} \right.\right) \\ = \; & p\left([\boldsymbol{x}_k, \boldsymbol{\delta}_k]^\top \left| \left[\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{\delta}_{k-1}^{(i)}\right]^\top \right.\right) \\ = \; & p\left(\boldsymbol{x}_k \left| \left[\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{\delta}_{k-1}^{(i)}\right]^\top \right.\right) \cdot \Pr\left(\boldsymbol{\delta}_k \left| \left[\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{\delta}_{k-1}^{(i)}\right]^\top \right.\right) \\ = \; & p\left(\boldsymbol{x}_k \left| \left[\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{\delta}_{k-1}^{(i)}\right]^\top \right.\right) \cdot \Pr\left(\boldsymbol{\delta}_k \left| \boldsymbol{\delta}_{k-1}^{(i)} \right.\right), \qquad (27) \end{aligned}$$

where $\Pr\left(\boldsymbol{\delta}_k \left| \left[\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{\delta}_{k-1}^{(i)}\right]^\top \right.\right) = \Pr\left(\boldsymbol{\delta}_k \left| \boldsymbol{\delta}_{k-1}^{(i)} \right.\right)$, because $\boldsymbol{\delta}_k$ is independent from $\boldsymbol{x}_k$. At the end of the time update step, we obtain new positions of the particles $\boldsymbol{\xi}_k^{(i)}$.

4.1.3 Measurement Update

At this step the particle weights are updated according to the observation at the current time instance k. The weights are updated according to the Baye's law, in which case if one particle's prior $\boldsymbol{\xi}_k^{(i)}$ is supported by the observation then the weight of the particle should increase, and vice versa.

Given the observation at current time as $\boldsymbol{y}_k$, and using the importance density $p\left(\boldsymbol{\xi}_k \middle| \boldsymbol{\xi}_{k-1}^{(i)}\right)$, this yields

$$w_k^{(i)} \propto w_{k-1}^{(i)} \cdot p\left(\boldsymbol{y}_k \middle| \boldsymbol{\xi}_k^{(i)}\right), \tag{28}$$

where $p\left(\boldsymbol{y}_k \middle| \boldsymbol{\xi}_k^{(i)}\right)$ is defined by (25). This corresponds to Action 5 in Figure 2.

4.1.4 A Modified Resampling Algorithm

Degeneracy is a common problem in PFs, where after a few iterations, all but one particle will have negligible weight (Arulampalam et al., 2002). Doucet et al. (2000) shows that the variance of the importance weights will increase over time, and thus it is impossible to avoid the degeneracy phenomenon. It is common to employ a resample step in the PF algorithm to counteract the degeneracy. Examples are the resample methods proposed in *multinomial resampling* in Smith and Gelfand (1992), *residual resampling* in Liu (1996), and *systematic resampling* in Carpenter et al. (1999).

However, these resampling methods are not suitable in our case. To understand the problem, consider (24) and assume that one mode has the marginal mass $\Pr\left(\Delta_k^{f^{(p)}} = 1\right) = 0.0001$ $(p \in \{1, \dots, N_m\})$. Then this mode rarely occurs in the system. To accurately describe the distribution, it is common to use more than $O\left(10^1\right)$ particles. So, if using 10 particles, which is a very low value, to represent the density $p\left(\boldsymbol{x}_k \middle| \Delta_k^{f^{(p)}} = 1\right)$, then in average $\frac{10}{0.0001} = 10^6$ particles in total are needed to describe the entire density $p\left(\boldsymbol{\xi}_k | \boldsymbol{y}_{1:k}\right)$. This large amount of particles is computationally infeasible.

A desired resampling algorithm should be able to generate a suitable amount of particles for each system mode. Since the probability of a fault to occur in a system is generally low, the particles in these modes will have light weights. With a standard resampling method, these light-weighted particles are unlikely to survive. Thus, the amount of particles regenerated by the resampling algorithm should make sure that the computational cost is not too high, at the same time as the particles from the system modes that are unlikely to happen will represent the conditional density of the states in these system modes.

For these reasons, an adaptive resampling method is proposed. This may give enough samples for the modes with small marginal probability mass, and at the same time it can restrict the total numbers of particles. Essentially this modified resampling algorithm must adaptively determine the number of samples in each system mode according to their significance, and then make a compromise between the computational complexity and the estimation performance.

At first, define $\breve{N}_s$ as the minimum number of particles required for sufficiently representing a system mode. Then there will be at least $\breve{N}_s N_m$ particles in the PF. So, if there are too many modes in the system, a large amount of particles cannot be avoided.

Consider those modes with significant marginal probability mass. If a system mode has a significant marginal probability mass, this means the system is more likely to work well in this mode. These modes are named *significant modes*. Define $\tilde{N}_s$ as the number of particles suitable for estimation. Then $\breve{N}_s\left(N_m - 1\right) + \tilde{N}_s$ should be an acceptable sample size for computing, as explained at the end of this section. Thus, we assign $\tilde{N}_s$ particles to each system mode according to their marginal probability mass,

$$N_{s,\boldsymbol{\delta}^{(p)},k} = \left\lceil \Pr\left(\boldsymbol{\Delta}_k = \boldsymbol{\delta}^{(p)} | \boldsymbol{y}_{1:k}\right) \cdot \tilde{N}_s \right\rceil,$$

where $\boldsymbol{\delta}^{(p)}$ goes through all elements in $\{0, 1\}^{N_m}$, $N_{s,\boldsymbol{\delta}^{(p)},k}$ is the number of particles required in mode $\boldsymbol{\delta}^{(p)}$ in the resampling step, and $\lceil a \rceil$ is the minimum integer which is larger than a. For any $N_{s,\boldsymbol{\delta}^{(p)},k} \leq \breve{N}_s$ we compulsorily assign $\breve{N}_s$ to $N_{s,\boldsymbol{\delta}^{(p)},k}$.

After assigning the number of samples $N_{s,\boldsymbol{\delta}^{(p)},k}$, resampling can be done mode-specifically with any standard method. Resulting from the adaptive resampling, it obtains a group of particles with uneven weights $\Pr\left(\boldsymbol{\Delta}_k = \boldsymbol{\delta}^{(p)} | \boldsymbol{y}_{1:k}\right) / N_{s,\boldsymbol{\delta}^{(p)},k}$ depending on the modes $\boldsymbol{\delta}^{(p)}$ that the particles belong to.

One can then derive the lower boundary of the estimated effective sample size (Doucet et al. (2000)) as

$$\begin{aligned} & \hat{N}_{eff} \\ = & \frac{1}{\sum_{i=1}^{N_{s,k}} \left(w_k^{(i)}\right)^2} \\ = & \frac{1}{\sum_{p=1}^{2^{N_m}} \left(\Pr\left(\boldsymbol{\Delta}_k = \boldsymbol{\delta}^{(p)} | \boldsymbol{y}_{1:k}\right)^2 / N_{s,\boldsymbol{\delta}^{(p)},k}\right)}, \end{aligned} \tag{29}$$

where $N_{s,k} = \sum_{p=1}^{2^{N_m}} N_{s,\boldsymbol{\delta}^{(p)},k}$ is the number of particles at time k. Since $N_{s,\boldsymbol{\delta}^{(p)},k} \geq \Pr\left(\boldsymbol{\Delta}_k = \boldsymbol{\delta}^{(p)}|\boldsymbol{y}_{1:k}\right) \cdot \tilde{N}_s$,

$$\frac{\Pr\left(\boldsymbol{\Delta}_k = \boldsymbol{\delta}^{(p)}|\boldsymbol{y}_{1:k}\right)^2}{N_{s,\boldsymbol{\delta}^{(p)},k}} \leq \frac{\Pr\left(\boldsymbol{\Delta}_k = \boldsymbol{\delta}^{(p)}|\boldsymbol{y}_{1:k}\right)}{\tilde{N}_s}, \quad (30)$$

for each $\boldsymbol{\delta}^{(p)}$. Thereby, we get

$$\sum_{p=1}^{2^{N_m}} \frac{\Pr\left(\boldsymbol{\Delta}_k = \boldsymbol{\delta}^{(p)}|\boldsymbol{y}_{1:k}\right)^2}{N_{s,\boldsymbol{\delta}^{(p)},k}}$$
$$\leq \frac{1}{\tilde{N}_s} \sum_{p=1}^{2^{N_m}} \Pr\left(\boldsymbol{\Delta}_k = \boldsymbol{\delta}^{(p)}|\boldsymbol{y}_{1:k}\right) \leq \frac{1}{\tilde{N}_s}. \quad (31)$$

Substituting this into (29) yields

$$\hat{N}_{eff} \geq \tilde{N}_s. \quad (32)$$

Thus, the proposed adaptive resampling method will keep the effective sample size larger or equal to $\tilde{N}_s$.

The least efficient case happens when one of the modes has the posterior probability 1. For instance, if there are $N_m = 24$ modes in the system, $\tilde{N}_s = 1000$ and $\breve{N}_s = 100$. In worst case we still have $\breve{N}_s\left(N_m - 1\right) + \tilde{N}_s$ particles in the system. However $\breve{N}_s\left(N_m - 1\right)$ of these have null weights. This gives the effective sample size $\hat{N}_{eff} = \tilde{N}_s = 1000$, and we waste computation on the other $\breve{N}_s\left(N_m - 1\right) = 2300$ particles.

4.2 Fault Detection and Diagnosis with Particle Filter

The system is working in mode $\boldsymbol{\Delta}_k = \boldsymbol{\delta}^{(p)}$ when the marginal probability mass $\Pr\left(\boldsymbol{\Delta}_k = \boldsymbol{\delta}^{(p)}|\boldsymbol{y}_{1:k}\right)$ becomes significant. This probability mass is obtained by marginalizing the distribution $p\left(\boldsymbol{\xi}_k|\boldsymbol{y}_{1:k}\right)$ along $\boldsymbol{x}_k$, according to

$$\Pr\left(\boldsymbol{\Delta}_k = \boldsymbol{\delta}^{(p)}|\boldsymbol{y}_{1:k}\right) = \int p\left(\boldsymbol{\xi}_k|\boldsymbol{y}_{1:k}\right) d\boldsymbol{x}$$
$$\approx \sum_{j=1}^{N_{s,k}} w_k^{(j)} \rho_{\boldsymbol{\Delta}_k,\boldsymbol{\delta}^{(p)}}. \quad (33)$$

We provide two methods to detect the failure mode. One may define thresholds $h_{\boldsymbol{\delta}^{(p)}}$ for each system mode. Once the estimated marginal mass of a mode other than the fault-free mode exceeds the threshold, then $h_{\boldsymbol{\delta}^{(p)}}$ is captured and the system is considered to suffer the corresponding failure mode. That is, when $\Pr\left(\boldsymbol{\Delta}_k = \boldsymbol{\delta}^{(p)}|\boldsymbol{y}_{1:k}\right) \geq h_{\boldsymbol{\delta}^p}$ it reports failure mode $\boldsymbol{\delta}^{(p)}$ to occur between time instants $k-1$ and k.

An alternative way is to simply draw the conclusion by picking up the most significant mode $\boldsymbol{\delta}^{(m)}$, where $\Pr\left(\boldsymbol{\Delta}_l = \boldsymbol{\delta}^{(m)}|\boldsymbol{y}_{1:l}\right) \geq \Pr\left(\boldsymbol{\Delta}_l = \boldsymbol{\delta}^{(p)}|\boldsymbol{y}_{1:l}\right)$, $\forall \boldsymbol{\delta}^{(p)}$, $p \neq m$. Once observed that the most significant mode $\boldsymbol{\delta}^{(m)}$ is other than the fault-free mode, the fault is detected.

5 Case Study

5.1 Detection of Common Failure Modes in Position-like Measurement

In industrial control systems typical failure modes in measurements are

- *bias* - the measurement has a constant-like bias relative to the true signal;
- *drift* - the measurement drifts off relative to the true signal, either by a stochastic process (Wiener process) or deterministically (ramp); and
- *outliers* - a sample from a measurement signal that lies abnormally far away from the other values.

For safety and reliability, in industry applications, it is commonly required for sensor redundancy. This is to install multiple sensors to measure the same system output or state. The fault detection and diagnosis in this case should focus on monitoring and identifying the conditions of sensors.

In this simulation, it is assumed that the true state being measured is $[0, 0]$, but it is embedded in zero mean white Gaussian noise. The above listed three failure modes will then occur, but it is assumed that they do not occur simultaneously.

It needs to be clear that the above proposed change detection on a zero mean white noise sequence is not trivial. A two dimensional zero mean white noise sequence can be interpreted as the residual of comparing homogeneous measurements from two sensors of the same two dimensional state, for instance the position of a surface vessel. Thus, this detection and diagnosis problem can easily be extended to a wide range of applications.

5.2 System Modeling

To model the signal with the above mentioned failure modes into (24) and (25), we first construct the state vector $\boldsymbol{x}$, consisting of the trivial position and the possible bias and drifting terms. So, we assign $\boldsymbol{x}_k = \begin{bmatrix}\boldsymbol{p}_k^\top & \boldsymbol{b}_k^\top & \boldsymbol{d}_k^\top\end{bmatrix}^\top$, where $\boldsymbol{p}_k \equiv \begin{bmatrix}0 & 0\end{bmatrix}^\top$ is just as the zero mean assumption, $\boldsymbol{b}_k \in \mathbb{R}^2$ is the bias term, and $\boldsymbol{d}_k \in \mathbb{R}^2$ is the drifting term. According to the different failure modes, we may define the dynamics of the system and measurement as in Table 1. Here, $\boldsymbol{p}_{m,k}$

System mode	System	Measurement
Δ^0 Fault-free	$\begin{bmatrix} \boldsymbol{p}_{k+1} \\ \boldsymbol{b}_{k+1} \\ \boldsymbol{d}_{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}$	$\boldsymbol{p_{m,k}} = \boldsymbol{p}_k + \boldsymbol{v}_k$
Δ^1 Bias fault	$\begin{bmatrix} \boldsymbol{p}_{k+1} \\ \boldsymbol{b}_{k+1} \\ \boldsymbol{d}_{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{b}^0 \\ \boldsymbol{0} \end{bmatrix}$	$\boldsymbol{p_{m,k}} = \boldsymbol{p}_k + \boldsymbol{b}_k + \boldsymbol{v}_k$
Δ^2 Drifting fault	$\begin{bmatrix} \boldsymbol{p}_{k+1} \\ \boldsymbol{b}_{k+1} \\ \boldsymbol{d}_{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{d}_k + \boldsymbol{w}_k^d \end{bmatrix}$	$\boldsymbol{p_{m,k}} = \boldsymbol{p}_k + \boldsymbol{d}_k + \boldsymbol{v}_k$
Δ^3 Outliers fault	$\begin{bmatrix} \boldsymbol{p}_{k+1} \\ \boldsymbol{b}_{k+1} \\ \boldsymbol{d}_{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}$	$\boldsymbol{p_{m,k}} = \boldsymbol{p}_k + \boldsymbol{v}_k'$

Table 1: System equations and measurement equations in each system mode.

denotes the measurement, $\boldsymbol{v}_k$ is the measurement noise in fault-free condition, and $\boldsymbol{v}_k'$ is the measurement noise in outliers failure mode. The measurement noise terms will be discussed in detail later. Correspondingly, Δ^0 to Δ^3 are assigned to the four system modes, $\boldsymbol{b}^0$ is the bias, that subjects to the prior probability $p(\boldsymbol{b})$, and $\boldsymbol{w}_k^d$ is driving noise for the drift.

The measurement noise $\boldsymbol{v}_k$ in mode Δ^0, Δ^1, and Δ^2 are assumed to be white, subject to $N(\boldsymbol{0}, \boldsymbol{I}_{2\times 2})$. System mode Δ^1 represents the bias fault. The bias is modeled as a step sequence

$$\boldsymbol{b}_k = \begin{cases} \boldsymbol{0} & 0 < k \le l \\ \boldsymbol{b}^0 & k > l \end{cases}, \tag{34}$$

where l is the time when the bias fault occurs, and $\boldsymbol{b}^0$ is uniformly distributed in the region $\left(\boldsymbol{b}^{\text{in}} \backslash \boldsymbol{b}^{\text{out}}\right) \subset \mathbb{R}^2$ [3]. Here, $\boldsymbol{b}^{\text{in}}$ covers the region where the bias should be, and the exclusion $\boldsymbol{b}^{\text{out}}$ is used to distinguish the fault-free mode and the bias mode. System mode Δ^2 represents the drift failure mode. The drifting term $\boldsymbol{d}_k$ is modeled as a random walking, such that $\boldsymbol{d}_{k+1} = \boldsymbol{d}_k + \boldsymbol{w}_k^d$, where $\left\{\boldsymbol{w}_k^d\right\}$ is a step sequence

$$\boldsymbol{w}_k^d = \begin{cases} \boldsymbol{0} & 0 < k \le l \\ \boldsymbol{w}^{d,0} & k > l \end{cases}, \tag{35}$$

where l is the time when the drift fault occurs, and $\boldsymbol{w}^{d,0}$ is uniformly distributed in the region $\left(\boldsymbol{w}^{d,\text{in}} \backslash \boldsymbol{w}^{d,\text{out}}\right) \subset \mathbb{R}^2$, similar to the bias case. The outliers can conceptually be seen as measurements with unusually large measurement noise. Hence, we can use $\boldsymbol{v}_k'$ belonging to a distribution with much larger variance than $\boldsymbol{v}_k$ to model this phenomenon. To simplify the calculation in the particle filter, when updating the weights of particles, the weights are assigned as $w_{k+1}^{(i)} = 0.05$ if $\left\|\boldsymbol{p}_{m,k}\right\|_2 > \tau$, where τ is a threshold, that is

$$p\left(\boldsymbol{v}_k'\right) = \begin{cases} 0 & \left\|\boldsymbol{v}_k'\right\| \le \tau \\ 0.05 & \left\|\boldsymbol{v}_k'\right\| > \tau. \end{cases} \tag{36}$$

						m		
				Δ^1	0	1	0	0
				Δ^2	0	0	1	0
			Pr_{mn}	Δ^3	0	0	0	1
	Δ^1	Δ^2	Δ^3	-	-	-	-	-
n	0	0	0	-	0.7	0.1	0.05	0.99
	1	0	0	-	0.1	0.9	0	0
	0	1	0	-	0.1	0	0.95	0
	0	0	1	-	0.1	0	0	0.01

Table 2: The Markov chain for the transition of combined mode $\left[\Delta^1\ \Delta^2\right]^\top$.

This simplification loses the probability nature since $\int_{\mathbb{R}^2} p\left(\boldsymbol{v}_k'\right) d\boldsymbol{v}_k'$ is infinite. However, it reflects the characteristics of outliers where the measurement shows a significant jump, and this is easier to calculate.

Second, the Markov chain representing the system mode transitions should be designed. The system mode is a combination of all failure modes, such as $\boldsymbol{\Delta}_k = \left[\Delta^1 \quad \Delta^2 \quad \Delta^3\right]^\top$, where Δ^1, Δ^2, and Δ^3 belonging to set $\{0, 1\}$ represents if their corresponding failure mode occurs or not. The Markov chains in Table 2 is designed for the mode transitions. For instance, the transition probability from $\boldsymbol{\Delta}_k = \{0, 0, 0\}$, which is the fault-free mode, to $\boldsymbol{\Delta}_{k+1} = \{0, 1, 0\}$, which is the drifting mode, is $\mathrm{Pr}_{\boldsymbol{\Delta}_k=\{0,0,0\},\boldsymbol{\Delta}_{k+1}=\{0,1,0\}} = 0.01$.

This Markov chain gives a more detailed description of the system behavior, but in a connotative manner. The system is assumed to never run into a multiple fault case, which is when more than one fault occur at the same time. The transition probabilities to such modes are therefore zero and neglected in Table 2. Another restriction of the mode transition is that the system mode can transfer to a fault case only from the fault-free mode. This is inspired by the assumption of the system behavior, and it helps when diagnosing an ambiguous fault (e.g. a small size bias can be mistaken for a drifting bias). Designing the transition probabilities related to the bias and the drift modes are tricky. Compared to the outliers, which is isolated events, detecting the bias and drift failures is close to estimating a time sequence. Hence, the transition probabilities from these modes back to the fault-free mode is small

[3] $A \backslash B$ is the complement of B in A.

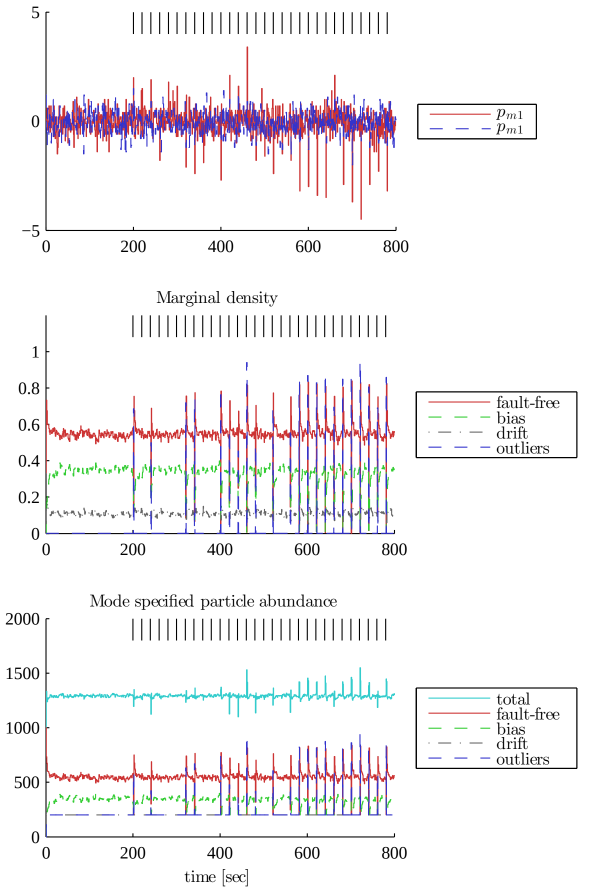

Figure 3: Outliers detection. The sizes of the outliers are $[1.5, -0.5]$ from 200 sec to 400 sec, $[2, 1]$ from 400 sec to 600 sec, and $[3, 1]$ from 600 sec to 800 sec, subject to Gaussian noise with covariance $\boldsymbol{I}_{2\times 2}$.

to make the particles stick to these modes and perform retentively.

5.3 Results

The simulation results in this section show the detection and diagnosis performance of the proposed algorithm to the different fault cases. In all the simulations, the PF adopts the same structure and parameters, and the number of particles is set to 1000. The mode transition Markov chain is given in Table 2. The covariance matrix of the measurement noise $\boldsymbol{v}_k$ takes the value $\boldsymbol{I}_{2\times 2}$. $\boldsymbol{b}^{\text{in}} = [-5, 5]^2$, $\boldsymbol{b}^{\text{out}} = \left\{\boldsymbol{x} \,\middle|\, \boldsymbol{x} \in \mathbb{R}^2, \|\boldsymbol{x}\|_2 < 2\sqrt{2}\right\}$, $\boldsymbol{w}^{d,\text{in}} = [-0.1, 0.1]^2$, $\boldsymbol{w}^{d,\text{out}} = \left\{\boldsymbol{x} \,\middle|\, \boldsymbol{x} \in \mathbb{R}^2, \|\boldsymbol{x}\|_2 < \sqrt{0.0001}\right\}$.

Detection of Outliers Figure 3 shows the detection results of outliers in the white Gaussian noise sequence.

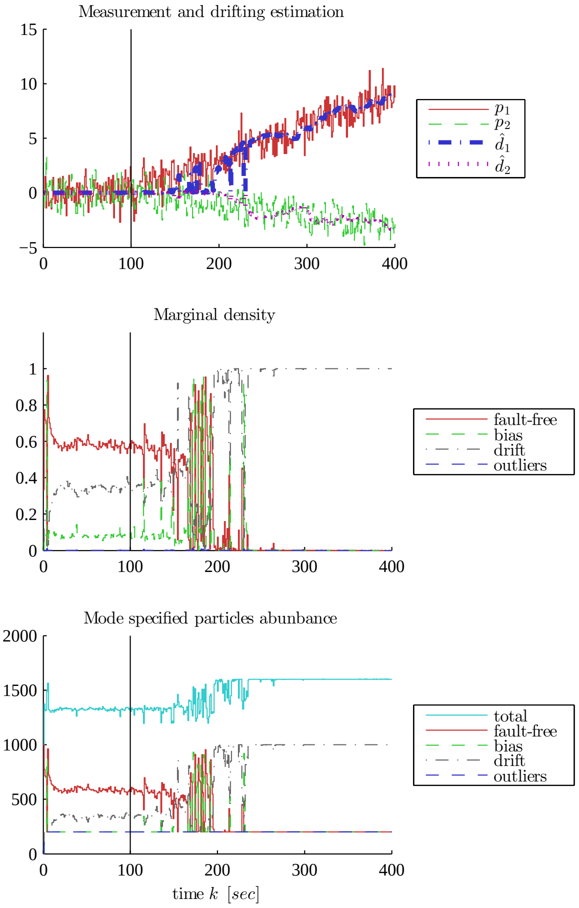

Figure 4: Drift detection. The upper graph shows the original signal and the estimation of the drift by the PF. The lower graph shows the abundance of the particles in each system mode.

In this simulation three different sizes of outliers are triggered; $[1.5\ 0.5]^\top$ from 200 sec to 400 sec, $[2\ 1]^\top$ from 400 sec to 600 sec, and $[3\ 1]^\top$ from 600 sec to 800 sec. The sizes of the outliers subject to a Gaussian noise with covariance $0.2 \times \boldsymbol{I}_{2\times 2}$.

It can be conclude from the result that the PF effectively detects the outliers with large enough size. If considering the outliers being the signal, the signal-to-noise ratio[4] for the different outliers are 1.25, 2.5, and 5, respectively. As the SNR increases, the PF's detection frequency of outliers also increases. SNR 2.5 is a critical value in the sense that the detection rate of the outliers is about 50%.

Drift Detection and Estimation Figures 4 and 5 show the detection and estimation of a drift fault. The drift speed is set to $[0.03\ -0.01]^\top$ meters per second.

[4] SNR for short, defined as A^2/σ^2, where A is the amplitude of the signal, and σ^2 is the variance of the noise.

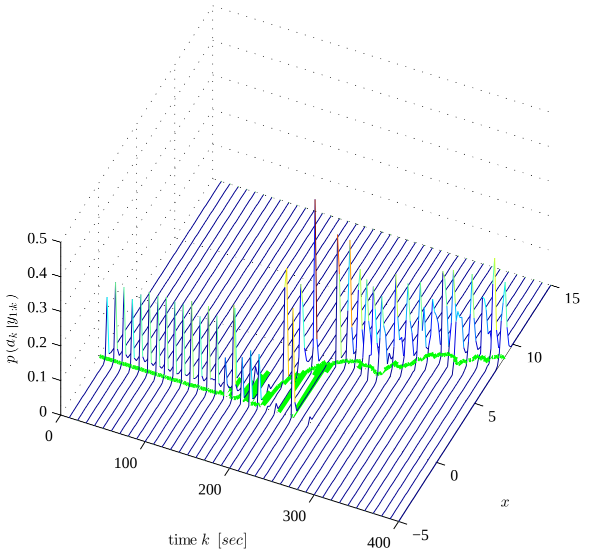

Figure 5: The estimated empirical distribution of the first component of $\boldsymbol{d}_k$, and the estimated mean as the curve in green.

This speed is relatively low compared to the variance of the noise.

As shown in Figure 4 that the drift is detected in the way that the marginal probability mass of the drifting mode exceeds the marginal probability mass of the fault-free mode at about 145 sec, which is about 45 seconds after the fault triggered. Because the drift speed is slow, the time-to-detection is relatively long. The detection happens when the SNR reaches 1.01, which is lower than the critical SNR 2.5 in the outliers case. Besides the consideration of sensitivity, this benefits from that the PF makes good use of the historical data to estimate the trend of the drift. Instead of only examining a measurement independently from the previous measurements, the detection of drift is like estimating a ramp sequence starting at an unknown time. However, not all historic information is used in the detection, since partial information leaks along the particle transfer from the drift mode to the fault-free mode. This information leak is an innate character of this algorithm, and it cannot be avoided by increasing the number of particles (unless we unrealistically let the number of particles increase with time).

In Figure 4 the bias mode has the largest marginal probability mass from 195 sec to 265 sec, which means that the PF diagnoses the current fault as a bias instead of a drift. This misdiagnosis is due to the behavior of the fault at this stage, which can be taken as either a bias or a drift. After 265 sec, the PF has cumulated enough information to distinguish these faults. Although there is a misdiagnosis during the process, the fault detection is always successful since the marginal probability mass of the fault-free mode is always low, and the estimate of the drift size is accurate.

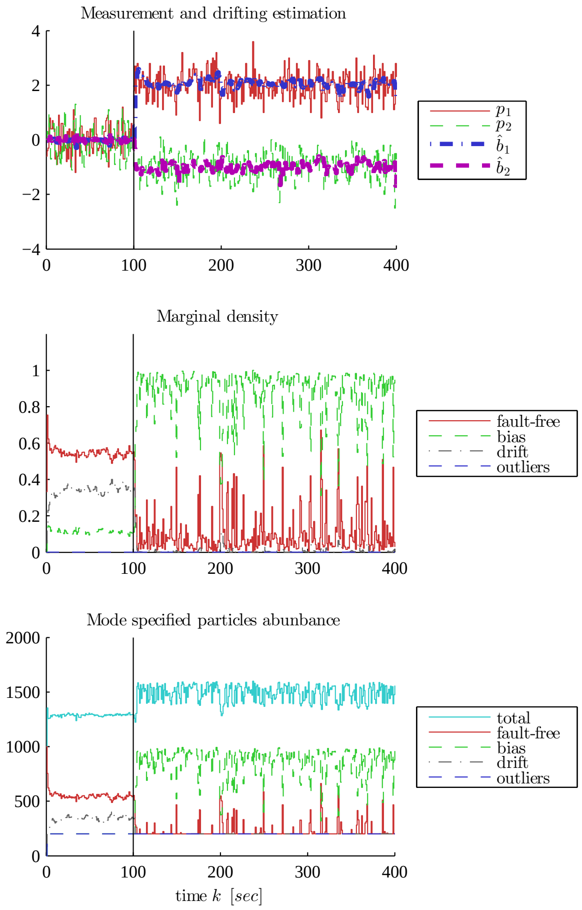

Figure 6: Bias detection. The upper graph shows the original signal and the estimation of the bias from the PF. The lower graph shows the abundance of the particles in each system mode.

Bias Detection and Estimation Figures 6 and 7 show the detection and estimation of a bias fault. The size of the bias is $[3, -1]^\top$, such that the SNR of the fault is 2.5. The bias is triggered at 100 second, and the detection is successful since the marginal probability mass of the bias mode is almost always the largest. The exception is a the few points where the measurement is close to the origin because of the noise.

6 Conclusion

In this paper, a PF based fault detection and diagnosis framework was proposed. In this framework, the sys-

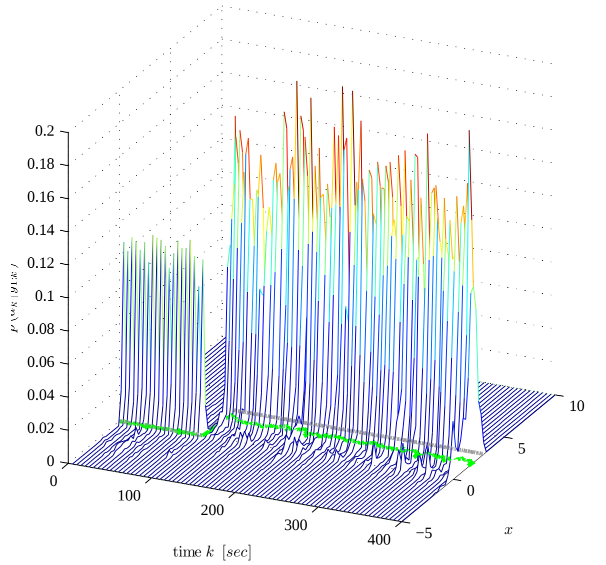

Figure 7: The estimated empirical distribution of the first component of $\boldsymbol{b}_k$, and the estimated mean as the curve in green.

tem with possible failures is modeled as a group of hidden Markov models, representing the system in fault-free and different failure modes, and a Markov chain, representing the transition between modes. This combined HMM and MC model can be used as a canonical model for a wide range of systems, since the HMM is compatible with the state-space model in control context, and the MC is also a generalized model of the system mode transition. By two introductory examples, we showed how to model a system into this form.

A modified PF algorithm was introduced to estimate the system mode and the states of the canonical model, and at the same time solve the FDD problem. This new PF algorithm extends the PFs into hybrid spaces of continuous and discrete components. A new resampling algorithm was developed along with the PF, to enhance the efficiency of the PF.

The proposed method is suitable for the system where the propagations of the distributions in fault-free and faulty conditions are known and can be properly modeled in switching mode hidden Markov model. In the case that the propagations of the distributions are not accurately known, rough approximations need to be used instead. These inaccurate models will affect the fault detection and diagnosis performance of the proposed method. To enhance the robustness of this method, a CUSUM algorithm Blanke et al. (2006) can be applied in addition. For example, we can calculate the time accumulation of the posterior probability of each mode, and compare with each other, or compare against predefined thresholds to detect the faults.

At last, a case study regarding the proposed FDD scheme for a faulty system was are given in detail. The case study demonstrates how to apply this PF- based FDD method, and it also verifies the method by simulations. The example is instructive and can be used as a template for developing new applications based on the same algorithm.

Acknowledgments

Research partly funded by Research Council of Norway (RCN) KMB project no. 199567: Arctic DP, with partners Kongsberg Maritime, Statoil, and DNV GL, and partly by RCN project no. 146025: CoE CeSOS.

We appreciate the constructive comments to the paper from PhD Jostein Bakkeheim (Kongsberg Maritime) and PhD Dong Trong Nguyen (Marine Cybernetics) during review of the text.

References

Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on*, 2002. 50(2):174 –188. doi:10.1109/78.978374.

Blanke, M., Kinnaert, M., Lunze, J., and Staroswiecki, M. *Diagnosis and Fault-Tolerant Control*. Springer Berlin Heidelberg, 2006.

Carpenter, J., Clifford, P., and Fearnhead, P. Improved particle filter for nonlinear problems. *Radar, Sonar and Navigation, IEE Proceedings* -, 1999. 146(1):2 –7. doi:10.1049/ip-rsn:19990255.

Chen, Z. Bayesian filtering: From kalman filters to particle filters, and beyond. Technical report, McMaster University, 2003.

Doucet, A. and Johansen, A. *The Oxford Handbook of Nonlinear Filtering*, chapter A tutorial on particle filtering and smoothing: Fifteen years later, pages 1–39. December. Cambridge University Press, Cambridge, 2009.

Doucet, A., Logothetis, A., and Krishnamurthy, V. Stochastic sampling algorithms for state estimation of jump markov linear systems. *Automatic Control, IEEE Transactions on*, 2000. 45(2):188 –202. doi:10.1109/9.839943.

Gordon, N., Salmond, D., and Smith, A. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 1993. 140(2):107 –113. URL ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=210672.

Gustafsson, F. *Adaptive Filtering and Change Detection*. John Wiley & Sons, Ltd, Chichester, UK, 1 edition, 2001.

Isermann, R. *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Verlag, 1 edition, 2006.

Kitagawa, G. Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics*, 1996. 5(1):pp. 1–25. doi:10.1080/10618600.1996.10474692.

Liu, J. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing*, 1996. 6:113–119. doi:10.1007/BF00162521.

Liu, J. S. and Chen, R. Sequential monte carlo methods for dynamic systems. *Journal of the American Statistical Association*, 1998. 93:1032–1044. doi:10.1080/01621459.1998.10473765.

Smith, A. F. M. and Gelfand, A. E. Bayesian statistics without tears: A Sampling-Resampling perspective. *The American Statistician*, 1992. 46(2):84–88. doi:10.1080/00031305.1992.10475856.

Zhao, B., Blanke, M., and Skjetne, R. Fault tolerant rov navigation system based on particle filter using hydroacoustic position and doppler velocity measurements. In *9th IFAC Conference on Manoeuvring and Control of Marine Craft*. 2012a. doi:10.3182/20120919-3-IT-2046.00048.

Zhao, B., Blanke, M., and Skjetne, R. Particle filter rov navigation using hydroacoustic position and speed log measurements. In *American Control Conference (ACC), 2012*. pages 6209 –6215, 2012b. doi:10.1109/ACC.2012.6315511.

Zhao, B., Skjetne, R., Blanke, M., and Dukan, F. Particle filter for fault diagnosis and robust navigation of underwater robot. *Control Systems Technology, IEEE Transactions on*, 2014. 22(6):2399–2407. doi:10.1109/TCST.2014.2300815.