



Research Note

## Robust fault detection and isolation scheme using fuzzy wavelet network with a hybrid design algorithm

M. Shahriari-kahkeshi\*

Faculty of Engineering, Shahrood University, Shahrood, P.O. Box 115, Iran.

Received 28 October 2015; received in revised form 12 March 2016; accepted 24 May 2016

### KEYWORDS

Robust fault detection and isolation;  
Fuzzy wavelet network;  
Adaptive threshold generation;  
Bounded-error approach;  
Artificial bee colony algorithm.

**Abstract.** This paper presents a new robust fault detection and isolation scheme using fuzzy wavelet network based on the bounded error approach. An efficient hybrid design algorithm, which consists of the orthogonal least square and the artificial bee colony algorithms, is proposed to design fuzzy wavelet network for modeling normal and faulty behaviors of the system. The proposed model provides an alternative description of the behavior of the system with high accuracy, but it suffers from model uncertainty because of model-reality mismatch in practical applications. To overcome this difficulty, the bounded error approach inspired from robust identification theory is applied to estimate the model uncertainty which defines a confidence interval of the model output and derives adaptive threshold for residual evaluation. Also, online fault isolation process is performed using fuzzy wavelet network models of the faulty system and analyzing the relation between a bank of residuals. Performance and efficiency of the proposed scheme is evaluated by simulating the nonlinear two-tank liquid level control system. Finally, some performance indexes are defined, and then the Monte-Carlo analysis is carried out to evaluate the reliability and robustness of the proposed scheme.

© 2017 Sharif University of Technology. All rights reserved.

### 1. Introduction

With growing the complexity of the industrial processes, probability of fault occurrence is increased; therefore, reliability and safety have become the most important system requirements which have motivated significant research in the Fault Detection and Isolation (FDI) field during the last two decades [1-5].

In general, FDI methods can be classified into model-based and data-based methods [6,7]. Among them, the model-based FDI methods have become most efficient tools and found widespread use because of their abilities to implement in real-time applications. Different models, such as analytical, knowledge-based,

and data-based models, can be used in the model-based FDI schemes [6,7].

Analytical model-based FDI schemes require precise mathematical model of the system. With growing the complexity of the system, modeling task becomes very hard and development of a precise mathematical model for system description will be difficult, so analytical models cannot be applied. To overcome such problems, some soft computing techniques are used to develop FDI schemes for industrial processes and engineering systems [2,8-17].

Due to their powerful function approximation and classification abilities, Neural Networks (NNs) are the most popular tools used to design FDI system [9,14-16]. In fact, the goal of designing NNs in FDI problems is to approximate a model and recognize a pattern as well. However, NN-based FDI systems suffer from local minimum traps, the difficulty of determining the size

\*. Tel.: +98 3832324401; Fax: +98 3832324438  
E-mail address: m.shahriari@eng.sku.ac.ir

of the hidden layer, learning rate, sufficient training data, and training method. Also, design of NN-based FDI scheme for complex problems requires very large network and this problem may lead to overfitting and poor generalization performance. Furthermore, NNs have block box structure, and it is not possible to utilize user experiences in modeling task.

When enough knowledge and experience are available, because of their interpretability and transparency, Fuzzy Systems (FSs) are used to develop FDI schemes [2,17]. However, in order to develop FSs, one must determine the number and location of the fuzzy basis functions and rules by trial and error. However, designing FDI system for complex problems with high input dimension is time-consuming which produces large and complex structure as a model. Also, FSs can only present local model of the system and cannot capture its local and global behaviors.

To overcome the mentioned disadvantages, one strategy is the use of wavelet functions and fuzzy systems in the network structure. This structure, known as Fuzzy Wavelet Network (FWN), not only has inference property of FSs and learning ability of NNs, but it also preserves MRA of wavelet transform [18]. The integration of the localization property of wavelets, learning ability of NNs, and inference property of FSs are the advantages of FWN in comparison to NNs and FSs for complex problem. Also, compared with NNs and FSs, the approximation accuracy, generalization and learning capability of FWN have been improved significantly by learning the translation parameters of the wavelets and adjusting the shape of fuzzy sets [19-21].

However, soft computing techniques can remedy the difficulties associated with the analytical models; however, the model uncertainty in industrial processes is inevitable. In practice, model uncertainty arises from wrong assumptions about the structure of the model or the type of disturbance which corrupts measurements and can dramatically decrease the reliability of fault detection. Hence, there is a need to develop robust FDI algorithms where model uncertainty is explicitly taken into account [22].

One strategy to develop robust FDI scheme, called 'passive', is based on enhancing the robustness of the fault detection scheme at the decision-making stage by propagating the uncertainty to the residual and generating adaptive thresholds [23]. This approach requires a mathematical description of the uncertainty that can be achieved by the parameter estimation methods like least square, maximum likelihood or Kalman filter. These methods assume that the error corrupting the data is a random variable with a known statistical property. However, in practice, the stochastic assumptions cannot be easily satisfied. So, an alternative method called 'the Bounded Error

Approach' (BEA) or 'set-membership' technique is introduced. The motivation for this approach is that in many practical cases, the unknown but bounded error description is more realistic and less demanding than the statistical description. This approach assumes that the measurements are affected by a bounded error [24]. Then, a set of parameters consistent with the considered model of the system, measurements, and error bound is computed as a closed set. When the measurement is not consistent with any member of this set, a fault is detected. This type of identification method has been applied to the development of several fault detection schemes [25-29].

The objective of this paper is to design a new robust fault detection and isolation scheme using fuzzy wavelet network based on bounded error approach. For this purpose, FWN is invoked to model the normal and faulty behaviors of the system. A new hybrid design algorithm is proposed to overcome the difficulties associated with the selection of the network structure and training process. The proposed hybrid design algorithm consists of the Orthogonal Least Square (OLS) and the Artificial Bee Colony (ABC) algorithms, which are used to optimize the structure and parameters of the network, respectively. In fact, the OLS algorithm purifies candidate wavelets; it automatically determines the network size, the number of fuzzy rules, sub-WNNs, and initial weights. Since the applied procedure of the OLS for efficient wavelets selection is not very sensitive to the input dimension, construction of the FWN for modeling purpose is free from problem dimension. Furthermore, to get rid of the troubles of usual training approaches, a self-tuning process which uses the ABC approach is used to adjust the parameters of dilation, translation, weights, and membership functions. By minimizing a quadratic measure of the error derived from the output of the system, the problem is formulated as an optimization problem and the ABC algorithm is used to solve it. The solution is directly obtained without any need for complicated computations. Then, to achieve robust fault detection scheme against disturbance, noise, and modeling errors, BEA approach is used to present a reliable estimation of the FWN model uncertainty. The estimated modeling uncertainty provides a confidence interval for the model output which is used to derive adaptive threshold for residual evaluation in the fault detection stage. Detection is based on checking whether the residual exceeds the dynamic thresholds which are determined by the method. Then, online fault isolation with no delay is performed using FWN models of the faulty system and analyzing the relation between a bank of residuals. The proposed scheme is applicable to real-time applications regardless of the input dimension. Also, due to using wavelets with different dilation values, it is able to represent both

local and global behaviors of the system, and it has high approximation accuracy and generalization capability.

This paper is organized as follows: In Section 2, a brief overview of the FWN is presented, then the proposed hybrid design algorithm is described. The description of the proposed passive robust FDI scheme is presented in Section 3. Some simulation and comparison results of the proposed robust FDI method are included in Section 4. Finally, the main concluding remarks are given in Section 5.

## 2. Fuzzy wavelet network overview and the proposed hybrid design algorithm

The fuzzy wavelet network is a powerful modeling tool that can represent an accurate model by combining the MRA of wavelet transform, learning ability of NNs, and inference property of fuzzy systems. A typical FWN can be described by a set of  $N$  fuzzy rules as in the following form [18]:

Rule I: If  $x_1$  is  $A_1^i$  and  $\dots$  and  $x_n$  is  $A_n^i$ ,

$$\text{then } \hat{y}_i = \sum_{k=1}^{T_i} w_{M_i, t_k} \psi_{M_i, t_k}^{(k)}(\underline{x}), \quad (1)$$

where  $x_j (j = 1, \dots, n)$  is the  $j$ th input variable, and  $\hat{y}_i (i = 1, \dots, N)$  is the output of the local model for the  $i$ th rule.  $w_{M_i, t_k}$  and  $T_i$  denote the weight of the output layer and the total numbers of wavelets for the  $i$ th rule, respectively.  $M_i$  and  $t_k$  are the dilation and translation parameters of wavelets, respectively, and  $\psi_{M_i, t_k}^{(k)}(\underline{x})$  represents the family of wavelets which is the dilated and translated versions of the single mother wavelet and expressed as:

$$\begin{aligned} \psi_{M_i, \underline{t}}^{(k)}(\underline{x}) &= 2^{M_i/2} \psi^{(k)}(2^{M_i} \underline{x} - \underline{t}^k) \\ &= \prod_{j=1}^n 2^{M_i/2} \psi^{(k)}(2^{M_i} x_j - t_j^k). \end{aligned} \quad (2)$$

In this paper, the Mexican hat wavelet function is chosen. So, dilated and translated versions of  $\psi_{M_i, t_j^k}(\underline{x})$  in Eq. (2) are obtained as:

$$\psi\left(\frac{\underline{x} - t_j^k}{M_i}\right) = \left(1 - \left(\frac{\underline{x} - t_j^k}{M_i}\right)^2\right) \exp\left(-\frac{1}{2} \left(\frac{\underline{x} - t_j^k}{M_i}\right)^2\right). \quad (3)$$

Finally, the output of the FWN is calculated as:

$$\hat{y}(\underline{x}) = \frac{\sum_{i=1}^N \mu_i(\underline{x}) \hat{y}_i}{\sum_{i=1}^N \mu_i(\underline{x})}, \quad (4)$$

where  $\underline{x} = [x_1, \dots, x_n]^T \in R^n$  is the input vector of

the network, and  $\mu_i(\underline{x}) = \prod_{j=1}^n A_j^i(x_j)$  is the membership degree of the  $i$ th rule.  $A_j^i$  is the fuzzy set that is characterized by the following Gaussian-type membership function:

$$A_j^i(x_j) = \exp\left[-\frac{1}{2} \left(\frac{x_j - c_j^i}{\sigma_j^i}\right)^2\right], \quad (5)$$

where  $c_j^i$  and  $\sigma_j^i$  represent the center and width of the Gaussian membership functions, respectively.

**Remark 1:** The Gaussian type fuzzy membership function and the Mexican hat wavelet function are smooth functions that are easily calculated by little computational load. Therefore, they are chosen to design FWN for modeling purpose.

An important issue in FWN modeling is to determine the efficient wavelet and number of fuzzy rules which have substantial influence on the quality of the FWN model. This problem will be discussed in Subsection 2.1. Another important issue is to update the adjustable parameters including dilation and translation parameters of wavelets, parameters of membership functions, and output weights. Subsection 2.2 is devoted to this issue.

### 2.1. Network construction

This subsection provides a brief description of designing FWN model based on the training data set,  $\{\underline{x}_i, y_i\}$ ,  $i = 1, 2, \dots, I$ ,  $\underline{x}_i \in R^n$ , and  $y_i \in R$ , so that the difference between the network output ( $\hat{y}_i$ ) and the desired output ( $y_i$ ) is minimized. At the first step, candidate wavelets are prepared by forming the wavelet lattice [30]. To reach this goal, wavelet functions which satisfy the inequality  $\psi_{M_i, \underline{t}}(\underline{x}\underline{x}_i)^T \psi_{M_i, \underline{t}}(\underline{x}\underline{x}_i) > \varepsilon_j$  for the training data set,  $\underline{x}\underline{x}_i \in R^I$  and  $j, i = 1, 2, \dots, n$ , are selected, and their dilation and translation parameters are saved. These wavelet functions are called candidate wavelets. So, set  $T_j$  contains all of the translation parameters of the wavelets with the dilation  $M_i$ , i.e.:

$$T_j = \left\{ \underline{t} \mid M_i : \psi_{M_i, \underline{t}}(\underline{x}\underline{x}_i)^T \psi_{M_i, \underline{t}}(\underline{x}\underline{x}_i) > \varepsilon_j \right\},$$

$$j = 1, 2, \dots, n,$$

where  $\varepsilon_j$  is a selected positive value which controls the total numbers of candidate wavelets. After determining the required translation set for each input variable, the required candidate wavelets are determined by searching through all of possible combinations of the terms as follows [18]:

$$W = \left\{ \psi_{M_i, t_1, t_2, \dots, t_n} : \prod_{j=1}^n \left( \psi_{M_i, t_j}(\underline{x}\underline{x}_i)^T \psi_{M_i, t_j}(\underline{x}\underline{x}_i) \right) \right\}$$

$$\geq \varepsilon, t_j \in T_j \} \tag{6}$$

Since some of the selected candidate wavelets with various translation parameters are often redundant for constructing FWN, they should be purified. For this purpose, OLS algorithm is utilized to select the best subset of  $W$  to approximate the desired output. This algorithm selects the wavelets that have the best fit to the desired output. Then, it repeatedly selects one of the remaining wavelets that can fit best to the data while combining with the previously selected wavelets. For computational efficiency, later selected wavelets are orthonormalized to earlier selected ones. Orthogonal least square algorithm uses error reduction ratio criteria to select the important wavelets which have significant contribution to the network output. After that, based on the dilation parameters of the efficient wavelets, number of fuzzy rules and sub-WNNs, dimension of the network, and initial weights are automatically determined.

**Remark 2:** Since the procedure of selecting the efficient wavelets by OLS algorithm is not very sensitive to the input dimension, the dimension of the modeling problem does not impose difficulty on network construction.

**2.2. Network learning**

After determining the FWN structure and initialization, its parameters must be tuned. In order to avoid trial and error, a self-tuning process which employs ABC algorithm is proposed for network learning. Artificial bee colony algorithm simulates the foraging behavior of a bee colony [31]. Due to its advantages such as simplicity in concept, easy implementation, and having fewer control parameters, this algorithm is utilized to solve different kinds of optimization problems such as training the artificial NNs [32] and training the FWN [33].

In the ABC algorithm, the colony of artificial honey bees consists of three types of bees: employed bees, onlookers, and scouts, in which half of the colony population are the employed bees, and the rest form the onlooker bees. Each solution to the optimization problem is called a food source in the search space. In other words, the searching process of bees for the food source stands for the finding process for the optimum solution to problem which has to be optimized. The fitness of solution corresponds to the profitability of food source. Also, the related profitability (fitness) of a food source (solution) is calculated by evaluation of the fitness function of the corresponding variables considering the related objective function. The number of solutions is known as SN, considered to be equal to the number of employed bees or the onlookers. The

employed bee, whose food source has been abandoned by the bees, becomes a scout. The main procedures of the ABC algorithm can be written as:

- **Step 1:** Initialize population;
- **Step 2:** Send the employed bees to their food sources and evaluate their nectar amounts (fitness);
- **Step 3:** Place each onlooker bee on a food source according to its nectar amounts (quality of its solution) based on the information provided by employed bees;
- **Step 4:** Determine the source to be abandoned and assign its employed bee as scout to search areas for discovering new food sources;
- **Step 5:** Memorize the best food source (solution) found so far;
- **Step 6:** Until the termination criterion is satisfied, repeat Steps 2 to 5.

For the modeling purpose, it is assumed that there are  $I$  input-output pairs,  $(\underline{x}(i), y(i))$ ,  $i = 1, \dots, I$ , where  $\underline{x} \in R^n$ . The modeling task is to design the FWN, such that the error between  $\hat{y}(\underline{x})$  and  $y(\underline{x})$  is minimized. The free parameters of the FWN network for training are  $c_j^i$ ,  $\sigma_j^i$ ,  $t_j^k$ , and  $w_{M_i, t_k}$ , where  $i = 1, \dots, c$ ,  $j = 1, \dots, n$ , and  $k = 1, \dots, T_i$ . Hence, the task is to expand the fuzzy wavelet basis function, such that the following objective function is minimized:

$$E_Q = \sum_{i=1}^I |\hat{y}(x(i)) - y(x(i))|^2, \tag{7}$$

where  $E_Q$  is the fitness of  $Q$ th food source. Considering Eqs. (1) to (4) and assuming that there exists  $I$  sample data over a time interval  $[0 t_s]$ , where  $t_s$  is the simulation time, the FWN output of  $Q$ th food source associated with sample  $s$  can be written as:

$$\hat{y}^{(Q)}(x_j(s)) = \sum_{i=1}^c \hat{\mu}_i(x_j(s)) \left( \sum_{k=1}^{T_i} w_{M_i, t_k}^Q \prod_{j=1}^n 2^{\frac{M_i}{2}} \psi_j^Q (2^{M_i} x_j(s) - t_j^k) \right), \tag{8}$$

where:

$$\hat{\mu}_i(x_j(s)) = \frac{\left[ \prod_{j=1}^n \exp \left[ -\frac{1}{2} \left( \frac{(x_j(s) - c_j^{i,Q})}{\sigma_j^{i,Q}} \right)^2 \right] \right]}{\sum_{i=1}^c \left[ \prod_{j=1}^n \exp \left[ -\frac{1}{2} \left( \frac{(x_j(s) - c_j^{i,Q})}{\sigma_j^{i,Q}} \right)^2 \right] \right]}. \tag{9}$$

In the ABC algorithm, each population is a solution to the problem which determines the parameters of FWN. Therefore,  $Q$ th food source is represented as:

$$F_Q = [c_j^{i,Q} \sigma_j^{i,Q} t_j^{k,Q} w_{M_i}^{i,Q}]^T. \tag{10}$$

Thus, all free design parameters that should be updated by ABC in the FWN are:

$$\begin{cases} \underline{c}_j^{i,(Q)} = [c_1^{1,(Q)} \dots c_1^{c,(Q)} \dots c_n^{1,(Q)} \dots c_n^{c,(Q)}] \\ \underline{\sigma}_j^{i,(Q)} = [\sigma_1^{1,(Q)} \dots \sigma_1^{c,(Q)} \dots \sigma_n^{1,(Q)} \dots \sigma_n^{c,(Q)}] \\ \underline{t}_j^{k,(Q)} = [t_1^{1,(Q)} \dots t_1^{T_1,(Q)} \dots t_n^{1,(Q)} \dots t_n^{T_c,(Q)}] \\ \underline{w}_{M_i}^{i,(Q)} = [w_{M_1}^{1,(Q)} \dots w_{M_c}^{c,(Q)}] \end{cases} \tag{11}$$

In the ABC algorithm, during each generation, the food sources are evaluated via measuring the fitness, which is calculated from the objective function defined in Eq. (7), and then the best food sources are chosen. In the current problem, the best food source is the one that has minimum fitness. After applying the ABC algorithm, the best food source of the final iteration is the solution. Summary of the proposed approach for constructing FWN model is illustrated in Figure 1.

**Remark 3:** Off-line learning phase is performed to achieve accurate models which can describe the normal and faulty behaviors of the system. Then, obtained models are applied to identify the system. This task reduces the on-line computational complexity and omits any trial and error process during real-time operation.

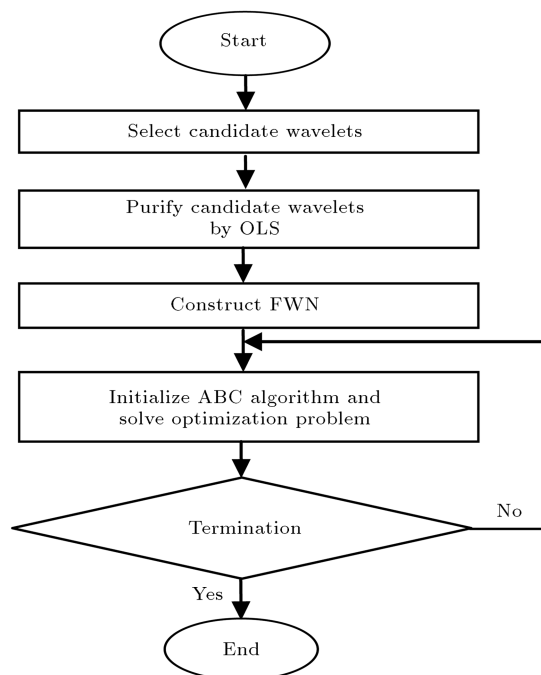


Figure 1. The hybrid algorithm for designing FWN model.

### 3. The proposed robust FDI scheme using FWN based on BEA

The proposed robust FDI scheme using FWN based on BEA consists of two steps:

- (i) Fault detection step that involves residual generation using model-based concepts and development of adaptive thresholds for residual evaluation;
- (ii) On-line fault isolation step that uses FWN models of the faulty system and analyzes the relation between a bank of residuals based on the sensitivity or insensitivity to different faults to localize the detected faults.

In order to detect the fault, FWN is designed to model the normal behavior of the system, and then residual signal is generated by comparing the output of the FWN model with the output of the system. It is well known that the reliability of the FDI scheme is strongly dependent on model uncertainty, i.e. the mismatch between a FWN and the considered system. So, accurate estimation of model uncertainty is an important problem.

In this work, BEA which is inspired from robust identification theory is applied to determine the feasible set of parameters that is consistent with the model uncertainty. It estimates the uncertainty associated with the FWN model and delivers the confidence interval for the model output which leads to development of adaptive thresholds for residual evaluation. Then, threshold exceeding test is applied, and any deviation in residual signal from the designed thresholds yields to the diagnostic signal that declares fault occurrences.

After fault detection, on-line fault isolation is done using the FWN models of the faulty system. For this purpose, FWNs are designed to represent the faulty behavior of the considered system for defined faults, and then, residual signals are generated which are sensitive to all faults except one, so an efficient tool is provided for localizing detected faults. Also, in the fault isolation step, to get rid of uncertainty consequences, BEA is again used to estimate uncertainty of all faulty models, and adaptive thresholds are generated to evaluate residuals for fault isolation. The following sub-sections explain the details of the proposed scheme.

#### 3.1. Robust fault detection scheme

The first step for fault detection is residual generation. The residual is defined as the difference between the output of the system and that of the FWN model, which is:

$$r(k) = y(k) - \hat{y}(k), \tag{12}$$

where  $y(k)$  and  $\hat{y}(k)$  denote the system and FWN model outputs, respectively. Ideally, the residual signal

should be close to zero in the healthy mode; otherwise, it should significantly deviate from zero. However, in practice, modeling errors, external disturbance, and noise affect the residual and deviate it from zero, and thus generate false alarms during fault-detection phase. In order to avoid false alarms, it is necessary to analyze how changes in the input and output of the system make the residual deviate from zero. This task is performed using adaptive thresholds on the residual. This method is based on the assumption that the uncertainty of the model can be presented in the form of confidence interval for the output of the model. It calculates the admissible set of parameters to determine the confidence interval. After threshold development, threshold-exceeding test is applied to residual signal for decision-making about fault occurrence, and it generates a diagnostic signal,  $s_i(k)$ , as:

$$s_i(k) = \begin{cases} 0 & : Th_{lower} \leq r_i(k) \leq Th_{upper} \\ 1 & : Th_{lower} > r_i(k) \text{ or } r_i(k) > Th_{upper} \end{cases} \quad (13)$$

where  $Th_{lower}$  and  $Th_{upper}$  denote the lower and upper thresholds, respectively. In this test, when the residual exceeds the range defined by the threshold, it is claimed that model uncertainty cannot explain the residual alone, and therefore, a fault must have occurred. In fact, development of adaptive threshold instead of constant one increases the fault-detection sensitivity.

In the following, BEA is applied to calculate the confidence interval of the FWN model output that can be directly applied to calculate the adaptive threshold for the residual signal in faultless operation.

Applying the BEA algorithm to compute the confidence interval of the model output, the output of the FWN model in Eq. (4) should be expressed in the form of a model with linear parameters as:

$$\hat{y}(k) = \underline{\zeta}^T(k) \hat{\theta}, \quad (14)$$

where:

$$\underline{\zeta}(k) = \begin{bmatrix} \hat{\mu}_1(k) \psi_{M_1, t_1^1} \\ \vdots \\ \hat{\mu}_1(k) \psi_{M_1, t_1^{T_1}} \\ \vdots \\ \hat{\mu}_i(k) \psi_{M_i, t_i^{T_i}} \\ \vdots \\ \hat{\mu}_C(k) \psi_{M_C, t_c^1} \\ \vdots \\ \hat{\mu}_C(k) \psi_{M_C, t_c^{T_c}} \end{bmatrix}, \quad \hat{\theta} = \begin{bmatrix} w_{M_1, t_1^1} \\ \vdots \\ w_{M_1, t_1^{T_1}} \\ \vdots \\ w_{M_i, t_i^{T_i}} \\ \vdots \\ w_{M_C, t_c^1} \\ \vdots \\ w_{M_C, t_c^{T_c}} \end{bmatrix}. \quad (15)$$

In Eq. (14),  $\underline{\zeta}(k)$  represents the fuzzy wavelet basis functions that are computed during off-line learning

phase, and  $\hat{\theta}$  is the time-invariant parameter vector which denotes the weight coefficients of the sub-WNNs.

Assume that the system output is described in the following form:

$$y(k) = \hat{y}(k) + e(k), \quad (16)$$

where  $y(k)$  is the  $k$ th measurement of the system output and  $e(k)$  consists of a structural deterministic error which results from the model-reality mismatch and a stochastic error due to the measurement noise. In practice, it is difficult to describe  $e(k)$  in a stochastic way; in many practical cases, the unknown but bounded error description is more realistic and less demanding than the statistical description, so BEA is applied. It considers a more realistic description of the error in the form of bounds on its instantaneous values as:

$$e^{\min}(k) \leq e(k) \leq e^{\max}(k), \quad (17)$$

where the bounds  $e^{\min}(k)$  and  $e^{\max}(k)$  can be estimated.

The idea underlying the BEA approach is to compute the feasible parameter set for each input-output pair of the system, such that the generated model output,  $\hat{y}(k)$ , be consistent with the system output,  $y(k)$ , and with the defined error bounds in Eq. (15). The admissible set of parameters for  $K$  data points is defined as:

$$\Theta = \left\{ \theta \in R^{\sum_{k=1}^K T_k} \left| \begin{aligned} &y(k) - e^{\max}(k) \leq \hat{y}(k) \\ &\leq y(k) - e^{\min}(k), \quad k = 1, \dots, K \end{aligned} \right. \right\}. \quad (18)$$

Each parameter vector in  $\Theta$  is a valid estimation of the parameter vector  $\theta$  which determines the weights of the FWN model. Considering Relations (17) and (18), the confidence interval for the output of the system is described as:

$$\underline{\zeta}^T(k) \theta^{\min}(k) + e^{\min}(k) \leq y(k) \leq \underline{\zeta}^T(k) \theta^{\max}(k) + e^{\max}(k), \quad (19)$$

where:

$$\theta_l^{\min}(k) = \arg \min_{\theta \in \Theta} \underline{\zeta}^T(k) \theta_l, \quad l = 1, \dots, T_c,$$

$$\theta_l^{\max}(k) = \arg \max_{\theta \in \Theta} \underline{\zeta}^T(k) \theta_l, \quad l = 1, \dots, T_c. \quad (20)$$

The minimum and maximum values of the parameters in Eq. (20) are determined using the linear programming technique [24]. In practice, center of  $\Theta$  is selected

as the estimation of the parameter vector,  $\underline{\theta}$ , and is obtained as:

$$\hat{\theta}_l = \frac{\theta_l^{\min} + \theta_l^{\max}}{2}, \quad l = 1, \dots, T_c. \quad (21)$$

Considering confidence interval of the system output in Eq. (19) and residual signal in Eq. (12), the following adaptive threshold is obtained:

$$\begin{aligned} \underline{\zeta}^T(k)\underline{\theta}^{\min}(k) + \varepsilon^{\min}(k) - \hat{y}(k) &\leq r(k) \\ &\leq \underline{\zeta}^T(k)\underline{\theta}^{\max}(k) + \varepsilon^{\max}(k) - \hat{y}(k). \end{aligned} \quad (22)$$

The lower and upper bounds of the residual signal in Eq. (22) determine the adaptive thresholds on the residual, making decisions about possible fault occurrences and providing robust performance against modeling uncertainty.

**Remark 4:** It is worth mentioning that the number of fuzzy rules and basis functions considerably decreases due to using wavelet lattice and OLS algorithm in the modeling step. So, the proposed modeling task can remedy the well-known computational problem of the BEA approach. Therefore, the proposed scheme provides the implementation of BEA for any problems regardless of their dimensions.

At this step, all faults are detected, but it is not possible to find out what fault has exactly occurred. In order to localize the occurred fault, it is necessary to perform fault isolation.

### 3.2. Fault isolation scheme

Once the fault is detected, it must be isolated in order to pinpoint its location among others within the considered system. The proposed fault isolation scheme assumes that there are  $N$  possible known faults that may occur in the process, and all of them are independent of each other which do not occur simultaneously.

The design procedure of the proposed fault isolation strategy is based on the Generalized Observer Scheme (GOS) with multiple FWN. It defines the relationship between the residuals based on the sensitivity and insensitivity to different faults that may occur in the system. It designs  $N + 1$  fuzzy wavelet network models,  $\{FWN_i\}_{i=0}^N$ , where  $FWN_0$  corresponds to the normal mode and  $FWN_i$  for  $i = 1, 2, \dots, N$  represents the  $i$ th faulty mode where the faults defined by the

FWN models are independent of each other. The proposed scheme consists of a bank of residuals running in parallel mode where each of them is sensitive to all faults except one. The decision task for fault isolation is done as follows: if the  $i$ th residual is zero or limited to the adaptive threshold bounds and all of the remaining residuals are nonzero or exceed their certain threshold, then a decision on the occurrence of the  $i$ th fault is made.

To achieve reliable results, the isolation task is done using adaptive threshold strategy in the decision-making step. For this, BEA is again used to estimate uncertainty of each faulty model, and then similar to the previous sub-section, adaptive thresholds are generated to evaluate a bank of residuals and to achieve robust isolation performance against modeling uncertainty. So, the isolation task is done as shown in Box I, where  $Th_l$  and  $Th_u$  denote the lower and upper bounds of the developed thresholds in the fault isolation step.

**Remark 5:** The proposed scheme not only has robust performance, but it can also detect and isolate faults, simultaneously. In other words, faults are isolated upon their detection, and there is no delay between detection and isolation of the fault.

## 4. Simulation results

Consider a two-tank liquid level control system, shown in Figure 2. The system consists of two tanks, one flow

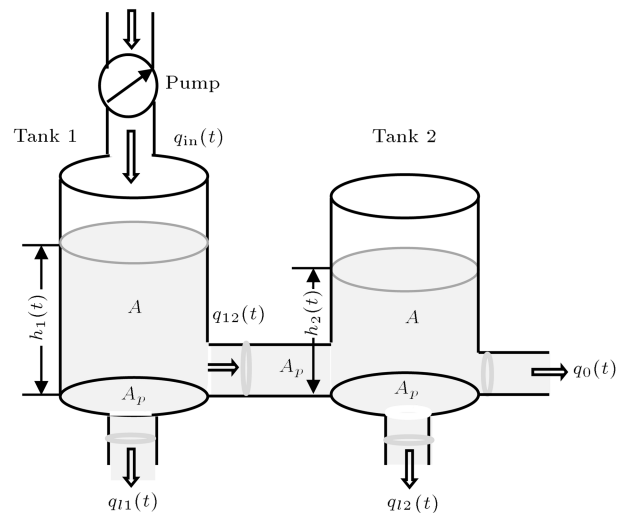


Figure 2. Two-tank system.

$$\left. \begin{aligned} Th_{li}(k) \leq r_i(k) \leq Th_{ui}(k) \\ r_j(k) < Th_{lj}(k) \quad \text{or} \quad r_j(k) > Th_{uj}(k) \quad \forall j \in \{1, \dots, i-1, i+1, \dots, N\} \end{aligned} \right\} \Rightarrow f_i(k) \neq 0. \quad (23)$$

Box I

source, one outlet pipe, and one connecting pipe [34]. Input and output of the system are considered to be the applied voltage to the pump,  $v_i(t)$ , and the liquid level in the second tank,  $h_2(t)$ , respectively. The dynamic model of the considered system is described by the following differential equations:

$$\begin{aligned} A\dot{h}_1(t) &= q_{in}(t) - q_{12}(t) - q_{l1}(t), \\ A\dot{h}_2(t) &= q_{12}(t) - q_o(t) - q_{l2}(t), \end{aligned} \quad (24)$$

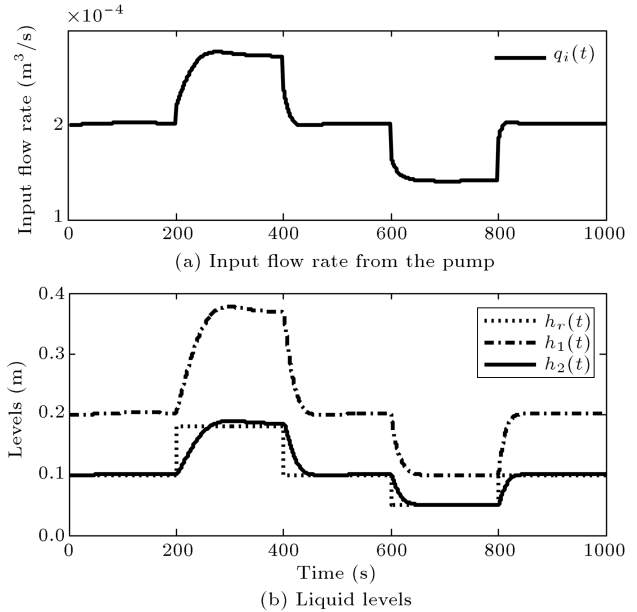
where  $A$  is the cross-sectional area of tanks, and  $h_1(t)$  and  $h_2(t)$  are the liquid levels of tanks 1 and 2, respectively.  $q_{in}(t)$ ,  $q_{12}(t)$ , and  $q_o(t)$  represent the liquid flow rates in  $\text{m}^3/\text{s}$  from the pump, from the tank 1 to tank 2 and the outflow from tank 2, respectively;  $q_{l1}(t)$  and  $q_{l2}(t)$  are the leakage flow rates of tanks 1 and 2, respectively. The liquid flow rates are defined as:

$$\begin{aligned} q_{12}(t) &= c_1 A_p \sqrt{2g(h_1(t) - h_2(t))}, \\ q_o(t) &= c_2 A_p \sqrt{2gh_2(t)}, \\ q_{l1}(t) &= l_1 A_p \sqrt{2gh_1(t)}, \\ q_{l2}(t) &= l_2 A_p \sqrt{2gh_2(t)}, \end{aligned} \quad (25)$$

where  $A_p$  is the cross-sectional area of the pipes,  $g$  is the gravitational constant,  $c_1$  is the flow coefficient of the interconnecting pipe that can measure the amount of clogging in pipe 1,  $c_2$  is the flow coefficient of the outflow pipe that can measure the amount of clogging in pipe 2. Also,  $l_1$  and  $l_2$  denote the leakage flow coefficients of tanks 1 and 2, respectively. The control objective is to regulate the liquid level,  $h_2(t)$ , at the reference value,  $h_r(t)$ . For this purpose, a discrete-time PI controller is applied to regulate the output of the system. It is assumed that the liquid flow rate,  $q_{in}(t)$ , from the pump is a linear function of the applied voltage,  $v_i(t)$ . The following parameters are used in the simulation to model the normal behavior of the system [34]:

$$\begin{aligned} A &= 4.3 \times 10^{-3} \text{ m}^2, & c_1 &= 0.5, & c_2 &= 0.5, \\ A_p &= 2.87 \times 10^{-4} \text{ m}^2, & l_1 &= 0, & l_2 &= 0. \end{aligned}$$

Also, Gaussian noise with zero mean and variance of  $0.001 \text{ m}^2$  is added to both  $h_1(t)$  and  $h_2(t)$ . An example of the system behavior in the normal mode is shown in Figure 3. System response shows that the liquid level in tank 2 converges to its reference value prior to the fault occurrence. The performance of the PI controller has not changed according to the FDI information.



**Figure 3.** (a) Input flow rate from the pump. (b) Reference value and actual liquid levels in tanks 1 and 2.

#### 4.1. Modeling normal behavior of the two-tank system using FWN

In this part, FWN is trained off-line using the set of normal operating data. Inputs of the network are  $q_{in}(k-1)$ ,  $h_1(k-1)$ , and  $h_2(k-1)$ ; its output is  $\hat{h}_2(k)$ . The OLS algorithm is applied to determine the network structure and efficient wavelets. Four fuzzy rules or sub-WNNs are selected to construct network for modeling normal behavior of the system. Each fuzzy rule has the following form:

Rule  $i$ : If  $x_1$  is  $A_1^i$  and  $x_2$  is  $A_2^i$  and  $x_3$  is  $A_3^i$ ,

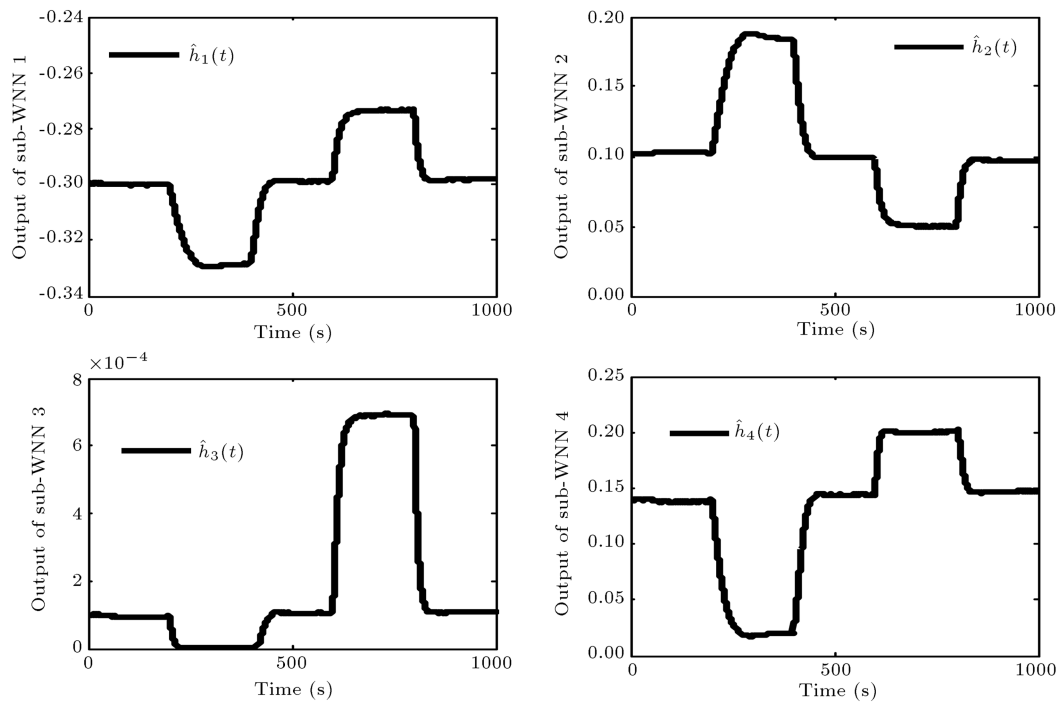
$$\text{then } \hat{h}_i = \sum_{k=1}^{T_i} w_{M_i,tk} \psi_{M_i,tk}^{(k)}(\underline{x}).$$

Number of fuzzy rules, sub-WNNs, and their corresponding dilation value assigned by the OLS algorithm are given in Table 1. After network initialization, the ABC algorithm is used to train the network as described in Section 2. Output of each sub-WNN, output of the FWN, and modeling error are shown in Figures 4 and 5, respectively. As can be seen from Table 1, the model structure is not complex. In fact, the proposed scheme is not only able to determine the network structure automatically, but it also can present highly accurate model using less number of rules and parameters. This feature provides possibility to achieve robust performance via applying BEA. Since only 8 parameters (sub-WNN weights) should be adjusted online by BEA, on-line computational load is low. Therefore, the proposed scheme can be implemented for the real-time application.

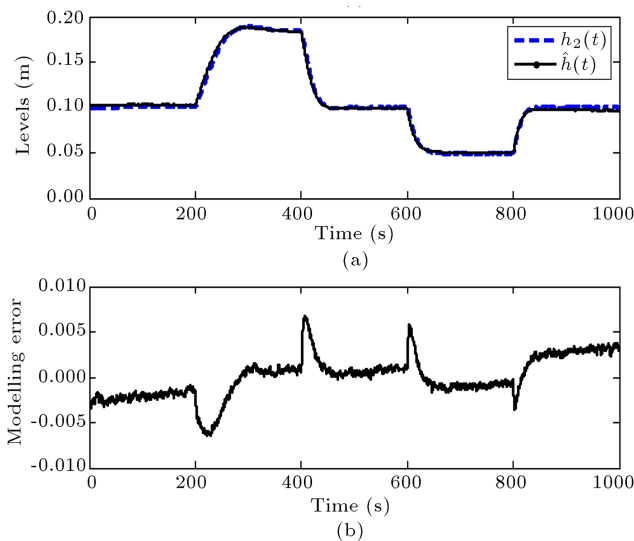


**Table 1.** Structure of the FWN for modeling two-tank system.

|  |       |       |       |       |
|--|-------|-------|-------|-------|
| Number of sub-WNN                                    | 4     |       |       |       |
| Total number of efficient wavelets by OLS            | 8     |       |       |       |
| Number of fuzzy rules                                | 4     |       |       |       |
| Rules, $R^i$   | $R^1$ | $R^2$ | $R^3$ | $R^4$ |
| Number of selected wavelets for each rule, $T_{i_g}$ | 3     | 3     | 1     | 1     |
| Dilation parameter of each rule, $M_{i_g}$           | -1    | 0     | 3     | 2     |
| Number of neuron for each rule                       | 3     | 3     | 1     | 1     |



**Figure 4.** Output of each sub-WNN.



**Figure 5.** (a) Actual output and output of the FWN model. (b) Modeling error of the FWN model (MSE =  $5.2878e-6$ ).

**4.2. Faulty scenarios of the two-tank system**

Faults can be classified according to the system components, modeling or time characteristics. In this paper, faults are classified according to the system components.

- **Actuator fault in pump:** The actuator fault is assumed by describing the actual input flow as  $q_{in}^f(t) = (1 - c_a)q_{in}(t)$ , where  $q_{in}(t)$  is the flow in no fault case and  $c_a \in [0, 1]$  is the fault constant; if  $c_a = 0$ , then there is no actuator fault; otherwise, it is faulty;
- **Component fault:** In the two-tank system, leakage flow in tanks 1 and 2, clogging in the interconnecting pipe between tanks 1 and 2, and clogging in the outflow pipe from tank 2 can be considered as component fault. These faults are simulated as a loss percentage of each component’s output from its normal operating output.

- Sensor fault in tanks 1 and 2:** The sensor fault is assumed by letting the actual level of tanks 1 and 2 be measured by  $h_1^f(t) = (1 - c_{s1})h_1(t)$  and  $h_2^f(t) = (1 - c_{s2})h_2(t)$ , where  $c_{s1} \in [0, 1]$  and  $c_{s2} \in [0, 1]$  are sensor fault constants, and  $h_1(t)$  and  $h_2(t)$  represent the actual level of tanks 1 and 2 in the fault-free conditions, respectively. If  $c_{s1} = 0$  and  $c_{s2} = 0$ , then there is no sensor fault; otherwise, it is faulty.

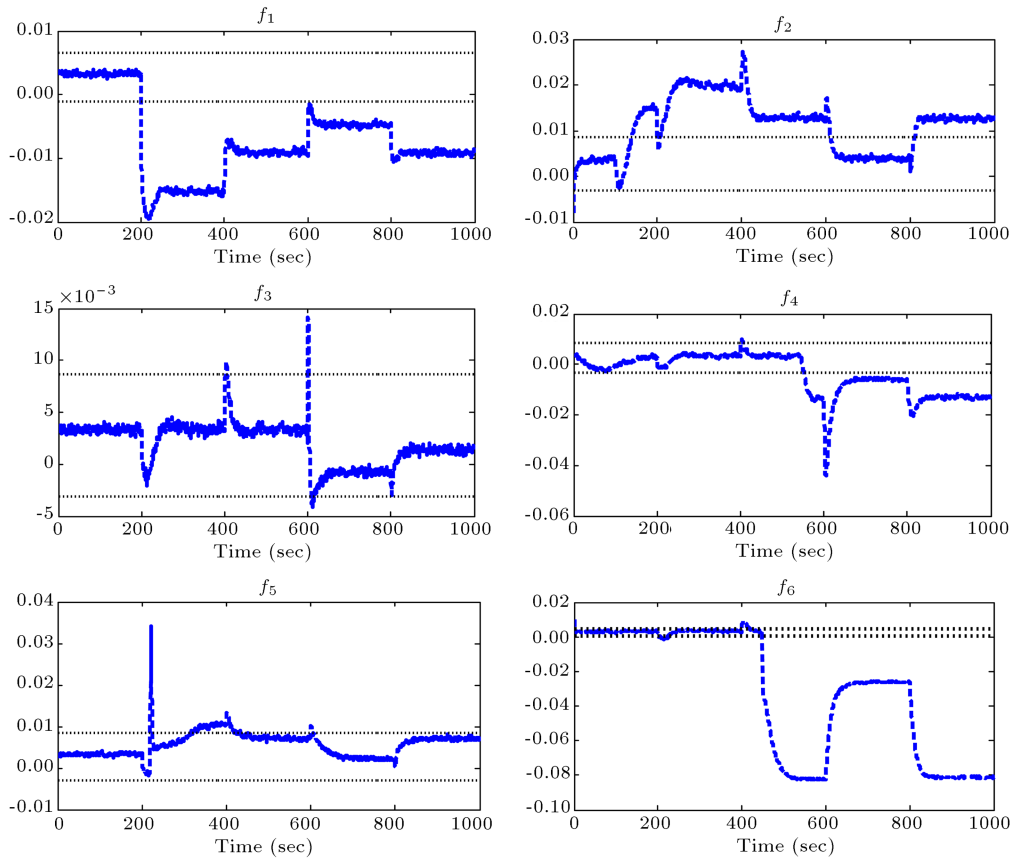
The proposed scheme is applied to detection and isolation of six different faults which are the component faults, the actuator faults, and the sensor faults. Table 2 presents the considered faulty scenarios, where the time of starting and ending of the fault is represented by  $t_{from}$  and  $t_{hor}$ , respectively.

### 4.3. Residual generation and decision making for fault detection

Residual generation is performed using the constructed FWN which models the normal behavior of the system. The first step for fault detection is the residual generation. The residual is generated for each faulty scenario. In order to perform fault detection, simple (constant) adaptive threshold strategies are used. Figure 6 shows the results of fault detection using the simple threshold strategy which is derived by  $\zeta$ -standard deviation test. Residual signal and simple thresholds are depicted for each defined faulty condition in Table 2. It is clearly seen from Figure 6 that a simple threshold strategy is not effective to cope with model uncertainty and noise;

**Table 2.** Simulated faults.

| Faulty scenarios                               | Name  | $t_{from}$ | $t_{hor}$ |
|--|-------|------------|-----------|
| 10% clogging in pipe 1 ( $c_1 = 0.45$ )        | $f_1$ | 200 s      | 1000 s    |
| 10% clogging in pipe 2 ( $c_2 = 0.45$ )        | $f_2$ | 100 s      | 1000 s    |
| 10% actuator fault ( $c_a = 0.1$ )             | $f_3$ | 600 s      | 800 s     |
| 10% leakage in tank 2 ( $l_2 = 0.1$ )          | $f_4$ | 500 s      | 1000 s    |
| 10% sensor fault in tanks 1 ( $c_{s1} = 0.1$ ) | $f_5$ | 190 s      | 1000 s    |
| 10% sensor fault in tanks 2 ( $c_{s2} = 0.1$ ) | $f_6$ | 450 s      | 1000 s    |



**Figure 6.** Generated residual and constant thresholds in the case of faults  $f_1$ - $f_6$  (dashed line: residual signal, dotted line: constant thresholds).

so, true detection rate is almost low. Furthermore, using larger thresholds for increasing true detection rate may cause considerable detection delay or missing the detection of some small faults.

Results of fault detection using the proposed adaptive thresholding scheme are given in Figures 7-9. In these figures, the residual signals and the

proposed adaptive thresholds based on the BEA are depicted. Also, decision level according to Subsection 3.1 is derived and shown. Comparison between the obtained results from two fault detection schemes demonstrates that the proposed scheme presents more reliable behavior than the simple threshold approach, and it not only achieves promising result in the

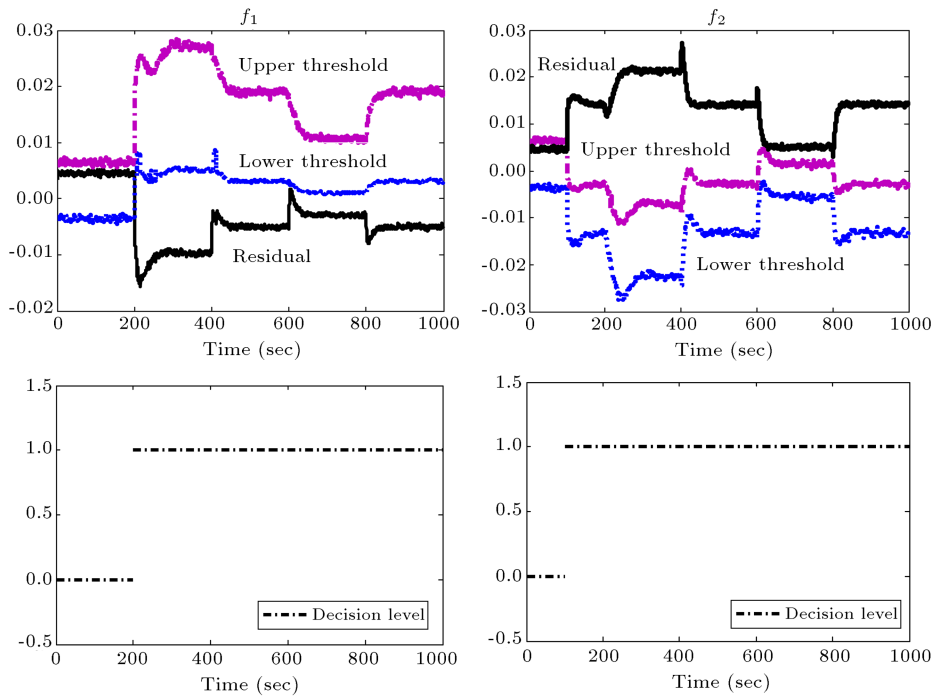


Figure 7. Fault detection using the proposed RFDI scheme: residual and adaptive thresholds, decision making for  $f_1$  and  $f_2$ .

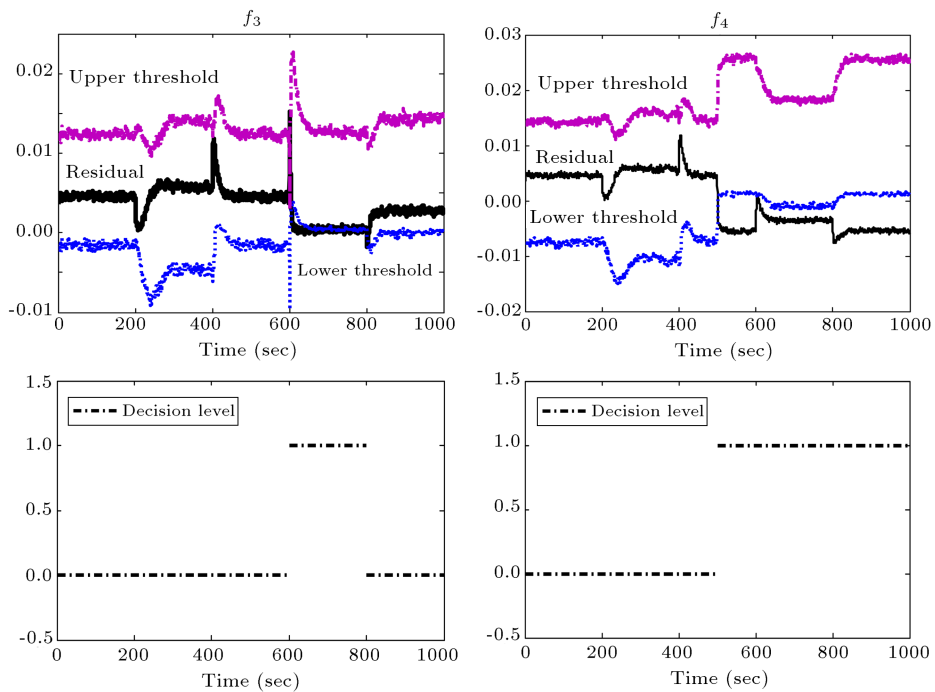
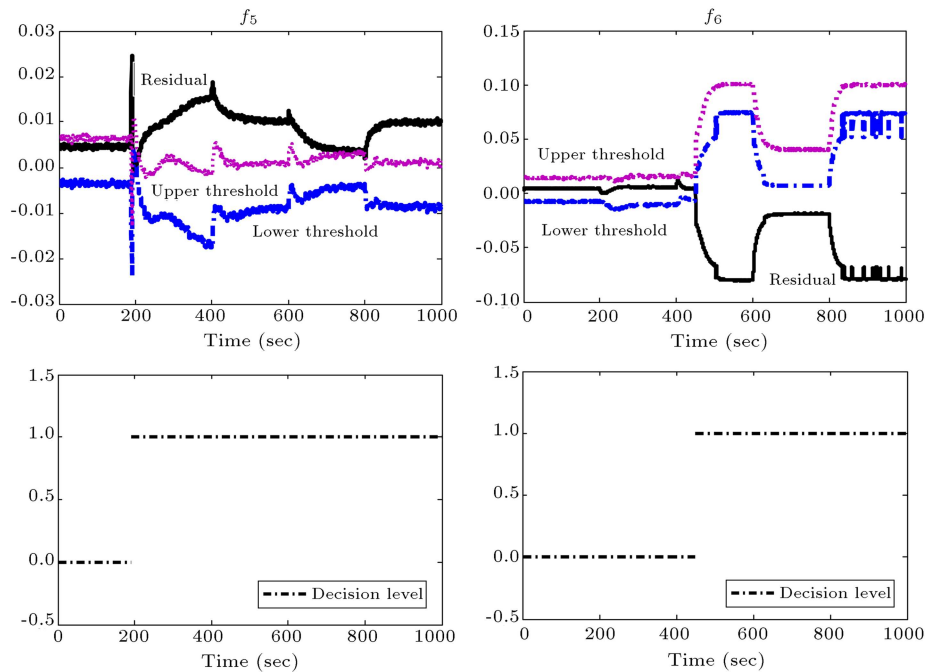


Figure 8. Fault detection using the proposed RFDI scheme: residual and adaptive thresholds, decision making for  $f_3$  and  $f_4$ .



**Figure 9.** Fault detection using the proposed RFDI scheme: residual and adaptive thresholds, decision making for  $f_5$  and  $f_6$ .

**Table 3.** Comparison of the modeling techniques.

| Method                    | Fuzzy rules | Neurons | Parameters | MSE       |
|---------------------------|-------------|---------|------------|-----------|
| Proposed FDI scheme       | 4           | 8       | 8          | 5.2878e-6 |
| NN-based FD scheme [25]   | —           | 35      | 140        | 1.8734e-2 |
| NFN-based FDI scheme [34] | 27          | —       | 108        | 4.5352e-3 |

detection time of the faults, but it also has highly true detection rate. This ability is due to the robustness of the proposed scheme against modeling uncertainty. Also, in order to show the performance of FWN to other modeling approaches, some comparison results are presented in Table 3. For this purpose, NNs and Neuro Fuzzy Networks (NFNs) as the popular techniques in the soft computing-based FDI schemes are adopted from literature and applied to modeling normal system [25,34]. As shown in Table 3, the number of fuzzy rules and adjustable parameters by the proposed scheme is reduced considerably, while the model accuracy is improved.

#### 4.4. Fault isolation

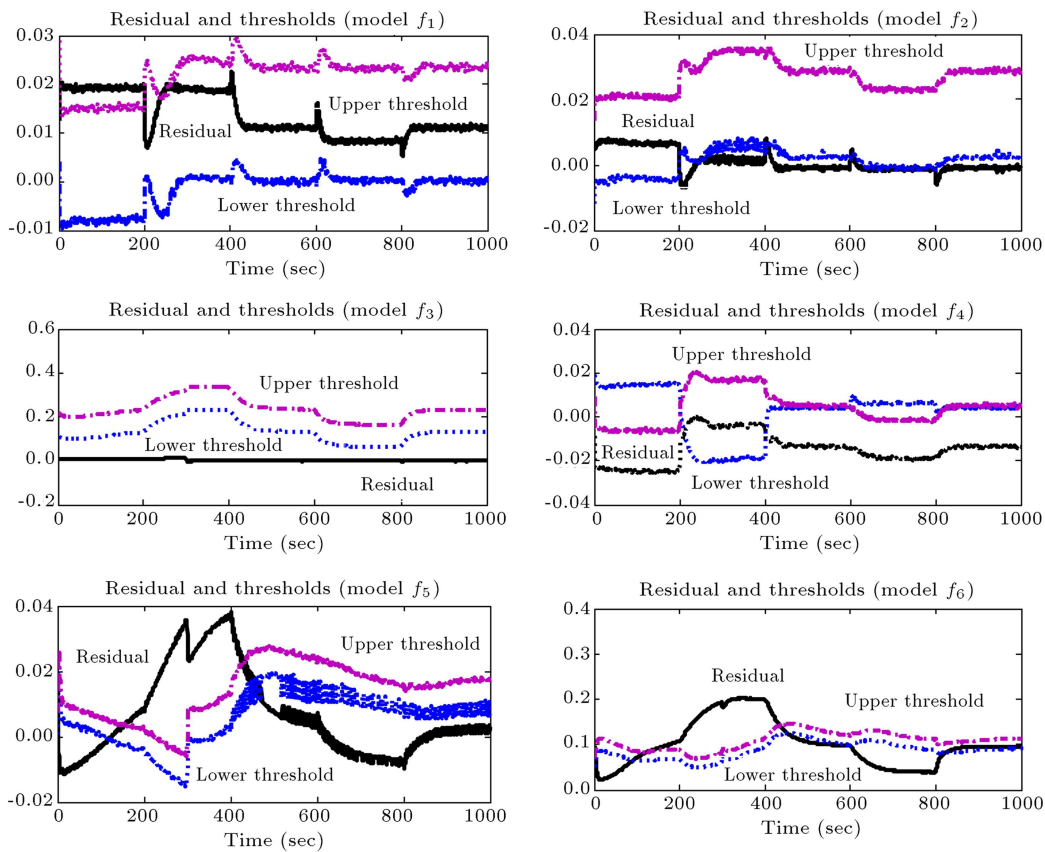
In order to implement fault isolation for the defined faulty scenarios, six FWNs are constructed to model the faulty behavior of the system. The main advantage of the proposed isolation scheme is its ability to isolate fault at the time of occurrence. In other words, there is no delay between fault detection and isolation. Also, as mentioned in Section 3.2, the residual of the defined fault must be zero. However, because of modeling

uncertainty, it may have a very small value. So, BEA is invoked to determine model uncertainty and develop adaptive thresholds for residual evaluation. Typically, simulation results for isolation of fault  $f_1$  is given. In order to isolate this fault, the residual generated by the fault model,  $f_1$ , should be near zero, while other faulty models should generate residuals different from zero. Figure 10 shows the residuals for all faulty behaviors. One can observe that this fault is isolated, because only the residual of the model  $f_1$  is near zero during the occurrence of  $f_1$ .

#### 4.5. Performance evaluation

In this section, the reliability and the robustness of the proposed FDI scheme have been evaluated by considering the following indexes [35]:

- **Average detection/isolation delay time** ( $\tau_{ad}$ ,  $\tau_{ai}$ ) is the difference between the moment the detection/isolation is done and the moment the fault occurs;
- **Missed fault rate** ( $r_{mf}$ ) is the total number of undetected faults versus the total number of Monte-Carlo runs;



**Figure 10.** Residuals and adaptive thresholds for  $f_1$ - $f_6$  in the case of the fault  $f_1$  (solid-line: residual, dash-dot line: upper threshold, and dotted line: lower threshold).

- **False alarm rate ( $r_{fa}$ )** is the total number of wrongly detected faults versus the total number of Monte-Carlo runs.

To compute the defined indexes, 1000 Monte-Carlo simulation tests have been carried out, and all indexes have been computed off-line for each faulty case. For each simulation run, it is assumed that the input-output data were affected by the measurement error, and also the value of fault and the time of its occurrence are randomly chosen. Table 4 shows the obtained results for each fault.

The reported results verify the reliability and robustness of the proposed scheme against measurement error, disturbance, and modeling uncertainty. It has low false alarm rate and missed fault rate because of

**Table 4.** Monte-Carlo analysis results.

| Fault | $\tau_{md}$ | $\tau_{mi}$ | $r_{mf}$ | $r_{fa}$ |
|-------|-------------|-------------|----------|----------|
| $f_1$ | 0.1 s       | 0.11 s      | 0.023    | 0.029    |
| $f_2$ | 0.15 s      | 0.15 s      | 0.031    | 0.035    |
| $f_3$ | 0.41 s      | 0.42 s      | 0.037    | 0.043    |
| $f_4$ | 0.32 s      | 0.33 s      | 0.038    | 0.041    |
| $f_5$ | 0.21 s      | 0.23 s      | 0.04     | 0.048    |
| $f_6$ | 0.12 s      | 0.13 s      | 0.031    | 0.036    |

using adaptive threshold in the decision making stage for both fault detection and isolation tasks. Other main advantage of the scheme is the low fault detection delay. The isolation time delay is equal to the detection time delay that verifies the simultaneous fault detection and isolation ability of the proposed scheme.

In the following, the performance of the proposed FDI scheme is compared with some model-based FDI schemes available in the literature. The comparison results were given in Table 5. Comparison results verify the superior performance of the proposed scheme over the recently proposed schemes in the literature. Also, the proposed scheme has less online computational load. This feature makes the application of the proposed scheme possible for real-time implementation. Furthermore, it has reliable performance and can isolate faults upon their detection.

### 5. Conclusion

This paper proposes a new robust fault detection and isolation scheme in a passive manner based on the fuzzy wavelet network with a hybrid design algorithm. It invokes fuzzy wavelet network for modeling normal and faulty behaviors of the system. A new hybrid design algorithm composed of orthogonal least

Table 5. Comparison of schemes.

| Method                                   | Criteria   |                           |                           |                           |
|--|------------|---------------------------|---------------------------|---------------------------|
|  | Robustness | Fault isolation delay     | Online computational load | Offline computation phase |
| The proposed FDI scheme                  | Reliable   | No                        | Low                       | Yes                       |
| NN-based FD scheme [25]                  | Reliable   | Unable to fault isolation | High                      | Yes                       |
| NFN-based FDI scheme [34]                | Unreliable | Yes                       | High                      | Yes                       |
| GOS-based FDI scheme [36]                | Reliable   | No                        | Very high                 | Yes                       |
| Multi ANFIS and NN-based FDI scheme [36] | Reliable   | Yes                       | Low                       | Yes                       |
| RFDI scheme [14]                         | Reliable   | Yes                       | Low                       | Yes                       |

square and artificial bee colony algorithm is proposed to optimize network structure and its parameters, respectively. While the proposed model has high approximation accuracy, there exists model uncertainty because of the model-reality mismatch in the practical applications. In order to achieve robust performance, the bounded error approach inspired from the robust identification theory is utilized to estimate modeling uncertainty both in the detection and isolation steps. The main advantage of the proposed scheme is its ability to handle high order problems regardless of input dimension. Also, it can isolate fault upon its occurrence and no delay is induced during isolation scheme. The presented simulation and comparison results verify the efficiency, reliability, and robustness of the proposed approach. In future research works, development of fault tolerant control scheme based on the proposed fault detection and isolation scheme can be investigated. Also, implementation of the proposed scheme on a real-time system can be considered as another interesting problem.

## References

- Samy, I., Postlethwaite, I. and Gu, D-W. "Survey and application of sensor fault detection and isolation schemes", *Control Eng. Pract.*, **19**(7), pp. 658-674 (2011).
- Komari Alaei, H., Salahshoor, K. and Komari Alaei, H. "A new integrated on-line fuzzy clustering and segmentation methodology with adaptive PCA approach for process monitoring and fault detection and diagnosis", *Soft Comput.*, **17**(3), pp. 345-362 (2012).
- Elshenawy, L. and Awad, H. "Recursive fault detection and isolation approaches of time-varying processes", *Ind. Eng. Chem. Res.*, **51**(29), pp. 9812-9824 (2012).
- Odendaal, H. and Jones, T. "Actuator fault detection and isolation: an optimized parity space approach", *Control Eng. Pract.*, **26**(5), pp. 22-232 (2014).
- Seera, M. and Chee Peng, L. "Online motor fault detection and diagnosis using a hybrid FMM-CART model", *IEEE Trans. Neural. Netw. Learn. Syst.*, **25**(4), pp. 806-812 (2014).
- Blanke, M., Kinnaert, M., Lunze, J. and Staroswiecki, M., *Diagnosis and Fault Tolerant Control*, Springer-Verlag, New York, Inc. Secaucus, NJ, USA (2006).
- Zhang, Y. and Jiang, J. "Bibliographical review on reconfigurable fault-tolerant control systems", *Annu. Rev. Control*, **32**(2), pp. 229-252 (2008).
- Razevi-far, R., Hadi, D., Vasile, P. and Lucas, C. "Model based fault detection and isolation of a steam generator using neuro-fuzzy networks", *Neurocomputing*, **72**(13-15), pp. 2939-2951 (2009).
- Shatnawi, Y. and Al-khassaweneh, M. "Fault diagnosis in internal combustion engines using extension neural networks", *IEEE Trans. Ind. Electron*, **61**(3), pp. 1434-1444 (2014).
- Salahshoor, K., Kordestani, M. and Khoshro, M.S. "Fault detection and diagnosis of an industrial steam turbine using fusion of SVM (support vector machine) and ANFIS (adaptive neuro-fuzzy inference system) classifiers", *Energy*, **35**(12), pp. 5472-5482 (2010).
- Shabaniyan, M. and Montazeri, M. "A neuro-fuzzy online fault detection and diagnosis algorithm for nonlinear and dynamic systems", *Int. J. Control Autom.*, **9**(4), pp. 665-670 (2011).
- Zhang, X., Chen, W., Wang, B. and Chen, X. "Intelligent fault diagnosis of rotating machinery using support vector machine with ant colony algorithm for synchronous feature selection and parameter optimization", *Neurocomputing*, **167**, pp. 260-279 (2015).
- Salahshoor, K., Khoshro, M. and Kordestani, M. "Fault detection and diagnosis of an industrial steam turbine using a distributed configuration of adaptive neuro-fuzzy inference systems", *Simul. Model. Pract. Th.*, **19**(5), pp. 1280-1293 (2011).
- Abbasi Nozari, H., Aliyari Shoorehdeli, M., Simani, S. and Dehghan Banadaki, H. "Model-based robust fault detection and isolation of an industrial gas turbine prototype using soft computing technique", *Neurocomputing*, **91**(15) pp. 29-47 (2012).
- Azadeh, A., Saberi, M., Kazem, A., Ebrahimipour, V., Nourmohammadizadeh, A. and Saberi, Z. "A flexible algorithm for fault diagnosis in centrifugal pump with corrupted data and noise based on ANN and support

- vector machine with hyper-parameters optimization”, *Appl. Soft Comput.*, **13**, pp. 1478-1485 (2013).
16. Shatnawi, Y. and Al-Khassaweneh, M. “Fault diagnosis in internal combustion engines using extension neural network”, *IEEE Trans. Ind. Electron.*, **61**(3), pp. 1434-1443 (2014).
  17. Sugumaran, V. and Ramachandran, K.I. “Fault diagnosis of roller bearing using fuzzy classifier and histogram features with focus on automatic rule learning”, *Expert Syst. Appl.*, **38**, pp. 4901-4907 (2011).
  18. Ho, D.W.C., Zhang, P.A. and Xu, J. “Fuzzy wavelet networks for function learning”, *IEEE Trans. Fuzzy Syst.*, **9**(1), pp. 200-211 (2001).
  19. Ebadat, A., Noroozi, N., Safavi, A.A. and Mousavi, S.H. “New fuzzy wavelet network for modeling and control: the modeling approach”, *Commun. Nonlinear Sci.*, **16**(8), pp. 3385-3396 (2011).
  20. Shahriari-kahkeshi, M., Sheikholeslam, F. and Zekri, M. “Design of adaptive fuzzy wavelet neural sliding mode control for a class of uncertain nonlinear systems”, *ISA Trans.*, **52**(3), pp. 342-350 (2013).
  21. Loussifi, H., Nouri, K. and Braiek, N.B. “A new efficient hybrid intelligent method for nonlinear dynamical systems identification: The wavelet kernel fuzzy neural network”, *Commun. Nonlinear Sci.*, **32**, pp. 10-30 (2016).
  22. Chen, J. and Patton, R., *Robust Model-Based Fault Diagnosis for Dynamic Systems*, Kluwer Academic Publisher Dordrecht (1991).
  23. Patan, K., *Artificial Neural Networks for the Modelling and Fault Diagnosis of Technical Processes*, Springer-Verlag, Berlin (2008).
  24. Milanese, M., Norton, J., Piet-Lahanier, H. and Walter, E., *Bounding Approaches to System Identification*, New York (1996).
  25. Mrugalski, M., Witczak, M. and Korbicz, J. “Confidence estimation of the multi-layer perceptron and its application in fault detection systems”, *Eng. Appl. Artif. Intel.*, **21**(6), pp. 895-906 (2008).
  26. Blesa, J., Puig, V. and Saludes, J. “Robust fault detection using polytope-based set-membership consistency test”, *IET Control Theory Appl.*, **6**(12), pp. 1767-1777 (2012).
  27. Tabatabaeipour, S.M. “Active fault detection and isolation of discrete linear time-varying systems: a set-membership approach”, *Int. J. Syst. Sci.*, **46**(11), pp. 1917-1933 (2015).
  28. Suárez Fábrega, A.J., Bravo Caro, J.M., Abad Herrera, P.J. and Gasca, R.M. “Data-driven bounded-error fault detection”, *Int. J. Adapt. Control*, **28**(12), pp. 1299-1324 (2014).
  29. Santos, D.M. “fault detection method based on bounded error and dynamic threshold techniques”, *Int. J. Adapt. Control*, **30**, pp. 256-270 (2015).
  30. Zhang, Q. “Using wavelet network in nonparametric estimation”, *IEEE Trans. Neural Netw.*, **8**, pp. 227-236 (1997).
  31. Karaboga, D. “An idea based on honey bee swarm for numerical optimization”, Technical Report, TR-06 (2005).
  32. Ozturk, C. and Karaboga, D. “Hybrid artificial bee colony algorithm for neural network training”, *2011 IEEE Congress on Evolutionary Computation*, pp. 84-88 (2011).
  33. Shahriari-kahkeshi, M. and Sheikholeslam, F. “Fuzzy wavelet neural network learning using artificial bee colony algorithm”, *20th Iranian Conference on Electric Engineering* (2012).
  34. Mok, H.T. and Chan, C.W. “Online fault detection and isolation of nonlinear systems based on neuro-fuzzy networks”, *Eng. Appl. Artif. Intel.*, **21**(2), pp. 171-181 (2008).
  35. Zolghadri, A., Henry, D., Cieslak, J., Efimov, D. and Goupil, P., *Fault Diagnosis and Fault-Tolerant Control and Guidance for Aerospace Vehicles, from Theory to Application*, Springer-Verlag London (2014).
  36. Yüksel, T. and Sezgin, A. “Two fault detection and isolation schemes for robot manipulators using soft computing techniques”, *Appl. Soft Comput.*, **10**, pp. 125-134 (2010).

## Biography

**Maryam Shahriari-kahkeshi** received her MS and PhD degrees in Control Engineering from the Isfahan University of Technology, Isfahan, Iran, in 2010 and 2014, respectively. She is currently an Assistant Professor with the Faculty of Engineering, Shahrood University, Shahrood, Iran. Her current research interests include system identification, fault diagnosis, and fault tolerant control systems.

Reproduced with permission of copyright owner. Further reproduction prohibited without permission.