Online Updating Belief-Rule-Base Using the RIMER Approach

Zhi-Jie Zhou, Chang-Hua Hu, Jian-Bo Yang, Dong-Ling Xu, and Dong-Hua Zhou, *Senior Member, IEEE*

***Abstract*—In order to determine the parameters of belief-rule-base (BRB) accurately, several optimization methods have been proposed for training BRB, on the basis of a generic rule-base inference methodology using the evidential reasoning (RIMER) approach. These optimization methods are implemented offline, and such are not suitable for training BRB in a dynamic fashion. In this paper, two recursive algorithms are proposed to update BRB online that can simulate dynamic systems. The main feature of the proposed algorithms is that only partial input and output information is required, which can be incomplete or vague, numerical or judgmental, or mixed. If the internal structure of a BRB is initially decided using expert judgments, domain-specific knowledge and/or commonsense rules, the proposed algorithms can be used to fine-tune the initial BRB online, once input and output datasets become available. Using the proposed algorithms, there is no need to collect a complete set of data before a BRB can be trained, which is necessary if the BRB is used to simulate a dynamic system. A numerical example and a case study are reported to demonstrate the potential of the algorithms for online fault diagnosis.**

***Index Terms*—Belief-rule-base (BRB), evidential reasoning (ER), inference, recursive algorithms, uncertainty.**

I. Introduction

FOR A dynamically changing engineering system, it is difficult to obtain a complete set of historical data for developing a mathematical model for reliable system simulation and consistent decision support [1], [2]. Conventional methods such as time-series analysis and Kalman filter [3]–[5] require accurate and complete data. On the other hand, decision making is a basic human activity and relies on the decision maker's judgments and preferences. It is therefore important to model and analyze decision problems using both numerical data and human judgments that are incomplete and inaccurate in nature [6].

Several methods have been developed to support decision making using human judgments, including IF-THEN rule-based methods [7], fuzzy IF-THEN rule-based methods [8]–[10], and rule-based expert systems [11], [12]. In these methods, human judgments and domain-specific knowledge can be represented in forms of IF-THEN rules. However, these methods cannot deal with ignorance caused by incomplete data or due to human inability to provide accurate judgments. In order to handle ignorance in knowledge-based systems [13], [14], Yang *et al.* proposed a generic rule-base inference methodology using the evidential reasoning (ER) (RIMER) approach to establish a nonlinear relationship between antecedent attributes and an associated consequent [15], [16]. RIMER is based on the ER approach [17]–[21] and can capture the dynamic nature of decision-making problems. The RIMER approach extends traditional IF-THEN rule-based systems for knowledge representation and is capable of capturing vagueness, ignorance, and nonlinear causal relationships. Equipped with the Windows-based and graphically designed intelligent decision system [22], RIMER has already been applied to the safety analysis of offshore systems and the leak detection of oil pipelines [23]–[25].

In RIMER, a basic element is a so-called belief-rule-base (BRB). In BRB, there are several types of parameters including rule weights, attribute weights, and belief degrees. It is difficult to accurately determine the values of these parameters by experts, in particular for a large-scale BRB which has hundreds of rules. A change in a rule weight or an attribute weight may lead to changes in performance of BRB [6]. If system input and output datasets are available, optimization methods can be used to adjust the parameters to improve the performance of BRB. Based on RIMER, Yang *et al.* proposed several optimization methods for training the parameters of BRB, although these methods are implemented in an offline fashion [6].

For a dynamic system, such offline optimization can become expensive and even impractical. There is a need to develop new optimal training methods for updating the parameters of BRB in such a way that the parameters can be updated recursively once new information becomes available. Such a trained BRB can then be used to perform step forward inference for system outputs from given system inputs with improved quality and robustness.

In this paper, two recursive algorithms are proposed for updating BRB online with distributed assessment output and numerical output, respectively. In the development of the

Manuscript received December 25, 2007; revised February 9, 2009; accepted April 15, 2010. Date of publication May 19, 2011; date of current version October 19, 2011. This work was supported in part by the National 973 Project under Grant 2010CB731800 and Grant 2009CB320602, in part by the National Science Foundation of China under Grant 60721003, Grant 60736026, and Grant 61004069, in part by the Program for New Century Excellent Talents in the University of China under Grant 07144, in part by the Foundation of Department of Education of JiLin Province of China under Grant 2009109, and in part by the projects that are supported by the U.K. Engineering and Physical Science Research Council under Grant EP/F024606/1. This paper was recommended by Associate Editor E. P. Blasch.

Z.-J. Zhou is with High-Tech Institute of Xi'an, Xi'an 710025, China, and also with the Department of Automation, TNList, Tsinghua University, Beijing 100084, China (e-mail: zhouzj04@mails.tsinghua.edu.cn).

C.-H. Hu is with High-Tech Institute of Xi'an, Xi'an 710025, China (e-mail: hch6603@263.net).

J.-B. Yang and D.-L. Xu are with the Manchester Business School, University of Manchester, M15 6PB Manchester, U.K. (e-mail: jian-bo.yang@mbs.ac.uk; Ling.Xu@mbs.ac.uk).

D.-H. Zhou is with the Department of Automation, TNList, Tsinghua University, Beijing 100084, China (e-mail: zdh@mail.tsinghua.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCA.2011.2147312

algorithms, it is recognized that belief is represented as probability in the ER approach [26]–[32], and it is assumed that the outputs of a BRB will be independent if its inputs are independent. The independence assumption allows the use of the recursive expectation maximization (EM) algorithm [33]–[37] for the development of the recursive algorithms.

This paper is organized as follows. The RIMER approach is briefly reviewed in Section II. In Section III the two algorithms for updating BRB online are proposed. A numerical example and a case study for pipeline oil leak detection are presented to illustrate the proposed algorithms in Section IV. The paper is concluded in Section V.

II. BRB Inference Using the ER Approach

A. BRB

In order to capture the dynamics of a system, a BRB consisting of a collection of belief rules is defined as follows [16]:

$$\begin{aligned} R_k: &\text{ If } A_1^k \wedge \ldots \wedge A_{M_k}^k, \text{ Then } \{(D_1, \beta_{1,k}), \ldots, (D_N, \beta_{N,k})\} \\ &\text{With a rule weight } \theta_k \text{ and attribute weight } \delta_{1,k}, \ldots, \delta_{M_k,k} \end{aligned} \tag{1}$$

where $A_i^k (i = 1, \ldots, M_k, k = 1, \ldots, L)$ is the referential value of the ith antecedent attribute in the kth rule; and $A_i^k \in A_i$. $A_i = \{A_{i,j}, j = 1, \ldots, J_i\}$ is a set of referential values for the ith antecedent attribute, and J_i is the number of the referential values. $\theta_k (k = 1, \ldots, L)$ is the relative weight of the kth rule, and $\delta_{1,k}, \ldots, \delta_{M_k,k}$ are the relative weights of the M_k antecedent attributes used in the kth rule. $\beta_{j,k} (j = 1, \ldots, N, k = 1, \ldots, L)$ is the belief degree assessed to D_j which denotes the jth consequent. If $\sum_{j=1}^N \beta_{j,k} = 1$, the kth rule is said to be complete; otherwise, it is incomplete. Note that "$\wedge$" is a logical connective to represent the "AND" operator. In addition, suppose that M is the total number of antecedent attributes used in BRB.

Assume that $[\hat{x}_1, \ldots, \hat{x}_M]$ represents the antecedent attribute vector in the kth rule. Then, the kth rule can also be written as

$$\begin{aligned} R_k: &\text{ If } \hat{x}_1 \text{ is } A_1^k \wedge \ldots \wedge \hat{x}_M \text{ is } A_M^k, \\ &\text{Then } \{(D_1, \beta_{1,k}), \ldots, (D_N, \beta_{N,k})\} \\ &\text{With a rule weight } \theta_k \text{ and attribute weight } \delta_{1,k}, \ldots, \delta_{M,k}. \end{aligned} \tag{2}$$

A BRB given in (2) represents a relationship between antecedents and consequents. It allows uncertainty in the relationship to be explicitly modeled and therefore is a more versatile scheme than a simple IF-THEN rule-base for knowledge representation. Note that the weights and the belief degrees can be assigned initially by experts and then trained or updated using dedicated learning algorithms.

B. Belief Rule Inference Using the ER Approach

The RIMER approach can provide an analytical description of relationships between BRB inputs and outputs that could be discrete or continuous, complete or incomplete, linear, nonlinear, or nonsmooth, or their mixture [6], [16]. As shown in Fig. 1, RIMER mainly consists of two main steps, where $\omega_k(n)$ is calculated by (3), $\theta_k (\in R^+, k = 1, \ldots, L)$ is the relative weight of the kth rule, and $\delta_i (\in R^+, i = 1, \ldots, M)$ is the relative weight of the ith antecedent attribute that is used in the kth rule. Because $\omega_k(n)$ will be eventually normalized so that $\omega_k(n) \in [0, 1]$ using (3), θ_k and δ_i can be assigned to any value in R^+. Without loss of generality, however, it is assumed that $\theta_k \in [0, 1]$ and $\delta_i \in [0, 1]$. $\alpha_{i,j}^k(n) (i = 1, \ldots, M)$, which is called the individual matching degree, is the belief degree to its jth referential value $A_{i,j}^k$ in the kth rule at nth step. $\alpha_k(n) = \prod_{i=1}^M (\alpha_{i,j}^k(n))^{\bar{\delta}_i}$ is called the normalized combined matching degree. $\alpha_{i,j}^k(n)$ can be generated in various ways, depending on the nature of an antecedent attribute and data available [6], [15], [16]. $\beta_j(n)$ denotes the belief degree in D_j at time instant n. Note that $\beta_j(n)$ is the function of the rule weights θ_k, the attribute weights $\delta_i (i = 1, \ldots, M)$, the belief degrees $\beta_{i,k} (i = 1, \ldots, N, k = 1, \ldots, L)$, and the input vector $\hat{\mathbf{x}}(n)$.

1) Calculation of the activation weight

The activation weight of the kth rule at time instant n, $\omega_k(n)$, is calculated by [16]

$$\omega_k(n) = \frac{\theta_k \prod_{i=1}^M \left(\alpha_{i,j}^k(n)\right)^{\bar{\delta}_i}}{\sum_{l=1}^L \theta_l \prod_{i=1}^M \left(\alpha_{i,j}^l(n)\right)^{\bar{\delta}_i}} \quad \text{and} \quad \bar{\delta}_i = \frac{\delta_i}{\max\limits_{i=1,\ldots,M}\{\delta_i\}} \tag{3}$$

2) Rule inference using the evidential reasoning approach

Using the analytical ER algorithms [6], [38], the final conclusion $O(\mathbf{Y}(n))$ that is generated by aggregating all rules that are activated by the actual input vector $\hat{\mathbf{x}}(n)$ at time instant n can be represented as follows:

$$O(\mathbf{Y}(n)) = f(\hat{\mathbf{x}}(n)) = \{(D_j, \beta_j(n)), \quad j = 1, \ldots, N\} \tag{4}$$

$$\beta_j(n) = \frac{\mu(n) \times \prod_{k=1}^L \left(\omega_k(n)\beta_{j,k} + 1 - \omega_k(n)\sum_{i=1}^N \beta_{i,k}\right)}{1 - \mu(n) \times \left[\prod_{k=1}^L (1 - \omega_k(n))\right]} - \frac{\mu(n) \times \prod_{k=1}^L \left(1 - \omega_k(n)\sum_{i=1}^N \beta_{i,k}\right)}{1 - \mu(n) \times \left[\prod_{k=1}^L (1 - \omega_k(n))\right]} \tag{5}$$

$$\mu(n) = \left[\sum_{j=1}^N \prod_{k=1}^L \left(\omega_k(n)\beta_{j,k} + 1 - \omega_k(n)\sum_{i=1}^N \beta_{i,k}\right) - (N-1)\prod_{k=1}^L \left(1 - \omega_k(n)\sum_{i=1}^N \beta_{i,k}\right)\right]^{-1} \tag{6}$$

Fig. 1. Block diagram of belief rule inference using ER approach.

III. Recursive Algorithms for Online Updating BRB Systems

In this section, based on the recursive EM algorithm which is a maximum likelihood (ML) algorithm in nature, the RIMER approach is extended by developing two recursive algorithms to update BRB online. The EM algorithm was proposed by Dempster *et al.* [33]. Titterington *et al.* suggested the recursive EM algorithm and proved its weak consistency [34].

In the proposed algorithms, datasets of system inputs and outputs are required. Similar to [6], we assume that a set of data pairs $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is available, where $\hat{\mathbf{x}}$ is a given input vector, $\hat{\mathbf{y}}$ is the corresponding given output vector, either assessed by experts or measured using instruments, and $\mathbf{y}$ is the simulated output that is generated by BRB. $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ can be either judgmental or numerical. We shall conduct case studies using both judgmental and numerical information.

A. Independence Assumption on Input and Output of the BRB System

In order to use the recursive EM algorithm to develop recursive BRB updating algorithms, we first discuss the independence assumption for belief distributions.

In the ER approach, evidence is represented by belief degrees assigned to mutually exclusive grades, and then all pieces of evidence are aggregated. Halpern *et al.* [32] proposed two useful but quite different ways of interpreting belief functions. The first is that a belief function is interpreted as a generalized probability function [26], [29]–[32]. In other words, it is an inner measure induced by a probability function. The second is that a belief function is used as a way for representing evidence [30], [32]. In the ER approach, it is recognized that belief in a proposition increases with the accumulation of evidence supporting the proposition [32].

In the proposed algorithms, we assume that if the inputs of a BRB (pieces of evidence) are independent and are represented as belief distributions or generalized probability distributions, the outputs of the BRB (the results of the ER algorithm) are also represented as belief distributions and are independent. This assumption provides a basis to use the ML algorithm to update BRB.

B. Recursive Parameter Estimation Algorithm Based on Distributed Output

In this case, $\hat{\mathbf{y}}(n)$ is judgmental and represented using a distributed assessment with belief degrees as follows:

$$\hat{\mathbf{y}}(n) = \left\{ \left(D_j, \hat{\beta}_j(n)\right), j = 1, \ldots, N \right\} \tag{7}$$

where D_j is a referential (linguistic) term in the consequent part of a rule; and $\hat{\beta}_j(n)$ is the belief degree to which D_j is assessed for the observed data at time instant n. This is indeed the default output format of RIMER, which provides a panoramic view about output. This format is useful to describe the distributed assessment of BRB output given by experts [6]. For simplicity, we use "the distributed output" to represent "the distributed assessment of BRB output" in the following study.

Let $\hat{\mathbf{B}}(n) = [\hat{\beta}_1(n), \ldots, \hat{\beta}_N(n)]^T$. It is assumed that $\hat{\beta}_j(j = 1, \ldots, N)$ is a random variable. Furthermore, assume that the conditional probability density function (pdf) of $\hat{\mathbf{B}}$ is $f(\hat{\mathbf{B}}|\mathbf{x}, \mathbf{Q})$, and $\mathbf{Q}$ is the unknown parameter vector. According to the assumption given in Section III-A, if the inputs of a BRB $(\hat{\mathbf{x}}(1), \ldots, \hat{\mathbf{x}}(n))$ are independent, its distributed outputs $(\hat{\mathbf{B}}(1), \ldots, \hat{\mathbf{B}}(n))$ are also independent. Then, there is

$$f\left(\hat{\mathbf{B}}(1), \ldots, \hat{\mathbf{B}}(n)|\hat{\mathbf{x}}(1), \ldots, \hat{\mathbf{x}}(n), \mathbf{Q}\right) = \prod_{\tau=1}^{n} f\left(\hat{\mathbf{B}}(\tau)|\hat{\mathbf{x}}(\tau), \mathbf{Q}\right) \tag{8}$$

According to (8), the expectation of the log-likelihood at time instant n can be defined as

$$\begin{aligned} &L_{n+1}(\mathbf{Q}) \\ &\quad \triangleq E\left\{\log \prod_{\tau=1}^{n} f\left(\hat{\mathbf{B}}(\tau)|\hat{\mathbf{x}}(\tau), \mathbf{Q}\right) |\hat{\mathbf{x}}(1), \ldots, \hat{\mathbf{x}}(n), \mathbf{Q}(n)\right\} \\ &\quad = E\left\{\sum_{\tau=1}^{n} \log f\left(\hat{\mathbf{B}}(\tau)|\hat{\mathbf{x}}(\tau), \mathbf{Q}\right) |\hat{\mathbf{x}}(1), \ldots, \hat{\mathbf{x}}(n), \mathbf{Q}(n)\right\} \end{aligned} \tag{9}$$

where $E\{\bullet|\bullet\}$ denotes the conditional expectation at $\mathbf{Q} = \mathbf{Q}(n)$.

Now consider the recursive formulation. The expectation of the log-likelihood in (9) can be written as

$$L_{n+1}(\mathbf{Q}) = L_n(\mathbf{Q}) + E\left\{\log f\left(\hat{\mathbf{B}}(n)|\hat{\mathbf{x}}(n), \mathbf{Q}\right) |\hat{\mathbf{x}}(n), \mathbf{Q}(n)\right\}. \tag{10}$$

Define

$$\Gamma_1(\mathbf{Q}(n)) \triangleq \nabla_{\mathbf{Q}} \log f\left(\hat{\mathbf{B}}(n)|\hat{\mathbf{x}}(n), \mathbf{Q}(n)\right) \tag{11}$$

$$\Xi_1\left(\mathbf{Q}(n)\right) \triangleq E\left\{-\nabla_{\mathbf{Q}} \nabla_{\mathbf{Q}}^T \log f\left(\hat{\mathbf{B}}(n)|\hat{\mathbf{x}}(n), \mathbf{Q}\right) |\hat{\mathbf{x}}(n), \mathbf{Q}(n)\right\} \tag{12}$$

where $\nabla_{\mathbf{Q}}$ is a column gradient operator with respect to $\mathbf{Q}$. Based on (10)–(12), the following recursive algorithm for estimating the parameter vector $\mathbf{Q}$ can be obtained:

$$\mathbf{Q}(n+1) = \mathbf{Q}(n) + \frac{1}{n} \left[\Xi_1\left(\mathbf{Q}(n)\right)\right]^{-1} \Gamma_1\left(\mathbf{Q}(n)\right) \tag{13}$$

where $\mathbf{Q}$ consists of the rule weights, attribute weights, belief degrees, and other possible parameters as given later. The detailed processes for generating the recursive algorithm in (13) are given in Appendix A.

The rule weights, attribute weights, and belief degrees must satisfy the following equality and inequality constraints [6], [16]:

1) A rule weight is normalized, so that it is between zero and one, i.e.,

$$0 \le \theta_k \le 1, \quad k = 1, \ldots, L. \tag{13a}$$

2) An attribute weight is normalized, so that it is between zero and one, i.e.,

$$0 \le \overline{\delta}_m \le 1, \quad m = 1, \ldots, M. \tag{13b}$$

3) A belief degree (subjective probability) must not be less than zeros or more than one, i.e.,

$$0 \leq \beta_{j.k} \leq 1, \qquad j = 1, \ldots, N, \qquad k = 1, \ldots, L. \quad (13c)$$

4) If the kth belief rule is complete, its total belief degree in the consequent will be equal to one; otherwise, the total belief degree is less than one, i.e.,

$$\sum_{j=1}^{N} \beta_{j,k} \leq 1, \qquad k = 1, \ldots, L. \quad (13d)$$

Hence, the algorithm (13) should be revised as follows:

$$\mathbf{Q}(n+1) = \prod_{H_1} \left\{ \mathbf{Q}(n) + \frac{1}{n} \left[\Xi_1\left(\mathbf{Q}(n)\right)\right]^{-1} \Gamma_1\left(\mathbf{Q}(n)\right) \right\} \quad (14)$$

where $\prod_{H_1}\{\bullet\}$ is the projection onto the constraint set H_1 which is composed of constraints (13a)–(13d).

C. Recursive Parameter Estimation Algorithm Based on Numerical Output

In this case, $\hat{\mathbf{y}}(n)$ is a numerical value. If the inputs of a BRB are independent, its true outputs, $\hat{\mathbf{y}}(1), \ldots, \hat{\mathbf{y}}(n)$, can also be assumed to be independent, so there is

$$f\left(\hat{\mathbf{y}}(1), \ldots, \hat{\mathbf{y}}(n) | \hat{\mathbf{x}}(1), \ldots, \hat{\mathbf{x}}(n), \mathbf{Q}\right) = \prod_{\tau=1}^{n} f\left(\hat{\mathbf{y}}(\tau) | \hat{\mathbf{x}}(\tau), \mathbf{Q}\right) \quad (15)$$

where $f(\hat{\mathbf{y}}(\tau)|\hat{\mathbf{x}}(\tau), \mathbf{Q})$ is the pdf of $\hat{\mathbf{y}}(\tau)$ at time instant τ.

Similar to the deducing process in Section III-B and Appendix A, the following recursive algorithm can be obtained:

$$\mathbf{Q}(n+1) = \prod_{H_2} \left\{ \mathbf{Q}(n) + \frac{1}{n} \left[\Xi_2\left(\mathbf{Q}(n)\right)\right]^{-1} \Gamma_2\left(\mathbf{Q}(n)\right) \right\} \quad (16)$$

where $\mathbf{Q}$ also consists of the rule weights, attribute weights, belief degrees, and other possible parameters. H_2 represents the constraint set which is composed of constraints (13a)–(13d) and other possible constraints as given later. We also have

$$\Gamma_2(\mathbf{Q}(n)) \triangleq \nabla_{\mathbf{Q}} \log f(\hat{\mathbf{y}}(n)|\hat{\mathbf{x}}(n), \mathbf{Q}(n)) \quad (16a)$$

$$\Xi_2\left(\mathbf{Q}(n)\right) \triangleq E\left\{-\nabla_{\mathbf{Q}} \nabla_{\mathbf{Q}}^{T} \log f\left(\hat{\mathbf{y}}(n)|\hat{\mathbf{x}}(n), \mathbf{Q}\right) | \hat{\mathbf{x}}(n), \mathbf{Q}(n)\right\} \quad (16b)$$

D. Recursive Algorithms Under Normal Distribution

In the recursive algorithms (14) and (16), the pdfs of $\hat{\mathbf{B}}$ and $\hat{\mathbf{y}}$ need to be known. In this aection, we assume that $\hat{\mathbf{B}}$ and $\hat{\mathbf{y}}$ are random variables and follow normal distribution. Based on our experience, the rationality of this assumption can be given as follows:

(a) If BRB output is represented by a distributed assessment, according to Section III-B, the proposed recursive algorithm under distributed output is indeed a ML approach, and normal distribution is usually assumed in a ML approach.

(b) If BRB output is numerical, in this paper, numerical output is mainly referred to the observation of a system obtained using a sensor, and it is random. Therefore, it is reasonable to assume normal distribution in this case.

Of course, we can assume that BRB output may obey other distributions in specific situations.

1) Recursive Algorithm Under Distributed Output and Normal Distribution: A subjective conclusion generated by aggregating the activated rules can also be represented using the same referential terms as for the observed output $\hat{\mathbf{y}}(n)$ as follows:

$$\mathbf{y}(n) = \{(D_j, \beta_j(n)), \qquad j = 1, \ldots, N\} \quad (17)$$

where $\beta_j(n)$ is generated by BRB using (5) for a given input. In this case, there is $\mathbf{y}(n) = O(\mathbf{Y}(n))$.

It is desirable that for a given input $\hat{\mathbf{x}}(n)$, a BRB can generate an output, represented by (17), which can be as close to $\hat{\mathbf{y}}(n)$ as possible. In other words, for the given data pairs $(\hat{\mathbf{x}}(n), \hat{\mathbf{y}}(n))$ at time instant n, the parameters of a BRB are updated to minimize the difference between the observed belief degree $\hat{\beta}_j(n)$ and the belief degree $\beta_j(n)$ generated by BRB for each referential term. Here, $\hat{\beta}_j(n)$ can be considered as a random variable and $\beta_j(n)$ as its expectation. Define $\hat{\mathbf{B}}(n) \triangleq [\hat{\beta}_1(n), \ldots, \hat{\beta}_N(n)]^T$. Suppose that $\hat{\mathbf{B}}(n)$ follows the following complex normal distribution:

$$f\left(\hat{\mathbf{B}}(n)|\mathbf{x}(n), \mathbf{Q}\right) = (2\pi)^{\frac{-N}{2}} |\Sigma|^{\frac{-1}{2}} \times \exp\left\{-\frac{1}{2}\left(\hat{\mathbf{B}}(n) - \mathbf{B}(n)\right)^T \Sigma^{-1} \left(\hat{\mathbf{B}}(n) - \mathbf{B}(n)\right)\right\} \quad (18)$$

where $\hat{\mathbf{B}}(n)$ is the given distributed output; and $\mathbf{B}(n) = [\beta_1(n), \ldots, \beta_N(n)]^T$ is generated by BRB using (5) for a given input. $\mathbf{Q}$ is a parameter vector and is composed of the parameter vector $\mathbf{V} = [\theta_k, \overline{\delta}_m, \beta_{j,k}]^T$ and the entries of the covariance matrix $\sum$ which is symmetrically positive definite. $\mathbf{V}$ is included in $\mathbf{B}(n)$ and $k = 1, \ldots, L$, $m = 1, \ldots, M$, and $j = 1, \ldots, N$.

Due to independence between the elements of $\mathbf{V}$ and the entries of $\sum$, $\Gamma_1(\mathbf{Q}(n))$, and $\Xi_1(\mathbf{Q}(n))$ in (13) can be written as

$$\Gamma_1\left(\mathbf{Q}(n)\right) = \left[\Gamma_1'\left(\mathbf{Q}(n)\right)^T, \Gamma_1''\left(\mathbf{Q}(n)\right)^T\right]^T \quad (19)$$

$$\Xi_1\left(\mathbf{Q}(n)\right) = \begin{bmatrix} \Xi_1'\left(\mathbf{Q}(n)\right) & \mathbf{0} \\ \mathbf{0} & \Xi_1''\left(\mathbf{Q}(n)\right) \end{bmatrix} \quad (20)$$

where $\Gamma_1'(\mathbf{Q}(n))$ and $\Xi_1'(\mathbf{Q}(n))$ are the derivatives with respect to $\mathbf{V}$. $\Gamma_1''(\mathbf{Q}(n))$ and $\Xi_1''(\mathbf{Q}(n))$ are the derivatives with respect to the entries of $\sum$. Obviously, there is

$$\left[\Xi_1\left(\mathbf{Q}(n)\right)\right]^{-1} = \begin{bmatrix} \left[\Xi_1'\left(\mathbf{Q}(n)\right)\right]^{-1} & \mathbf{0} \\ \mathbf{0} & \left[\Xi_1''\left(\mathbf{Q}(n)\right)\right]^{-1} \end{bmatrix}. \quad (20a)$$

When we consider parameter vector $\mathbf{V}$ only, according to (19) and (20a), the recursive algorithm (13) changes to the following form:

$$\mathbf{V}(n+1) = \mathbf{V}(n) + \frac{1}{n}\left[\Xi_1'\left(\mathbf{Q}(n)\right)\right]^{-1} \Gamma_1'\left(\mathbf{Q}(n)\right) \quad (21)$$

In (21), $\mathbf{V}(n)$ is known. By definitions (11) and (12), the ath element of the gradient vector $\Gamma_1'(\mathbf{Q}(n))$ and the entries of $\Xi_1'(\mathbf{Q}(n))$ at time instant n are given by

$$[\Gamma_1'(\mathbf{Q}(n))]_a = \frac{\partial \mathbf{B}(n)^T}{\partial V_a} \sum(n)^{-1}\left(\hat{\mathbf{B}}(n) - \mathbf{B}(n)\right)\Big|_{\mathbf{V}=\mathbf{V}(n)} \tag{21a}$$

$$\begin{aligned}[\Xi_1'(\mathbf{Q}(n))]_{a,b} &= E\left\{\frac{\partial \mathbf{B}(n)^T}{\partial V_a}\sum(n)^{-1}\frac{\partial \mathbf{B}(n)}{\partial V_b} - \frac{\partial^2 \mathbf{B}(n)^T}{\partial V_a \partial V_b}\right.\\ &\qquad \left.\times \sum(n)^{-1}\left(\hat{\mathbf{B}}(n) - \mathbf{B}(n)\right)|\mathbf{Q}(n)\right\}\\ &= \frac{\partial \mathbf{B}(n)^T}{\partial V_a}\sum(n)^{-1}\frac{\partial \mathbf{B}(n)}{\partial V_b}\Big|_{\mathbf{V}=\mathbf{V}(n)}\end{aligned} \tag{21b}$$

where $a = 1, \ldots, L + M + L \times N$ and $b = 1, \ldots, L + M + L \times N$. The derivatives in (21a) and (21b) are given in Appendix B.

In (21a) and (21b), the covariance matrix $\sum(n)$ is required. Because the belief degrees $\hat{\beta}_1(n), \ldots, \hat{\beta}_N(n)$ should satisfy the constraint $\sum_{j=1}^{N} \hat{\beta}_j(n) \le 1$, they are not independent. In order to simplify the calculation, without loss of generality, we suppose that $\sum = (a_{i,j})_{N\times N}$ satisfies

$$\begin{cases} a_{i,j} = \sigma_1, & i = j\\ a_{i,j} = \sigma_2, & i \ne j.\end{cases} \tag{22}$$

Therefore, under this assumption, there is $\mathbf{Q} = [\mathbf{V}^T, \sigma_1, \sigma_2]^T$.

When the parameter vector $\mathbf{V}(n)$ is available, $\sigma_i(n)$ can be estimated as follows:

$$\sigma_i(n) = \arg\max_{\sigma_i} \log f\left(\hat{\mathbf{B}}(n)|\mathbf{x}(n), \mathbf{Q}\right)\Big|_{\mathbf{V}=\mathbf{V}(n)} \tag{23}$$

where $i = 1, 2$. The details of the algorithm to estimate $\sigma_i(n)$ are given in Appendix C.

Because $\mathbf{V}$ should satisfy constraints (13a)–(13d), the recursive algorithm with the constraints given in (14) should be adopted. First, let $\mathbf{V} = [V_1, \ldots, V_{U_1}]^T$, where $U_1 = L + M + L \times N$. Constraints (13a)–(13c) can be represented as

$$h_i^1(\mathbf{V}) = -V_i \le 0, \quad i = 1, \ldots, S_1, S_1 = U_1 \tag{24}$$
$$h_i^2(\mathbf{V}) = V_i - 1 \le 0, \quad i = 1, \ldots, S_2, S_2 = U_1. \tag{25}$$

Constraint (13d) can be represented as

$$h_k^3(\mathbf{V}) = h_k^3\left(V_{L+M+(k-1)\times N+1}, \ldots, V_{L+M+(k-1)\times N+N}\right) \le 0, \quad k = 1, \ldots, S_3, S_3 = L \tag{26}$$

where

$$h_k^3\left(V_{L+M+(k-1)\times N+1}, \ldots, V_{L+M+(k-1)\times N+N}\right) = \sum_{j=1}^{N} \beta_{j,k} - 1. \tag{27}$$

Define

$$\tilde{h}_{s_j}^j(\mathbf{V}) \triangleq \max\left[0, h_{s_j}^j(\mathbf{V})\right] \text{ and}$$
$$\Psi_j(\mathbf{V}) = \sum_{s_j=1}^{S_j}\left[\tilde{h}_{s_j}^j(\mathbf{V})\right]^2 \tag{28}$$

$$\phi_j(\mathbf{V}) = \left[\frac{\partial \Psi_j(\mathbf{V})}{\partial V_1}, \ldots, \frac{\partial \Psi_j(\mathbf{V})}{\partial V_{U_1}}\right] \tag{29}$$

where $j = 1, 2, 3$.

Suppose $\mathbf{I}_u$ is an identity matrix whose dimension is u. The recursive algorithm (21) is revised and the following algorithm for dealing with constraints (24)–(26) can be obtained [35], [36]:

$$\begin{aligned}\mathbf{V}(n+1) = \mathbf{V}(n) + \frac{\alpha_1}{n}&\left\{\boldsymbol{\pi}_1(\mathbf{V}(n))\left[\Xi_1'(\mathbf{Q}(n)) + \gamma_1 \mathbf{I}_{U_1}\right]^{-1}\right.\\ &\left.\times \Gamma_1'(\mathbf{Q}(n)) - \frac{K_1}{2}\phi(\mathbf{V}(n))\right\}\end{aligned} \tag{30}$$

where $\phi(\mathbf{V}(n)) = \sum_{j=1}^{3}\phi_j(\mathbf{V}(n))$. $\alpha_1 \ge 1$ is the step factor and can change the convergence speed. Because only some rules in BRB may be activated and the matrix $\Xi_1'(\mathbf{Q}(n))$ may be singular at time instant n, $\Xi_1'(\mathbf{Q}(n))$ is amended using $\gamma_1\mathbf{I}_{U_1}$ so that it becomes positive definite and $\gamma_1 > 0$. K_1 denotes a positive real number, and its value may change from case to case. There is

$$\begin{aligned}\boldsymbol{\pi}_1(\mathbf{V}(n)) &= \mathbf{I}_{U_1} - \mathbf{H}_1(\mathbf{V}(n))^T\\ &\quad \times \left(\mathbf{H}_1(\mathbf{V}(n))\mathbf{H}_1(\mathbf{V}(n))^T\right)^{-1}\mathbf{H}_1(\mathbf{V}(n))\end{aligned} \tag{31}$$

where $\mathbf{H}_1(\mathbf{V}(n))$ denotes the Jacobian matrix of $\mathbf{h}_1(\mathbf{V}(n))$ that denotes the value of $\mathbf{h}_1(\mathbf{V})$ at $\mathbf{V} = \mathbf{V}(n)$. $\mathbf{h}_1(\mathbf{V})$ is defined as

$$\begin{aligned}\mathbf{h}_1(\mathbf{V}) \triangleq \big[&h_1^1(\mathbf{V}), \ldots, h_{S_1}^1(\mathbf{V}), h_1^2(\mathbf{V}), \ldots, h_{S_2}^2(\mathbf{V}),\\ &h_1^3(\mathbf{V}), \ldots, h_{S_3}^3(\mathbf{V})\big]^T.\end{aligned} \tag{32}$$

As a result of the aforementioned discussion, the procedure of the proposed recursive algorithms for online updating the BRB parameters based on the distributed output and the normal distribution assumption is summarized as the following steps:

- Step 1) Give the initial values of the parameter vector $\mathbf{V}(0)$ and the covariance matrix $\sum(0)$. $\mathbf{V}(0)$ must satisfy constraints (13a)–(13d).
- Step 2) When the data $\hat{\mathbf{x}}(0)$ and $\hat{\mathbf{y}}(0)$ are given, the recursive algorithm (30) is used to estimate $\mathbf{V}(1)$. Then, $\sum(1)$ is estimated using (23).
- Step 3) When $\hat{\mathbf{x}}(n)$, $\hat{\mathbf{y}}(n)$, $\mathbf{V}(n)$, and $\sum(n)$ are available at time instant n. *Step 2* is reused to estimate $\mathbf{V}(n+1)$.
- Step 4) Once BRB is updated, it can be used to perform inference from given inputs.

2) Recursive Algorithm Under Numerical Output and Normal Distribution: The output $O(\mathbf{Y}(n))$, as shown in (4), is represented as a distribution, and its average score is calculated by [15], [16], [21]

$$\mathbf{y}(n) = \sum_{j=1}^{N} \mu_j(n)\beta_j(n) \tag{33}$$

where $\mu_j(n)$ denotes the utility (or score) of an individual consequent D_j which can be either given using a scale or estimated using the decision maker's preferences.

It is desirable that for a given input, $\hat{\mathbf{x}}(n)$, a BRB can generate an output, represented as (33), which can be as close to $\hat{\mathbf{y}}(n)$ as possible. Here, $\hat{\mathbf{y}}(n)$ is considered as a random variable, and $\mathbf{y}(n)$ can be considered as its expectation. We assume that the pdf of $\hat{\mathbf{y}}(n)$ obey the following normal distribution:

$$f\left(\hat{\mathbf{y}}(n)|\mathbf{x}(n),\mathbf{Q}\right)=\frac{1}{\sqrt{2\pi\sigma}}\exp\left\{-\frac{\left(\hat{\mathbf{y}}(n)-\mathbf{y}(n)\right)^2}{2\sigma}\right\} \quad (34)$$

where $\mathbf{Q}=[\mathbf{W}^T,\sigma]^T$ denotes the parameter vector. $\mathbf{W}=[\mathbf{V}^T,\mu_1,\ldots,\mu_N]^T$ and σ denotes the variance.

Similar to (19)–(21), due to independence between the elements of $\mathbf{W}$ and σ, the recursive algorithm (16) can also be changed into the following form when we consider only $\mathbf{W}$:

$$\mathbf{W}(n+1)=\prod_{H_2}\left\{\mathbf{W}(n)+\frac{1}{n}\left[\Xi_2'\left(\mathbf{Q}(n)\right)\right]^{-1}\Gamma_2'\left(\mathbf{Q}(n)\right)\right\} \quad (35)$$

where $\Gamma_2'(\mathbf{Q}(n))$ and $\Xi_2'(\mathbf{Q}(n))$ are the derivatives with respect to $\mathbf{W}$.

Let $\mathbf{W}=[W_1,\ldots,W_{U_2}]$ and $U_2=L+M+L\times N+N$. In (35), according to definitions (16a) and (16b), $\Gamma_2'(\mathbf{Q}(n))$ and $\Xi_2'(\mathbf{Q}(n))$ have the following forms:

1) If $a,b=1,\ldots,U_1$ and $U_1=L+M+L\times N$, the ath element of the gradient vector $\Gamma_2'(\mathbf{Q}(n))$ and the entries of $\Xi_2'(\mathbf{Q}(n))$ at time instant n are given by

$$\left[\Gamma_2'\left(\mathbf{Q}(n)\right)\right]_a=\frac{\left(\hat{\mathbf{y}}(n)-\mathbf{y}(n)\right)}{\sigma(n)}\times\sum_{j=1}^{N}\mu_j(n)\left.\frac{\partial\beta_j(n)}{\partial W_a}\right|_{\mathbf{W}=\mathbf{W}(n)} \quad (35a)$$

$$\begin{aligned}\left[\Xi_2'\left(\mathbf{Q}(n)\right)\right]_{a,b}&=E\left\{\frac{1}{\sigma}\frac{\partial\mathbf{y}(n)}{\partial W_a}\frac{\partial\mathbf{y}(n)}{\partial W_b}-\frac{1}{\sigma}\frac{\partial^2\mathbf{y}(n)}{\partial W_a\partial W_b}\times\left(\hat{\mathbf{y}}(n)-\mathbf{y}(n)\right)|\mathbf{Q}(n)\right\}\\&=\frac{1}{\sigma(n)}\left[\sum_{j=1}^{N}\mu_j(n)\frac{\partial\beta_j(n)}{\partial W_a}\right]\times\left.\left[\sum_{j=1}^{N}\mu_j(n)\frac{\partial\beta_j(n)}{\partial W_b}\right]\right|_{\mathbf{W}=\mathbf{W}(n)}\end{aligned} \quad (35b)$$

where the derivatives are also given in Appendix A.

2) If $a,b=U_1+1,\ldots,U_2$, there are

$$\left[\Gamma_2'\left(\mathbf{Q}(n)\right)\right]_a=\left.\frac{\beta_{a-U_1}(n)\left(\hat{\mathbf{y}}(n)-\mathbf{y}(n)\right)}{\sigma(n)}\right|_{\mathbf{W}=\mathbf{W}(n)} \quad (35c)$$

$$\left[\Xi_2'\left(\mathbf{Q}(n)\right)\right]_{a,b}=\left.\frac{\beta_{a-U_1}(n)\beta_{b-U_1}(n)}{\sigma(n)}\right|_{\mathbf{W}=\mathbf{W}(n)} \quad (35d)$$

Moreover, $\sigma(n)$ is required in (35a)–(35d). If $\hat{\mathbf{x}}(n)$, $\hat{\mathbf{y}}(n)$, and $\mathbf{W}(n)$ are available, it can be estimated by

$$\begin{aligned}\sigma(n)&=\arg\max_{\sigma}\log f\left.\left(\hat{\mathbf{y}}(n)|\hat{\mathbf{x}}(n),\mathbf{Q}\right)\right|_{\mathbf{W}=\mathbf{W}(n)}\\&=\left.\left(\hat{\mathbf{y}}(n)-\mathbf{y}(n)\right)^2\right|_{\mathbf{W}=\mathbf{W}(n)}\end{aligned} \quad (36)$$

In (35), the constraint set H_2 is composed of constraints (24)–(26) and the following two constraints:

1) The more preferred a consequent is, the higher its score, i.e.,

$$\mu_i<\mu_j \text{ if } i<j;\ i,\qquad j=1,\ldots,N. \quad (37)$$

2) For qualitative output, the score (utility) of a consequent can be normalized so that it is between zero and one, i.e.,

$$0\leq\mu_j\leq 1;\qquad j=1,\ldots,N. \quad (38)$$

Now we will consider the following two cases.

Case 1. If output is not qualitative, the constraint set H_2 is composed of constraints (24)–(26) and (37).

Similarly, constraints (24)–(26) can also be written as

$$h_i^1(\mathbf{W})=-W_i\leq 0,\qquad i=1,\ldots,S_1, S_1=U_1 \quad (39)$$

$$h_i^2(\mathbf{W})=W_i-1\leq 0,\qquad i=1,\ldots,S_2, S_2=U_1 \quad (40)$$

$$\begin{aligned}h_k^3(\mathbf{W})&=h_k^3\left(W_{L+M+(k-1)\times N+1},\ldots,W_{L+M+(k-1)\times N+N}\right)\\&=\sum_{j=1}^{N}\beta_{j,k}-1\leq 0,\qquad k=1,\ldots,S_3, S_3=L.\end{aligned} \quad (41)$$

The inequality constraints (37) can be represented as

$$h_g^4(\mathbf{W})=h_g^4\left(W_{U_1+i},W_{U_1+j}\right)=\mu_i-\mu_j<0 \quad (42)$$

where $g=(i-1)(N-1)+j-i-\sum_{k=1}^{i-2}(i-k-1)$, $i=1,\ldots,N-1$, and $j=i+1,\ldots,N$.

Let $S_4=(N-2)(N-1)+1-\sum_{k=1}^{N-3}(N-2-k)$. Also define

$$\tilde{h}_{s_j}^j(\mathbf{W})\triangleq\max\left[0,h_{s_j}^j(\mathbf{W})\right] \text{ and}$$

$$\Psi_j(\mathbf{W})=\sum_{s_j=1}^{S_j}\left[\tilde{h}_{s_j}^j(\mathbf{W})\right]^2 \quad (43)$$

$$\phi_j(\mathbf{W})=\left[\frac{\partial\Psi_j(\mathbf{W})}{\partial W_1},\ldots,\frac{\partial\Psi_j(\mathbf{W})}{\partial W_{U_2}}\right] \quad (44)$$

where $j=1,\ldots,4$.

The recursive algorithm (35) is revised, and the following algorithm for dealing with constraints (39)–(42) is obtained:

$$\begin{aligned}\mathbf{W}(n+1)=\mathbf{W}(n)+\frac{\alpha_2}{n}&\left\{\boldsymbol{\pi}_2\left(\mathbf{W}(n)\right)\left[\Xi_2'\left(\mathbf{Q}(n)\right)+\gamma_2\mathbf{I}_{U_2}\right]^{-1}\right.\\&\left.\times\Gamma_2'\left(\mathbf{Q}(n)\right)-\frac{K_2}{2}\phi'\left(\mathbf{W}(n)\right)\right\}\end{aligned} \quad (45)$$

where $\phi'(\mathbf{W}(n))=\sum_{j=1}^{4}\phi_j(\mathbf{W}(n))$. $\alpha_2\geq 1$ is the step factor. $\Xi_2'(\mathbf{Q}(n))$ is amended using $\gamma_2\mathbf{I}_{U_2}$ so that it becomes positive definite, and $\gamma_2>0$. K_2 denotes a positive real number, and its value may change from case to case. There is

$$\begin{aligned}\boldsymbol{\pi}_2\left(\mathbf{W}(n)\right)=\mathbf{I}_{U_2}-\mathbf{H}_2\left(\mathbf{W}(n)\right)^T&\\\times\left(\mathbf{H}_2\left(\mathbf{W}(n)\right)\mathbf{H}_2\left(\mathbf{W}(n)\right)^T\right)^{-1}\mathbf{H}_2\left(\mathbf{W}(n)\right)&\end{aligned} \quad (46)$$

where $\mathbf{H}_2(\mathbf{W}(n))$ denotes the Jacobian matrix of $\mathbf{h}_2(\mathbf{W}(n))$ that is defined as

$$\mathbf{h}_2(\mathbf{W}) \triangleq \left[h_1^1(\mathbf{W}), \ldots, h_{S_1}^1(\mathbf{W}), \ldots, h_1^4(\mathbf{W}), \ldots, h_{S_4}^4(\mathbf{W})\right]^T \quad (47)$$

Case 2. If output is qualitative, the constraint set H_2 is composed of constraints (38)–(42).

Constraints (38)–(40) can be represented as

$$h_i^5(\mathbf{W}) = -W_i \leq 0, \qquad i = 1, \ldots, S_5, S_5 = U_2 \quad (48)$$
$$h_i^6(\mathbf{W}) = W_i - 1 \leq 0, \qquad i = 1, \ldots, S_6, S_6 = U_2. \quad (49)$$

Then, the constraint set H_2 is composed of constraints (41), (42), (48), and (49). Also define

$$\tilde{h}_{s_j}^j(\mathbf{W}) \triangleq \max\left[0, h_{s_j}^j(\mathbf{W})\right] \text{ and}$$
$$\Psi_j(\mathbf{W}) = \sum_{s_j=1}^{S_j} \left[\tilde{h}_{s_j}^j(\mathbf{W})\right]^2 \quad (50)$$
$$\phi_j(\mathbf{W}) = \left[\frac{\partial \Psi_j(\mathbf{W})}{\partial W_1}, \ldots, \frac{\partial \Psi_j(\mathbf{W})}{\partial W_{U_2}}\right] \quad (51)$$

where $j = 3, \ldots, 6$.

Similarly, the recursive algorithm (35) is revised, and the following algorithm for dealing with constraints (41), (42), (48), and (49) can be obtained as follows [35], [36]:

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \frac{\alpha_2}{n}\left\{\boldsymbol{\pi}_3\left(\mathbf{W}(n)\right)\left[\Xi_2'\left(\mathbf{Q}(n)\right) + \gamma_2 \mathbf{I}_{U_2}\right]^{-1} \times \Gamma_2'\left(\mathbf{Q}(n)\right) - \frac{K_2}{2}\phi''\left(\mathbf{W}(n)\right)\right\} \quad (52)$$

where $\phi''(\mathbf{W}(n)) = \sum_{j=3}^{6} \phi_j(\mathbf{W}(n))$, and there is

$$\boldsymbol{\pi}_3\left(\mathbf{W}(n)\right) = \mathbf{I}_{U_2} - \mathbf{H}_3\left(\mathbf{W}(n)\right)^T \times \left(\mathbf{H}_3\left(\mathbf{W}(n)\right)\mathbf{H}_3\left(\mathbf{W}(n)\right)^T\right)^{-1}\mathbf{H}_3\left(\mathbf{W}(n)\right) \quad (53)$$

where $\mathbf{H}_3(\mathbf{W}(n))$ denotes the Jacobian matrix of $\mathbf{h}_3(\mathbf{W}(n))$ that is defined as

$$\mathbf{h}_3(\mathbf{W}) \triangleq \left[h_1^3(\mathbf{W}), \ldots, h_{S_3}^3(\mathbf{W}), \ldots, h_1^6(\mathbf{W}), \ldots, h_{S_6}^6(\mathbf{W})\right]^T. \quad (54)$$

As a result of the aforementioned discussion, the procedure of the proposed recursive algorithm for online updating BRB parameters based on numerical output and normal distribution assumption may be summarized as the following steps:

Step 1) Give the initial values of the parameter vector $\mathbf{W}(0)$ and the variance $\sigma(0)$. $\mathbf{W}(0)$ must satisfy constraints (13a)–(13d) and (37), or constraints (13a)–(13d), (37) and (38).

Step 2) When the data $\hat{\mathbf{x}}(0)$ and $\hat{\mathbf{y}}(0)$ are given, the recursive algorithm (45) or (52) is used to estimate $\mathbf{W}(1)$. Then, $\sigma(1)$ is estimated using (36).

Step 3) When $\hat{\mathbf{x}}(n)$, $\hat{\mathbf{y}}(n)$, $\mathbf{W}(n)$, and $\sigma(n)$ are available at time instant n. *Step 2* is reused to estimate $\mathbf{W}(n+1)$.

Step 4) Once BRB is updated, it can be used to perform inference from given inputs.

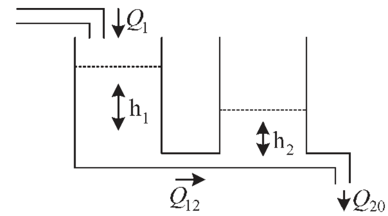


Fig. 2. Layout of two tanks.

Remark: The proposed recursive algorithms for updating BRB parameters online, based on distributed output or numerical output, are stochastic approximation algorithms under the equality and inequality constraints. The convergence theorem of the stochastic approximation algorithm has been proved in the literature [33] and [34]. Therefore, we can prove the convergence of the recursive algorithms in the same way, if the appropriate initial values of the parameters are chosen [37], [38].

IV. NUMERICAL EXAMPLE AND A CASE STUDY

A numerical example is studied in this section to show the implementation of the proposed algorithms. Furthermore, a case study for pipeline oil leak detection is examined to illustrate the algorithms to show that it can be widely applied in engineering.

A. Numerical Example

1) Problem Formulation: A system with two tanks is analyzed, as shown in Fig. 2. Water flows into the first tank at the rate Q_1 (m^3/s), then flows to the second tank at the rate Q_{12} (m^3/s), and finally flows out of the second tank at the rate Q_{20} (m^3/s). Suppose that a slow jam fault occurs at the output of tank 2.

The dynamic model of the two tanks is given as follows:

$$\begin{cases} A\dot{h}_1 = Q_1 - Q_{12} \\ A\dot{h}_2 = Q_{12} - Q_{20} \end{cases} \quad (55)$$

with

$$\begin{cases} Q_{12} = a_1 s \operatorname{sgn}(h_1 - h_2)\sqrt{2g|h_1 - h_2|} \\ Q_{20} = a_2 s \sqrt{2gh_2} \end{cases} \quad (56)$$

where a_i and h_i ($i = 1, 2$) are the outflow coefficients and the liquid level (meters) of the two tanks; $\operatorname{sgn}(z)$ is the sign of the argument z; s is the section area of the connection pipe (m^2); A is the section area of the two tanks (m^2) which are of the same size; and T_s is the sampling period. The values of the parameters used in (56) are shown in Table I.

Suppose that a_2 decreases along the time as follows:

$$\phi(n) = a_2 - 0.001n \quad (57)$$

where $\phi(n)$ is the outflow coefficient of the second tank when the system is in the slow jam fault.

The levels of the two tanks during the slow jam fault are given in Fig. 3. In order to verify the proposed recursive algorithms, we use *Level 1* and *Level 2* as the inputs of a BRB, and the outflow coefficient a_2 which is denoted by *OC*

TABLE I
VALUES OF THE PARAMETERS OF THE DOUBLE TANKS

Technical Parameter	Value	Unit
A	0.15	m^2
s	0.00005	m^2
Q_1	0.00005	m^3/s
T_s	1	s
g	9.8	cm/s^2
a_1	0.4	
a_2	0.5	

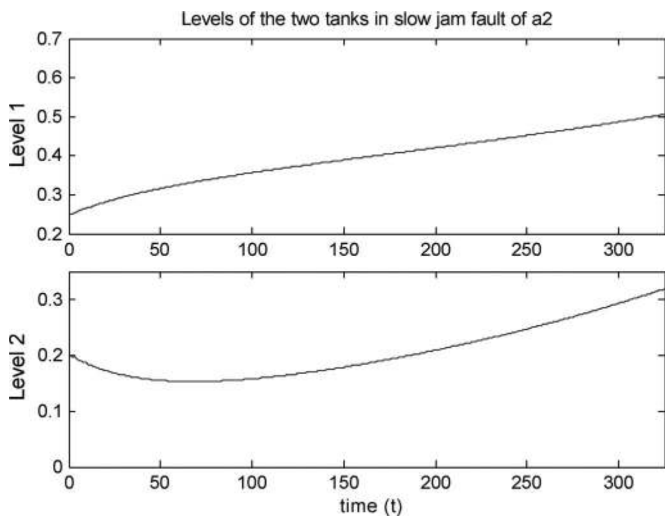


Fig. 3. Levels of the two tanks during the slow jam fault.

TABLE II
REFERENTIAL POINTS OF *Level 1*

Linguistic terms	VS	S	M	L
Numerical values	0.25	0.33	0.44	0.51

as the output. In other words, two levels are considered as the antecedent attributes, and the outflow coefficient is the consequent in BRB.

2) Referential Points of the Antecedent Attributes and Consequent: The antecedent attributes and consequent should be given some referential points. For *Level 1* and *Level 2*, the same referential points are used, and they are very small (VS), small (S), medium (M), and large (L), i.e.,

$$A_i^k \in \{VS, S, M, L\} \tag{58}$$

where $i = 1, 2$.

For OC, two referential points are used, and they are normal (N) and fault (F), i.e.,

$$\mathbf{D} = (D_1, D_2) = (N, F). \tag{59}$$

The referential points defined earlier for the antecedent attributes and the consequent are in linguistic terms and need to be quantified. The quantified results are listed in Tables II–IV, respectively.

TABLE III
REFERENTIAL POINTS OF *Level 2*

Linguistic terms	VS	S	M	L
Numerical values	0.15	0.2	0.23	032

TABLE IV
REFERENTIAL POINTS OF OC

Linguistic terms	N	F
Numerical values	0.5	0.175

3) Rules: For estimating the fault of the two tanks, a belief rule can be represented as

$$\begin{aligned} R_k:\ & \text{If Level 1 is } A_1^k \wedge Level\ 2 \text{ is } A_2^k, \\ & \text{Then } OC \text{ is } \{(\text{normal}, \beta_{1,k}), (\text{fault}, \beta_{2,k})\} \\ & \text{With a rule weight } \theta_k \text{ and attribute weight } \delta_{1,k}, \delta_{2,k} \end{aligned} \tag{60}$$

where A_1^k and A_2^k $(k = 1, \ldots, 4)$ are the referential values as defined in Tables II and III, respectively.

4) Simulation Results Based on Distributed Output: In order to validate the proposed recursive algorithm under distributed output and normal distribution assumption, the following simulation is conducted. In this case, the output is transformed into the following distributed output format:

$$OC = \{(\text{normal}, \beta_1), (\text{fault}, \beta_2)\} \tag{61}$$

where β_1 and β_2 are generated using the quantitative data transformation technique [15].

Three BRBs are constructed for this validation analysis. The first BRB is directly constructed from the given relationship between the system output OC and the two inputs *Level 1* and *Level 2*, as defined by the slow jam fault model (57) and the dynamic model given in (55) and (56). The second BRB is given by an expert. Finally, the initial BRB model given by the expert is trained using the proposed recursive algorithms and the data generated for constructing the first BRB, leading to the third optimally trained BRB.

Step 1—Directly construct a benchmark BRB using the system model given in (55)–(57): For given values of *Level 1* and *Level 2*, the values of OC can be generated from (55)–(57). For example, if at time $n = 1$ s, *Level 1* is VS (very small or 0.25), and *Level 2* is S (small or 0.2), then the value of OC at $n = 1$ s is given by 0.499. This value can be transformed into the following equivalent distribution: $OC = \{(\text{normal}, 0.9969), (\text{fault}, 0.0031)\}$. By equivalent, we mean that the average score of this distribution is equal to 0.449. Therefore, the first belief rule can be constructed as follows:

$$\text{If } Level\ 1 \text{ is VS and } Level\ 2 \text{ is } S, \text{ Then } OC \text{ is } \{(N, 0.9969), (F, 0.0031)\} \tag{62}$$

which is shown in the second row in Table V. In other words, we have $\hat{\beta}_{1,1} = 0.9969$ and $\hat{\beta}_{2,1} = 0.0031$.

Similarly, other belief rules can be generated from (57) and Fig. 3. In Table V, the four belief rules constitute a BRB where

TABLE V
BELIEF DEGREES IN THE BENCHMARK BRB

Rule number	*Level 1* AND *Level 2*	*OC* distribution $\{D_1, D_2\}$={N, F}	The real *OC* value
1	VS and S	$\{(D_1, 0.9969), (D_2, 0.0031)\}$	0.499
2	S and VS	$\{(D_1, 0.7969), (D_2, 0.2031)\}$	0.434
3	M and M	$\{(D_1, 0.2862), (D_2, 0.7138)\}$	0.268
4	L and L	$\{(D_1, 0.0031), (D_2, 0.9969)\}$	0.176

TABLE VI
INITIAL BELIEF DEGREES PROVIDED BY AN EXPERT

Rule number	*Level 1* AND *Level 2*	*OC* distribution $\{D_1, D_2\}$ = {N, F}
1	VS and S	$\{(D_1, 1), (D_2, 0)\}$
2	S and VS	$\{(D_1, 0.6), (D_2, 0.4)\}$
3	M and M	$\{(D_1, 0.4), (D_2, 0.6)\}$
4	L and L	$\{(D_1, 0), (D_2, 1)\}$

all the belief degrees are generated in the same way as shown earlier, and the values of θ_k and $\overline{\delta}_j$ are all set to 1, where $k = 1, \ldots, 4$ and $j = 1, 2$. This BRB can precisely represent the relationship of the outflow coefficient OC with respect to *Level 1* and *Level 2*, at the four discrete points shown in the antecedent conditions of the four belief rules listed in Table V. This BRB is used as a benchmark to check how closely a BRB, which is initialized either randomly or using expert knowledge, can be trained using the proposed recursive training algorithm to simulate the true relationship. Therefore, the BRB shown in Table V is referred to as benchmark BRB for short.

To apply the benchmark BRB to simulate the output OC based on the values of *Level 1* and *Level 2*, the input values $[Level\ 1(n), Level\ 2(n)]$ also need to be transformed and represented in terms of the referential values as defined in (58), [6], [16], [25]. Then, (5) is used to generate the distributed outputs of the benchmark BRB as defined in Table V. The generated distributed outputs are then used to simulate the output OC of the slow jam fault model defined in (57). In order to use the proposed recursive algorithm, it is assumed that the distributed output is a random variable and obeys normal distribution.

Step 2—Set the parameters of the initial BRB: The belief degrees in the initial BRB are given by an expert and listed in Table VI. The initial values of θ_k and $\overline{\delta}_j$ are all set at 1, where $k = 1, \ldots, 4$ and $j = 1, 2$. The initial belief degrees in Table VI are determined by the expert according to the running patterns and historical data of the two tanks and the change of the OC values over time. For example, according to historical information, if *Level 1* is S and *Level 2* is VS, the expert judges that the possibility of the system in the normal (N) state is larger than in the fault (F) state. Therefore, the expert assesses that the belief degree to N is 0.6, and the belief degree to F is 0.4. Thus, the initial belief rule can be obtained as the third row of Table VI. The expert had good knowledge about the dynamics of the two tank system, as well as the RIMER methodology. However, he did not use any optimal training methods to fine-tune the initially given weights and belief degrees.

Fig. 4 shows that the belief degrees to the consequents calculated by (5) using the initial BRB do not well match the distributed outputs generated by the benchmark BRB as defined in Table V. This means that the initial BRB provided by the expert is not good enough. Therefore, it is necessary to use the available information to update the initial BRB online.

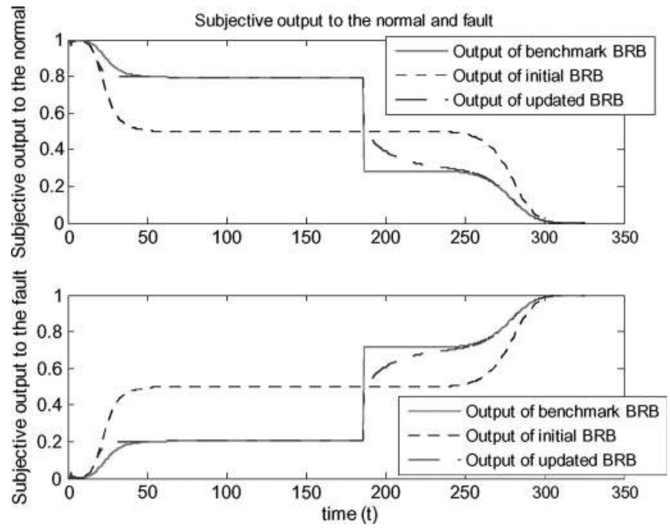


Fig. 4. Distributed outputs generated by the benchmark, initial, and updated BRBs.

TABLE VII
UPDATED RULE WEIGHTS AND BELIEF DEGREES UNDER DISTRIBUTE OUTPUT

Rule number	Updated rule weight	*Level 1* AND *Level 2*	*OC* distribution $\{D_1, D_2\}$ = {N, F}
1	1.0000	VS and S	{(D1, 0.9994), (D2, 0.0006)}
2	0.8390	S and VS	{(D1, 0.7969), (D2, 0.2031)}
3	0.9995	M and M	{(D1, 0.2921), (D2, 0.1079)}
4	1.0000	L and L	{(D1, 0.0017), (D2, 0.9983)}

Step 3—Update the BRB constructed initially using the expert judgments in Step 2: Based on the data generated for constructing the benchmark BRB as shown in *Step 1* and the initial BRB given by the expert in *Step 2*, the recursive algorithm (30) is used to update the initial BRB, in order to examine how closely the initial BRB can be trained to simulate the benchmark BRB. The updated rule weights and belief degrees are listed in Table VII. As shown in Fig. 4, it is obvious that the distributed outputs generated by the updated BRB can match the benchmark BRB more closely than the initial BRB.

Step 4—Quantitative analysis: In order to further demonstrate the proposed recursive algorithms, the mean absolute percentage error (MAPE) [41] is used. The MAPE between the benchmark BRB and the initial BRB in terms of belief degrees to the linguistic term "fault" is 21.04%. On the other hand, the MAPE between the benchmark BRB and the updated BRB in terms of the belief degrees to "fault" is 1.76%. Obviously, the trained BRB can replicate the relationship among *Level 1*, *Level 2*, and the output OC more accurately than the initial BRB.

Step 5—Convergence analysis of the proposed recursive algorithm: In order to study the performance of the recursive algorithm, we choose the mean squared error (MSE) [41], which is defined as $\|\beta_{j,k}(n) - \hat{\beta}_{j,k}\|^2$ to measure the estimation accuracy, where $\| \bullet \|$ denotes Euclidean norm. As

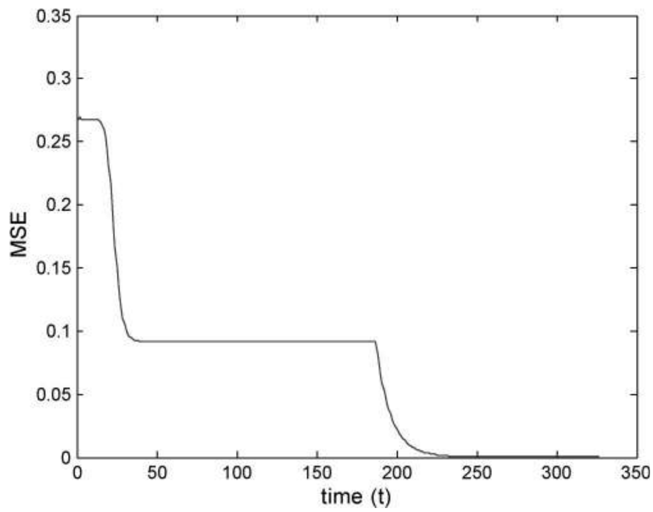


Fig. 5. MSE between the distributed outputs generated by the benchmark and updated BRBs.

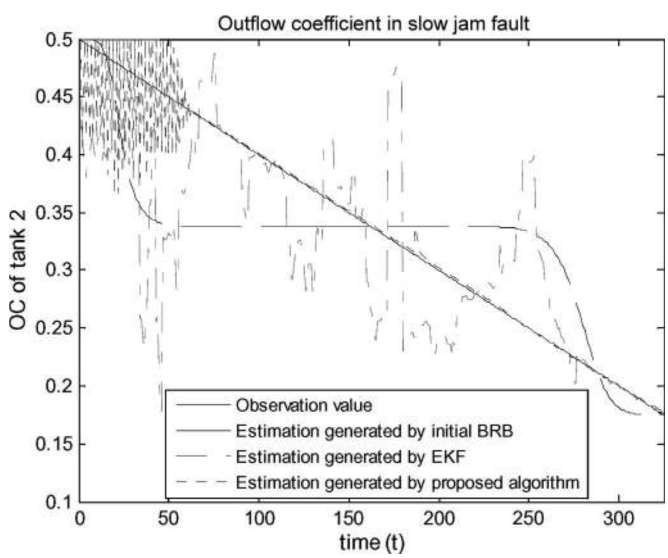


Fig. 6. Observed OC values and the estimated OC values generated by the initial and updated BRBs.

shown in Fig. 5, if the appropriate initial parameters are chosen, the estimates generated by the proposed recursive algorithm converge to the parameters in the benchmark BRB as defined in Table V quickly.

5) Simulation Results Based on Numerical Output: In order to validate the proposed recursive algorithm under numerical output and normal distribution assumption, we will give the following simulation.

Step 1—Set initial parameters: The initial rule weights, attribute weights, and belief degrees are the same as for the initial BRB constructed in *Step 2* of the previous section. As shown in Fig. 6, it is obvious that the estimated values of OC generated by the initial BRB do not match the observed values. This means that the initial BRB provided by an expert is indeed rather bad. Therefore, it is necessary to update the initial BRB online.

Step 2—Update the initial BRB: After the input values $[Level\ 1(n), Level\ 2(n)]$ are transformed and represented in

TABLE VIII
UPDATED RULE WEIGHTS AND BELIEF DEGREES UNDER NUMERICAL OUTPUT AND PROPOSED ALGORITHM

Rule number	Updated rule weight	*Level 1* AND *Level 2*	*OC* distribution $\{D_1, D_2\}$ = {N, F}
1	0.9988	VS and S	$\{(D_1, 0.9957), (D_2, 0.0043)\}$
2	0.6587	S and VS	$\{(D_1, 0.4307), (D_2, 0.5693)\}$
3	0.9989	M and M	$\{(D_1, 0.3159), (D_2, 0.6841)\}$
4	0.9840	L and L	$\{(D_1, 0.007), (D_2, 0.993)\}$

TABLE IX
UPDATED RULE WEIGHTS AND BELIEF DEGREES UNDER NUMERICAL OUTPUT AND EKF

Rule number	Updated rule weight	*Level 1* AND Level 2	*OC* distribution $\{D_1, D_2\}$ = {N, F}
1	1	VS and S	$\{(D_1, 0.9946), (D_2, 0.0554)\}$
2	0.9988	S and VS	$\{(D_1, 0.001), (D_2, 0.0.999)\}$
3	0.2342	M and M	$\{(D_1, 0.429), (D_2, 0.571)\}$
4	0.1499	L and L	$\{(D_1, 0.0914), (D_2, 0.9086)\}$

terms of the referential values, the recursive algorithm (45) is used to update the initial BRB. The updated rule weights and belief degrees are given in Table VIII. Fig. 6 shows that the updated BRB can replicate the relationship among *Level 1*, *Level 2,* and OC much more closely than the initial one after the algorithm converged at about time 50 s.

Step 3—Comparative studies: Some other methods, such as extended Kalman filter (EKF) [42], interactive multiple model [43], multiple model adaptive estimation [44], and multiple model reference control [45], can be used for model learning. In order to validate the efficiency of the proposed recursive algorithm, we choose one of the aforementioned methods, EKF, to update the initial BRB. The updated rule weights and belief degrees are given in Table IX. In Fig. 6, it can be seen that EKF can update the initial BRB and converges at about time 280 s. Therefore, the learning speed of EKF is much slower than the proposed recursive algorithm.

Step 4—Quantitative analysis: In order to further demonstrate the accuracy of the proposed algorithm, the MAPE is also used. The MAPE between the observed OC values and the OC values generated by the initial BRB is 5.92%, and the MAPE between the observed OC values and the OC values generated by the EKF updated BRB is 4.86%. On the other hand, the MAPE between the observed OC values and the OC values generated by the recursively updated BRB is 1%. It is obvious that the recursively updated BRB can replicate the relationship among *Level 1*, *Level 2*, and OC much more accurately than the initial one and the EKF updated one.

6) Concluding Remarks: From the aforementioned numerical study, we have seen that the initial BRB given by an expert are not accurate, and the proposed recursive algorithms can be used to update the initial BRB whether the output of BRB is single numerical values or 2-D distributions. Moreover, if an appropriate initial BRB is given by an expert, the proposed algorithms can converge fairly fast.

Moreover, RIMER allows direct expert intervention, which differentiates it from other methods. The expert intervention can be used to determine the initial values of the unknown

TABLE X
REFERENTIAL POINTS OF *FlowDiff*

Linguistic terms	NL	NM	NS	NVS	Z	PS	PM	PL
Numerical values	-10	-5	-3	-1	0	1	2	3

TABLE XI
REFERENTIAL POINTS OF *PressureDiff*

Linguistic terms	NL	NM	NS	Z	PS	PM	PL
Numerical values	-0.042	-0.025	-0.01	0	0.01	0.025	0.042

parameters, which is very useful to improve the model learning speed.

B. Case Study

In order to demonstrate the potential application of the proposed recursive algorithms in engineering, we will apply the scheme to build an expert system for pipeline oil leak detection, with data taken from an operational long distance oil pipeline installed in Great Britain.

1) Problem Formulation: Similar to [25], the leak data includes the difference between inlet flow and outlet flow, the average pipeline pressure change over time and the leak rate, denoted by *FlowDiff*, *PressureDiff*, and *LeakSize*, respectively. *FlowDiff* and *PressureDiff* are the two very important factors in detecting whether there is leak in the pipeline, and they can be treated as the antecedent attributes of a rule base, and their calculation equations are given in [25]. Obviously, *LeakSize* is the consequent of the rule base.

We will use the data to update a BRB expert system for detecting leaks and estimate leak sizes without generating false alarms.

2) Referential Points of Antecedent Attributes and Consequent: The antecedent attributes and consequent in the rule base should be given some referential points. Similar to [25], we choose these points as follows:

For *FlowDiff*, eight referential points are used, and they are negative large (NL), negative medium (NM), negative small (NS), negative very small (NVS), zero (Z), positive small (PS), positive medium (PM), and positive large (PL), i.e.,

$$A_1^k \in \{NL, NM, NS, NVS, Z, PS, PM, PL\}. \quad (63)$$

For *PressureDiff*, seven referential points are used and they are NL, NM, NS, Z, PS, PM, PL, i.e.,

$$A_2^k \in \{NL, NM, NS, Z, PS, PM, PL\}. \quad (64)$$

For *LeakSize*, five referential points are used: Z, very small (VS), medium (M), high (H), and very high (VH), i.e.,

$$\mathbf{D} = (D_1, D_2, D_3, D_4, D_5) = (Z, VS, M, H, VH). \quad (65)$$

The referential points defined earlier for the antecedent attributes and consequent are in linguistic terms and need to be quantified. The quantified results are given in Tables X–XII, respectively.

TABLE XII
REFERENTIAL POINTS OF *LeakSize*

Linguistic terms	Z	VS	M	H	VH
Numerical values	0	2	4	6	8

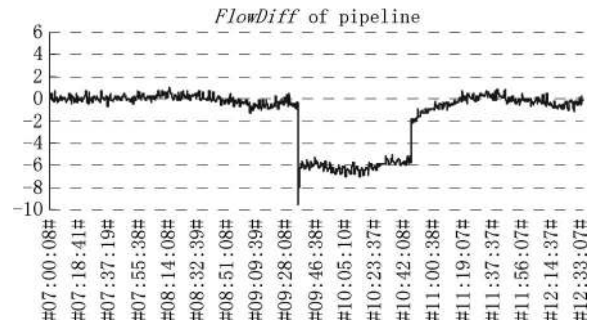


Fig. 7. *FlowDiff* of the pipeline.

3) Rules: In the BRB for the pipeline leak detection, a belief rule can be represented as

$$\begin{aligned} R_k:\ & \text{If } FlowDiff \text{ is } A_1^k \wedge PressureDiff \text{ is } A_2^k \\ & \text{Then } LeakSize \text{ is} \\ & \{(Z, \beta_{1,k}), (VS, \beta_{2,k}), (M, \beta_{3,k}), (H, \beta_{4,k}), (VH, \beta_{5,k})\} \\ & \text{With a rule weight } \theta_k \text{ and attribute weight } \delta_{1,k}, \delta_{2,k} \end{aligned} \quad (66)$$

where A_1^k and A_2^k $(k = 1, \ldots, 56)$ are the referential values as defined in Tables X and XI, respectively. Because *FlowDiff* and *PressureDiff* are divided into eight and seven terms, respectively, there are 56 combinations of the two antecedent attributes leading to 56 belief rules in total. The initial belief degrees of the BRB are given by an expert, as shown in Table XIII of Appendix D. However, the initially given belief degrees for *LeakSize* may not be accurate. It is necessary to update the belief degrees so that the performance of the expert system can be improved or optimized in a sense.

4) Updating and Testing of BRB: Similar to [25], during the leak trial, 2008 samples of 4, 16, and 25% leak data were collected at the rate of 10 s per sample, respectively. Figs. 7 and 8 give the *FlowDiff* and *PressureDiff*, respectively, when there is no leak and 25% leak. In order to update the BRB, 800 datasets are selected, which include 200 from no leak, 200 from 25% leak, 200 from 16% leak, and 200 from 4% leak. Then, these data are used to update the BRB using the proposed recursive algorithms. The process of updating and testing the BRB is implemented using MATLAB.

Step 1—Set initial parameters: The initial belief degrees are given by an expert and listed in Table XIII of Appendix D. θ_k and $\overline{\delta}_j$ are all assumed to be 1, where $k = 1, \ldots, 56$ and $j = 1, 2$. As shown in Fig. 9, it is obvious that the values of the estimated *LeakSize* calculated by the initial BRB do not match the observed values when the leak is 25%. This means that the initial BRB provided by an expert is not accurate. Hence, it is necessary to update the BRB online.

Step 2—Update the initial BRB: After the input values $[FlowDiff(n), PressureDiff(n)]$ are transformed and

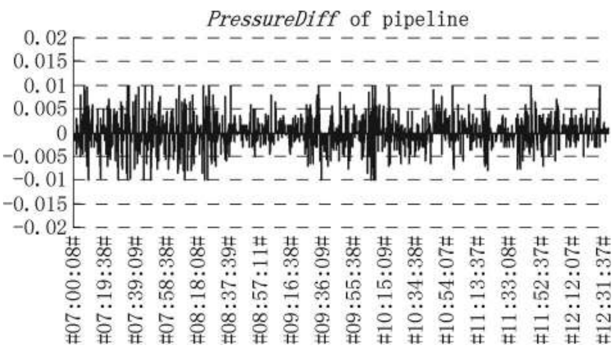


Fig. 8. *PressureDiff* of the pipeline.

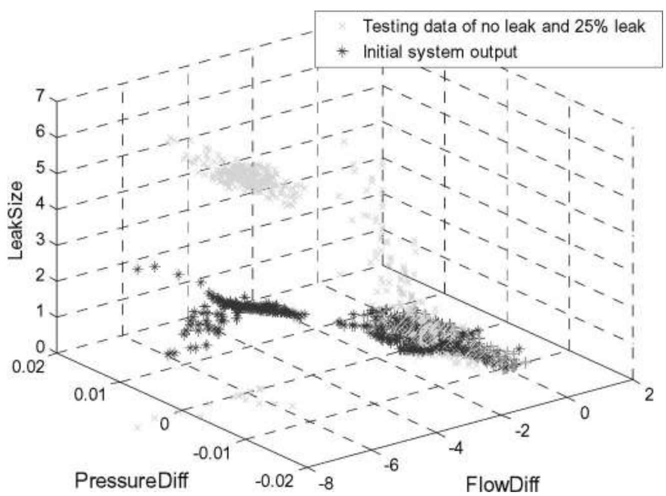


Fig. 9. Testing data of no leak and 25% leak and outputs generated by the initial BRB.

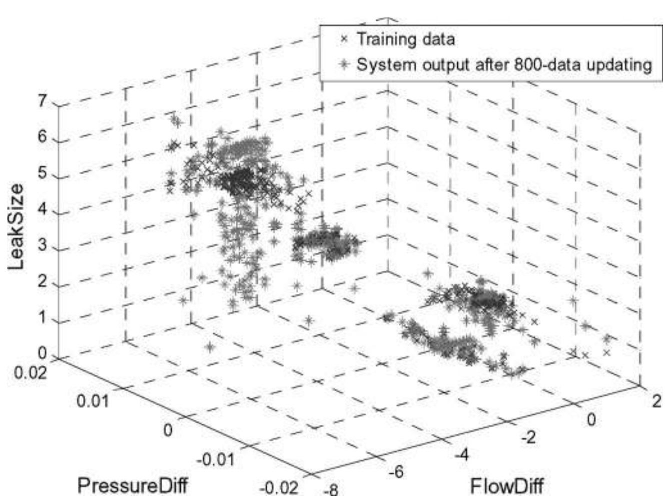


Fig. 10. Training data and outputs generated by the updated BRB.

represented in terms of the referential values as defined in (63) and (64) at time instant n, the recursive algorithm (45) is used to update the initial BRB. The updated rule weights and belief degrees are listed in Table XIV of Appendix D. Figs. 10 and 11 on the time scale show that the updated BRB can closely replicate the relationship among *FlowDiff*, *PressureDiff*, and *LeakSize* in the training data. Moreover, the calculation speed of the recursive algorithm is very high.

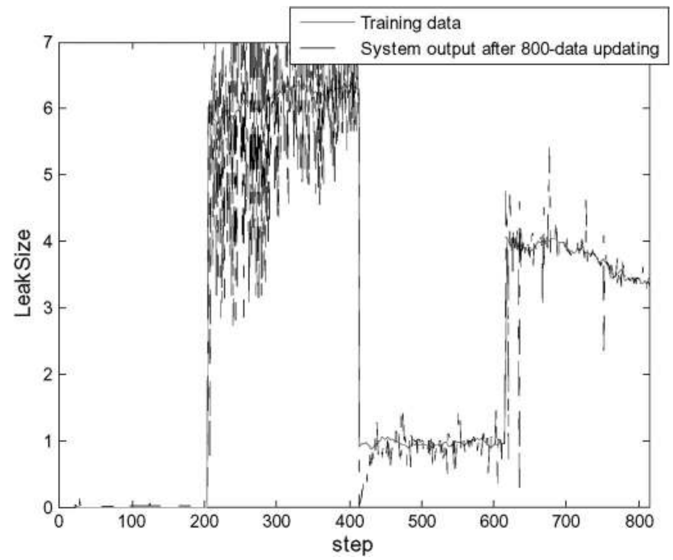


Fig. 11. Training data and outputs generated by the updated BRB on the time scale.

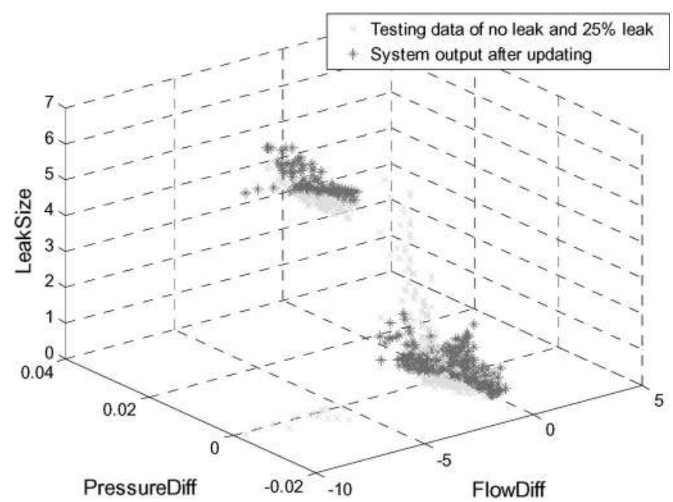


Fig. 12. Testing data of no leak and 25% leak and outputs generated by the updated BRB.

From Fig. 11, we can see that there is noise in the 25% leak detected, which may be due to noise in data recorded from instruments. Therefore, in a real leak detection system, some kind of noise reduction process to smooth data should be included.

Step 3—Test the updated BRB: For testing the updated BRB, all the 2008 data shown in Figs. 7 and 8 are used. Fig. 12 shows the observed *LeakSize* and the estimated *LeakSize* for the same antecedent values $[FlowDiff(n), PressureDiff(n)]$. It demonstrates that the estimated *LeakSize* matches the observed one very closely. Fig. 13 shows the observed and estimated *LeakSize* on the time scale. It shows that the rule base can detect the leak which happened at around 9:34 A.M. and ended at around 10:50 A.M.

5) Concluding Remarks: When 16 and 4% leak data have been used, the similar results are obtained using the updated BRB. In summary, the initial BRB for pipeline leak detection

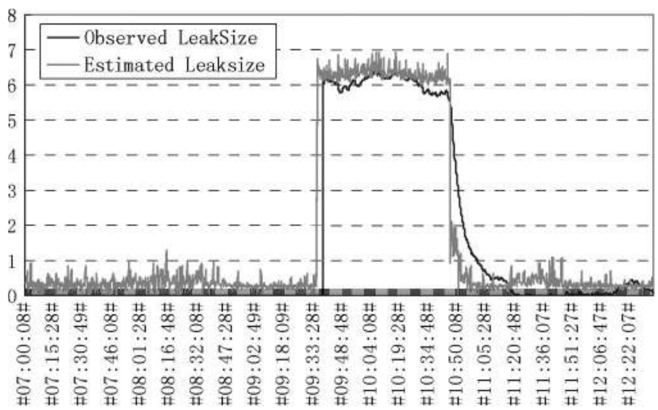


Fig. 13. Testing data of no leak and 25% leak and outputs generated by the updated BRB.

given by an expert are not accurate. When the new information becomes available, the proposed recursive algorithm can update the BRB quickly. Once the BRB is updated, it can be used to forecast future leak.

From this case study, it can also be concluded that the proposed algorithms can be widely applied in engineering.

V. Conclusion

This paper has been concerned with developing the recursive algorithms for extending the recently developed belief rule inference methodology (RIMER) for online updating BRB. The two proposed recursive algorithms provide an innovative way to enhance the capability of RIMER to simulate dynamic systems where both expert knowledge and partial input–output data are available. A numerical example and a case study for the pipeline oil leak detection are examined to demonstrate how the proposed algorithms can be implemented, which shows that the approach may be widely applied in engineering.

There are three features in the proposed approach. First, the proposed algorithms are recursive and analytic in nature. This ensures that different from the other optimization models used for training BRB, the proposed algorithms can be used to train a BRB to simulate a dynamic system without having to reply on the availability of complete training information. Second, the proposed algorithms can be used to process incomplete or vague information, which inherits from RIMER. Finally, since RIMER allows the direct expert intervention, the proposed algorithms can also take into account expert knowledge to determine the initial values of BRB structures and parameters, which is helpful to improve learning speed. The aforementioned features equip the proposed algorithms with the capability of online updating BRB models to simulate a range of real systems, especially when there is a requirement for real-time analysis and updating.

However, if there are too many belief rules in an initial rule base for a complex real-world problem, there will also be a lot of parameters to be updated, which will have impact on the calculation speed of the recursive algorithms, or may result in overfitting. On the other hand, if there are too few rules in an initial rule base, it may lead to underfitting. Moreover, there may be conflicting rules which are qualitatively incorrect. In fact, these problems may stem from the irrational structure of BRB constructed using limited or incorrect expert knowledge. In addition, the proposed recursive algorithms are locally optimal. Consequently, an updated BRB may not be able to achieve overall optimal performances. Therefore, there is a need to develop appropriate principles to check conflicting rules and adjust the structure of a BRB. Such principles may be transformed into constraints to develop new global optimization training models to achieve overall optimal performances. These requirements pose challenges for future research.

Appendix A
Calculation of the Derivatives

In this Appendix, the detailed processes to obtain the recursive algorithm as shown in (13) are given.

To obtain a proper approximation of $L_{n+1}(\mathbf{Q})$, we will consider the Taylor expansion of the first terms on the right-hand side of (10). Approximately

$$L_n(\mathbf{Q}) \approx L_n\left(\mathbf{Q}(n)\right) + \left[\nabla_{\mathbf{Q}} L_n\left(\mathbf{Q}(n)\right)\right]\left(\mathbf{Q} - \mathbf{Q}(n)\right) + \frac{1}{2}\left(\mathbf{Q} - \mathbf{Q}(n)\right)^T \left[\nabla_{\mathbf{Q}} \nabla_{\mathbf{Q}}^T L_n\left(\mathbf{Q}(n)\right)\right]\left(\mathbf{Q} - \mathbf{Q}(n)\right) \quad \text{(A.1)}$$

where $\nabla_{\mathbf{Q}}$ is a column gradient operator with respect to the parameter vector $\mathbf{Q}$.

By the definition of $L_n(\mathbf{Q})$, $\nabla_{\mathbf{Q}} \nabla_{\mathbf{Q}}^T L_n(\mathbf{Q}(n))$ is approximately given by [34], [35]

$$\nabla_{\mathbf{Q}} \nabla_{\mathbf{Q}}^T L_n\left(\mathbf{Q}(n)\right) \approx -(n-1)\Xi_1\left(\mathbf{Q}(n)\right) \quad \text{(A.2)}$$

where $\Xi_1(\mathbf{Q}(n))$ is the augmented information matrix calculated at $\mathbf{Q}(n)$ and

$$\Xi_1(\mathbf{Q}(n)) \triangleq E\left\{-\nabla_{\mathbf{Q}} \nabla_{\mathbf{Q}}^T \log f\left(\hat{\mathbf{B}}(n)|\hat{\mathbf{x}}(n), \mathbf{Q}\right)|\hat{\mathbf{x}}(n), \mathbf{Q}(n)\right\} \quad \text{(A.3)}$$

Because $\mathbf{Q} = \mathbf{Q}(n)$ is the maximum point of $L_n(\mathbf{Q})$ in (A.1), there is

$$\nabla_{\mathbf{Q}} L_n\left(\mathbf{Q}(n)\right) = 0. \quad \text{(A.4)}$$

Substituting (A.3) and (A.4) into (A.1), we can obtain

$$L_n(\mathbf{Q}) \approx L_n\left(\mathbf{Q}(n)\right) - \frac{1}{2}\left(\mathbf{Q} - \mathbf{Q}(n)\right)^T \times \left[(n-1)\Xi_1\left(\mathbf{Q}(n)\right)\right]\left(\mathbf{Q} - \mathbf{Q}(n)\right). \quad \text{(A.5)}$$

Then by Taylor expansion, approximately

$$\begin{aligned}\log f\left(\hat{\mathbf{B}}(n)|\hat{\mathbf{x}}(n), \mathbf{Q}\right) &\approx \log f\left(\hat{\mathbf{B}}(n)|\hat{\mathbf{x}}(n), \mathbf{Q}(n)\right) \\ &+ \left(\nabla_{\mathbf{Q}} \log f\left(\hat{\mathbf{B}}(n)|\hat{\mathbf{x}}(n), \mathbf{Q}(n)\right)\right)\left(\mathbf{Q} - \mathbf{Q}(n)\right) \\ &+ \frac{1}{2}\left(\mathbf{Q} - \mathbf{Q}(n)\right)^T \left(\nabla_{\mathbf{Q}} \nabla_{\mathbf{Q}}^T \log f\left(\hat{\mathbf{B}}(n)|\hat{\mathbf{x}}(n), \mathbf{Q}(n)\right)\right) \\ &\times \left(\mathbf{Q} - \mathbf{Q}(n)\right).\end{aligned} \quad \text{(A.6)}$$

Define

$$\Gamma_1(\mathbf{Q}(n)) \triangleq \nabla_{\mathbf{Q}} \log f\left(\hat{\mathbf{B}}(n)|\hat{\mathbf{x}}(n), \mathbf{Q}(n)\right) \tag{A.7}$$

where $\nabla_{\mathbf{Q}} \log f(\hat{\mathbf{B}}(n)|\hat{\mathbf{x}}(n), \mathbf{Q}(n))$ represents the gradient vector at $\mathbf{Q}(n)$.

Hence, the conditional expectation of (A.6) can be written as

$$\begin{aligned} &E\left\{\log f\left(\hat{\mathbf{B}}(n)|\hat{\mathbf{x}}(n), \mathbf{Q}\right)|\hat{\mathbf{x}}(n), \mathbf{Q}(n)\right\} \\ &\quad = E\left\{\log f\left(\hat{\mathbf{B}}(n)|\hat{\mathbf{x}}(n), \mathbf{Q}(n)\right)|\hat{\mathbf{x}}(n), \mathbf{Q}(n)\right\} \\ &\qquad + \Gamma_1(\mathbf{Q}(n))(\mathbf{Q}-\mathbf{Q}(n)) \\ &\qquad + \frac{1}{2}(\mathbf{Q}-\mathbf{Q}(n))^T E \\ &\qquad \times \left\{\nabla_{\mathbf{Q}}\nabla_{\mathbf{Q}}^T \log f\left(\hat{\mathbf{B}}(n)|\hat{\mathbf{x}}(n), \mathbf{Q}(n)\right)|\hat{\mathbf{x}}(n), \mathbf{Q}(n)\right\} \\ &\qquad \times (\mathbf{Q}-\mathbf{Q}(n)). \end{aligned} \tag{A.8}$$

According to (10), (A.5), and (A.8), the following expression can be obtained:

$$\begin{aligned} L_{n+1}(\mathbf{Q}) = &L_n(\mathbf{Q}(n)) \\ &+ E\left\{\log f\left(\hat{\mathbf{B}}(n)|\hat{\mathbf{x}}(n), \mathbf{Q}(n)\right)|\hat{\mathbf{x}}(n), \mathbf{Q}(n)\right\} \\ &+ \Gamma_1(\mathbf{Q}(n))(\mathbf{Q}-\mathbf{Q}(n)) - \frac{n}{2}(\mathbf{Q}-\mathbf{Q}(n))^T \\ &\times [\Xi_1(\mathbf{Q}(n))](\mathbf{Q}-\mathbf{Q}(n)). \end{aligned} \tag{A.9}$$

The first and second terms of (A.9) are constants, so the maximizing parameter $\mathbf{Q}(n+1)$ is given by

$$\mathbf{Q}(n+1) = \mathbf{Q}(n) + \frac{1}{n}[\Xi_1(\mathbf{Q}(n))]^{-1}\Gamma_1(\mathbf{Q}(n)). \tag{A.10}$$

Appendix B
Calculation of the Derivatives

In (21a) and (21b) and (35a) and (35b), some derivatives are used and are given in this Appendix.

When n is shaded, (18) is written as

$$\begin{aligned} f(\hat{\mathbf{B}}|\mathbf{x}, \mathbf{Q}) = &(2\pi)^{\frac{-N}{2}}|\Sigma|^{\frac{-1}{2}} \\ &\times \exp\left\{-\frac{1}{2}(\hat{\mathbf{B}}-\mathbf{B})^T\Sigma^{-1}(\hat{\mathbf{B}}-\mathbf{B})\right\} \end{aligned} \tag{B.1}$$

$$\begin{aligned} \log f(\hat{\mathbf{B}}|\mathbf{x}, \mathbf{Q}) = &-\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\Sigma| \\ &-\frac{1}{2}(\hat{\mathbf{B}}-\mathbf{B})^T\Sigma^{-1}(\hat{\mathbf{B}}-\mathbf{B}) \end{aligned} \tag{B.2}$$

where $\sum = \sigma \times \mathrm{diag}\{\underbrace{1,\ldots,1}_{N}\}$; $\hat{\mathbf{B}} = [\hat{\beta}_1, \ldots, \hat{\beta}_N]^T$; and $\mathbf{B} = [\beta_1, \ldots, \beta_N]^T$.

The first derivatives of $\log f(\hat{\mathbf{B}}|\mathbf{x}, \mathbf{Q})$ with respect to V_a $(a = 1, \ldots, L+M+L\times N)$ are

$$\frac{\partial \log f(\hat{\mathbf{B}}|\mathbf{x}, \mathbf{Q})}{\partial V_a} = \frac{\partial \mathbf{B}^T}{\partial V_a}\sum^{-1}(\hat{\mathbf{B}}-\mathbf{B}) \tag{B.3}$$

where $\partial\beta_j/\partial V_a$ needs to be calculated.

Assume

$$\begin{aligned} B_j = &\prod_{k=1}^{L}\left(\omega_k\beta_{j,k} + 1 - \omega_k\sum_{i=1}^{N}\beta_{i,k}\right) \\ &- \prod_{k=1}^{L}\left(1 - \omega_k\sum_{i=1}^{N}\beta_{i,k}\right) \end{aligned} \tag{B.4}$$

$$\begin{aligned} C = &\sum_{j=1}^{N}\prod_{k=1}^{L}\left(\omega_k\beta_{j,k} + 1 - \omega_k\sum_{i=1}^{N}\beta_{i,k}\right) \\ &- (N-1)\prod_{k=1}^{L}\left(1 - \omega_k\sum_{i=1}^{N}\beta_{i,k}\right) - \prod_{k=1}^{L}(1-\omega_k) \end{aligned} \tag{B.5}$$

Then, there is

$$\beta_j = \frac{B_j}{C} \tag{B.6}$$

Define

$$\phi(q) \triangleq \prod_{i=1}^{M}(\alpha_i^q)^{\overline{\delta}_i} \tag{B.7}$$

$$\varphi \triangleq \sum_{l=1}^{L}\theta_l\prod_{i=1}^{M}(\alpha_i^l)^{\overline{\delta}_i} \tag{B.8}$$

$$\xi_1(q) \triangleq \prod_{\substack{k=1\\k\neq q}}^{L}\left(1 - \omega_k\sum_{i=1}^{N}\beta_{i,k}\right) \tag{B.9}$$

$$\chi_1(q,j) \triangleq \prod_{\substack{k=1\\k\neq q}}^{L}\left(\omega_k\beta_{j,k} + 1 - \omega_k\sum_{i=1}^{N}\beta_{i,k}\right). \tag{B.10}$$

The first order derivatives of $\beta_j (j = 1, \ldots, N)$ with respect to $V_a (a = 1, \ldots, L+M+L\times N)$ are represented as

$$\frac{\partial\beta_j}{\partial V_a} = \frac{1}{C^2}\left(\frac{\partial B_j}{\partial V_a}C - \frac{\partial C}{\partial V_a}B_j\right) \tag{B.11}$$

with

$$\frac{\partial B_j}{\partial\theta_s} = \sum_{q=1}^{L}\frac{\partial\omega_q}{\partial\theta_s}\frac{\partial B_j}{\partial\omega_q} \tag{B.12}$$

$$\frac{\partial B_j}{\partial\overline{\delta}_m} = \sum_{q=1}^{L}\frac{\partial\omega_q}{\partial\overline{\delta}_m}\frac{\partial B_j}{\partial\omega_q} \tag{B.13}$$

$$\frac{\partial C}{\partial\theta_s} = \sum_{q=1}^{L}\frac{\partial\omega_q}{\partial\theta_s}\frac{\partial C}{\partial\omega_q} \tag{B.14}$$

$$\frac{\partial C}{\partial\overline{\delta}_m} = \sum_{q=1}^{L}\frac{\partial\omega_q}{\partial\overline{\delta}_m}\frac{\partial C}{\partial\omega_q} \tag{B.15}$$

$$\frac{\partial B_j}{\partial\beta_{z,p}} = \begin{cases} -\omega_p\chi_1(p,j) + \omega_p\xi_1(p), & j\neq z \\ \omega_p\xi_1(p), & j = z \end{cases} \tag{B.16}$$

$$\frac{\partial C}{\partial\beta_{z,p}} = -\sum_{\substack{j=1\\j\neq z}}^{N}\omega_p\chi_1(p,j) + (N-1)\omega_p\xi_1(p) \tag{B.17}$$

TABLE XIII
INITIAL BELIEF DEGREES FOR PIPELINE OIL LEAK DETECTION PROVIDED BY AN EXPERT

Rule number	*FlowDiff* AND *PressureDiff*	*LeakSize* distribution $\{D_1, D_2, D_3, D_4, D_5\}$={0, 2, 4, 6, 8}
1	NL AND NL	{(D_1, 0), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 1)}
2	NL AND NM	{(D_1, 0), (D_2, 0), (D_3, 0), (D_4, 0.3), (D_5, 0.7)}
3	NL AND NS	{(D_1, 0), (D_2, 0), (D_3, 0.2), (D_4, 0.8), (D_5, 0)}
4	NL AND Z	{(D_1, 0), (D_2, 0), (D_3, 0.8), (D_4, 0.2), (D_5, 0)}
5	NL AND PS	{(D_1, 0.65), (D_2, 0.35), (D_3, 0), (D_4, 0), (D_5, 0)}
6	NL AND PM	{(D_1, 0.85), (D_2, 0.15), (D_3, 0), (D_4, 0), (D_5, 0)}
7	NL AND PL	{(D_1, 0.95), (D_2, 0.05), (D_3, 0), (D_4, 0), (D_5, 0)}
8	NM AND NL	{(D_1, 0), (D_2, 0), (D_3, 0.1), (D_4, 0.9), (D_5, 0)}
9	NM AND NM	{(D_1, 0), (D_2, 0), (D_3, 0.7), (D_4, 0.3), (D_5, 0)}
10	NM AND NS	{(D_1, 0), (D_2, 0.7), (D_3, 0.3), (D_4, 0), (D_5, 0)}
11	NM AND Z	{(D_1, 0), (D_2, 0.9), (D_3, 0.1), (D_4, 0), (D_5, 0)}
12	NM AND PS	{(D_1, 0.8), (D_2, 0.2), (D_3, 0), (D_4, 0), (D_5, 0)}
13	NM AND PM	{(D_1, 0.9), (D_2, 0.1), (D_3, 0), (D_4, 0), (D_5, 0)}
14	NM AND PL	{(D_1, 0.99), (D_2, 0.01), (D_3, 0), (D_4, 0), (D_5, 0)}
15	NS AND NL	{(D_1, 0), (D_2, 0), (D_3, 0.4), (D_4, 0.6), (D_5, 0)}
16	NS AND NM	{(D_1, 0), (D_2, 0), (D_3, 0.8), (D_4, 0.2), (D_5, 0)}
17	NS AND NS	{(D_1, 0), (D_2, 0.3), (D_3, 0.6), (D_4, 0.1), (D_5, 0)}
18	NS AND Z	{(D_1, 0.1), (D_2, 0.7), (D_3, 0.2), (D_4, 0), (D_5, 0)}
19	NS AND PS	{(D_1, 0.7), (D_2, 0.3), (D_3, 0), (D_4, 0), (D_5, 1)}
20	NS AND PM	{(D_1, 0.9), (D_2, 0.1), (D_3, 0), (D_4, 0), (D_5, 1)}
21	NS AND PL	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
22	NVS AND NL	{(D_1, 0.02), (D_2, 0.11), (D_3, 0.39), (D_4, 0.48), (D_5, 0)}
23	NVS AND NM	{(D_1, 0.1), (D_2, 0.78), (D_3, 0.12), (D_4, 0), (D_5, 0)}
24	NVS AND NS	{(D_1, 0.36), (D_2, 0.64), (D_3, 0), (D_4, 0), (D_5, 0)}
25	NVS AND Z	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
26	NVS AND PS	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
27	NVS AND PM	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
28	NVS AND PL	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
29	Z AND NL	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
30	Z AND NM	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
31	Z AND NS	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
32	Z AND Z	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
33	Z AND PS	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
34	Z AND PM	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
35	Z AND PL	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
36	PS AND NL	{(D_1, 0.39), (D_2, 0.61), (D_3, 0), (D_4, 0), (D_5, 0)}
37	PS AND NM	{(D_1, 0.9), (D_2, 0.1), (D_3, 0), (D_4, 0), (D_5, 0)}
38	PS AND NS	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
39	PS AND Z	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
40	PS AND PS	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
41	PS AND PM	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
42	PS AND PL	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
43	PM AND NL	{(D_1, 0.1), (D_2, 0.9), (D_3, 0), (D_4, 0), (D_5, 0)}
44	PM AND NM	{(D_1, 0.3), (D_2, 0.7), (D_3, 0), (D_4, 0), (D_5, 0)}
45	PM AND NS	{(D_1, 0.85), (D_2, 0.15), (D_3, 0), (D_4, 0), (D_5, 0)}
46	PM AND Z	{(D_1, 0.98), (D_2, 0.02), (D_3, 0), (D_4, 0), (D_5, 0)}
47	PM AND PS	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
48	PM AND PM	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
49	PM AND PL	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
50	PL AND NL	{(D_1, 0.9), (D_2, 0.1), (D_3, 0), (D_4, 0), (D_5, 0)}
51	PL AND NM	{(D_1, 0.99), (D_2, 0.01), (D_3, 0), (D_4, 0), (D_5, 0)}
52	PL AND NS	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
53	PL AND Z	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
54	PL AND PS	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
55	PL AND PM	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
56	PL AND PL	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}

where

$$\frac{\partial \omega_q}{\partial \theta_s} = \begin{cases} -\theta_q \phi(q)\phi(s)\varphi^{-2}, & s \neq q \\ \phi(s)\varphi^{-1} - \theta_s \phi(s)^2 \varphi^{-2}, & s = q \end{cases} \tag{B.18}$$

$$\frac{\partial \omega_q}{\partial \bar{\delta}_m} = \theta_q \left(\ln \alpha_m^q\right) \phi(q)\varphi^{-1}$$

$$- \theta_q \phi(q)\varphi^{-2} \sum_{l=1}^{L} \theta_l \left(\ln \alpha_m^l\right) \phi(l) \tag{B.19}$$

$$\frac{\partial B_j}{\partial \omega_q} = \left(\beta_{j,q} - \sum_{i=1}^{N} \beta_{i,q}\right) \chi_1(q, j)$$

TABLE XIV
UPDATED RULE WEIGHTS AND BELIEF DEGREES

Rule number	Updated rule weight	*FlowDiff* AND *PressureDiff*	*LeakSize* distribution {D_1, D_2, D_3, D_4, D_5}={0, 2, 4, 6, 8}
1	1	NL AND NL	{(D_1, 0), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 1)}
2	1	NL AND NM	{(D_1, 0), (D_2, 0), (D_3, 0), (D_4, 0.3), (D_5, 0.7)}
3	1	NL AND NS	{(D_1, 0.001), (D_2, 0.001), (D_3, 0.0473), (D_4, 0.4727), (D_5, 0.4780)}
4	0.7669	NL AND Z	{(D_1, 0.0061), (D_2, 0.0075), (D_3, 0.0106), (D_4, 0.5417), (D_5, 0.4341)}
5	0.8852	NL AND PS	{(D_1, 0.0079), (D_2, 0.0092), (D_3, 0.001), (D_4, 0.4399), (D_5, 0.542)}
6	1	NL AND PM	{(D_1, 0.85), (D_2, 0.15), (D_3, 0), (D_4, 0), (D_5, 0)}
7	1	NL AND PL	{(D_1, 0.95), (D_2, 0.05), (D_3, 0), (D_4, 0), (D_5, 0)}
8	0.9616	NM AND NL	{(D_1, 0.1619), (D_2, 0.0771), (D_3, 0.1005), (D_4, 0.6595), (D_5, 0.0009)}
9	1	NM AND NM	{(D_1, 0.1641), (D_2, 0.0865), (D_3, 0..5916), (D_4, 0.1566), (D_5, 0.0011)}
10	0.8619	NM AND NS	{(D_1, 0.0707), (D_2, 0.0392), (D_3, 0.0303), (D_4, 0.3982), (D_5, 0.4615)}
11	0.562	NM AND Z	{(D_1, 0.0678), (D_2, 0.0786), (D_3, 0.0131), (D_4, 0.4946), (D_5, 0.346)}
12	0.9196	NM AND PS	{(D_1, 0.0142), (D_2, 0.0048), (D_3, 0.0079), (D_4, 0.486), (D_5, 0.4871)}
13	0.955	NM AND PM	{(D_1, 0.4595), (D_2, 0.0128), (D_3, 0.0014), (D_4, 0.179), (D_5, 0.3473)}
14	0.9981	NM AND PL	{(D_1, 0.3508), (D_2, 0.001), (D_3, 0.0022), (D_4, 0.2097), (D_5, 0.4363)}
15	1	NS AND NL	{(D_1, 0.2221), (D_2, 1111), (D_3, 0.3248), (D_4, 0.3411), (D_5, 0.001)}
16	0.9999	NS AND NM	{(D_1, 0.1458), (D_2, 0.0762), (D_3, 0.69), (D_4, 0.0853), (D_5, 0.0026)}
17	1	NS AND NS	{(D_1, 0.0554), (D_2, 0.3101), (D_3, 0.5708), (D_4, 0.0623), (D_5, 0.0015)}
18	0.9978	NS AND Z	{(D_1, 0.0479), (D_2, 0.5779), (D_3, 0.1816), (D_4, 0.074), (D_5, 0.1186)}
19	0.9939	NS AND PS	{(D_1, 0.5544), (D_2, 0.2444), (D_3, 0.0008), (D_4, 0.0688), (D_5, 0.1316)}
20	0.9521	NS AND PM	{(D_1, 0.4925), (D_2, 0.0053), (D_3, 0.0009), (D_4, 0.1703), (D_5, 0.3311)}
21	0.9597	NS AND PL	{(D_1, 0.5472), (D_2, 0.001), (D_3, 0.001), (D_4, 0.1434), (D_5, 0.3075)}
22	0.9946	NVS AND NL	{(D_1, 0.0406), (D_2, 0.1176), (D_3, 0.3825), (D_4, 0.4584), (D_5, 0.001)}
23	1	NVS AND NM	{(D_1, 0.1015), (D_2, 0.7771), (D_3, 0.1194), (D_4, 0.001), (D_5, 0.001)}
24	1	NVS AND NS	{(D_1, 0.3590), (D_2, 0.6381), (D_3, 0.001), (D_4, 0.0009), (D_5, 0.0009)}
25	0.9674	NVS AND Z	{(D_1, 0.9496), (D_2, 0.0007), (D_3, 0), (D_4, 0.001), (D_5, 0.001)}
26	1	NVS AND PS	{(D_1, 0.998), (D_2, 0), (D_3, 0), (D_4, 0.001), (D_5, 0.001)}
27	1	NVS AND PM	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
28	1	NVS AND PL	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
29	1	Z AND NL	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
30	1	Z AND NM	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
31	0.9996	Z AND NS	{(D_1, 0.9948), (D_2, 0.0006), (D_3, 0.001), (D_4, 0.0015), (D_5, 0.002)}
32	0.9899	Z AND Z	{(D_1, 0.9475), (D_2, 0.0011), (D_3, 0.0002), (D_4, 0.0161), (D_5, 0.0351)}
33	0.9994	Z AND PS	{(D_1, 0.9931), (D_2, 0.0001), (D_3, 0.001), (D_4, 0.0022), (D_5, 0.0036)}
34	1	Z AND PM	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
35	1	Z AND PL	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
36	1	PS AND NL	{(D_1, 0.39), (D_2, 0.61), (D_3, 0), (D_4, 0), (D_5, 0)}
37	1	PS AND NM	{(D_1, 0.9), (D_2, 0.1), (D_3, 0), (D_4, 0), (D_5, 0)}
38	0.9962	PS AND NS	{(D_1, 0.9681), (D_2, 0.001), (D_3, 0.0008), (D_4, 0.0101), (D_5, 0.0201)}
39	0.9999	PS AND Z	{(D_1, 0.792), (D_2, 0.0089), (D_3, 0.0024), (D_4, 0.0659), (D_5, 0.1308)}
40	0.9933	PS AND PS	{(D_1, 0.9532), (D_2, 0.0005), (D_3, 0.0006), (D_4, 0.015), (D_5, 0.0307)}
41	1	PS AND PM	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
42	1	PS AND PL	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
43	1	PM AND NL	{(D_1, 0.1), (D_2, 0.9), (D_3, 0), (D_4, 0), (D_5, 0)}
44	1	PM AND NM	{(D_1, 0.3), (D_2, 0.7), (D_3, 0), (D_4, 0), (D_5, 0)}
45	0.9985	PM AND NS	{(D_1, 0.8354), (D_2, 0.144), (D_3, 0.0009), (D_4, 0.0064), (D_5, 0.0133)}
46	1	PM AND Z	{(D_1, 0.8526), (D_2, 0.0027), (D_3, 0.0009), (D_4, 0.0475), (D_5, 0.0962)}
47	0.9955	PM AND PS	{(D_1, 0.9723), (D_2, 0.0002), (D_3, 0.0008), (D_4, 0.0086), (D_5, 0.0182)}
48	1	PM AND PM	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
49	1	PM AND PL	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
50	1	PL AND NL	{(D_1, 0.9), (D_2, 0.1), (D_3, 0), (D_4, 0), (D_5, 0)}
51	1	PL AND NM	{(D_1, 0.99), (D_2, 0.01), (D_3, 0), (D_4, 0), (D_5, 0)}
52	1	PL AND NS	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
53	1	PL AND Z	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
54	1	PL AND PS	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
55	1	PL AND PM	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}
56	1	PL AND PL	{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)}

$$+\ \xi_1(q)\sum_{i=1}^{N}\beta_{i,q} \tag{B.20}$$

$$\frac{\partial C}{\partial \omega_q} = \sum_{j=1}^{N}\left(\beta_{j,q} - \sum_{i=1}^{N}\beta_{i,q}\right)\chi_1(q,j)$$

$$+\,(N-1)\xi_1(q)\sum_{i=1}^{N}\beta_{i,q} + \prod_{\substack{k=1\\k\neq q}}^{L}(1-\omega_k) \tag{B.21}$$

where $j = 1,\ldots,N$; $s = 1,\ldots,L$; $p = 1,\ldots,L$; $z = 1,\ldots,N$; $q = 1,\ldots,L$; and $m = 1,\ldots,M$. ■

APPENDIX C
ALGORITHM TO ESTIMATE THE PARAMETERS OF THE COVARIANCE MATRIX

According to (22), the determinant of $\sum$ can be calculated by

$$|\Sigma| = \sigma_1^N + \sum_{k=2}^{N}(-1)^{k-1}(k-1)C_N^k\sigma_1^{N-k}\sigma_2^k \tag{C.1}$$

The inverse matrix $\sum^{-1} = (b_{i,j})_{N\times N}$ can be determined as follows:

$$\begin{cases} b_{i,j} = \eta_1, & i = j \\ b_{i,j} = \eta_2, & i \neq j \end{cases} \tag{C.2}$$

where

$$\eta_1 = \frac{\sigma_1 + (N-2)\sigma_2}{\sigma_1^2 + (N-2)\sigma_1\sigma_2 - (N-1)\sigma_2^2} \tag{C.3}$$

$$\eta_2 = \frac{-\sigma_2}{\sigma_1^2 + (N-2)\sigma_1\sigma_2 - (N-1)\sigma_2^2} \tag{C.4}$$

According to (18), we can obtain

$$\frac{\partial \log f\left(\hat{\mathbf{B}}(n)|\mathbf{x}(n),\mathbf{V}(n)\right)}{\partial\sigma_1} = -\frac{1}{2}\frac{\partial\log|\Sigma|}{\partial\sigma_1} - \frac{1}{2}\left[\frac{\partial\eta_1}{\partial\sigma_1}\gamma_1(n) + 2\frac{\partial\eta_2}{\partial\sigma_1}\gamma_2(n)\right] \tag{C.5}$$

$$\frac{\partial \log f\left(\hat{\mathbf{B}}(n)|\mathbf{x}(n),\mathbf{V}(n)\right)}{\partial\sigma_2} = -\frac{1}{2}\frac{\partial\log|\Sigma|}{\partial\sigma_2} - \frac{1}{2}\left[\frac{\partial\eta_1}{\partial\sigma_2}\gamma_1(n) + 2\frac{\partial\eta_2}{\partial\sigma_2}\gamma_2(n)\right] \tag{C.6}$$

where

$$\frac{\partial\log|\Sigma|}{\partial\sigma_1} = \frac{1}{|\Sigma|}\left[N\sigma_1^{N-1} + \sum_{k=2}^{N}(-1)^{k-1}(k-1)(N-k) \times C_N^k\sigma_1^{N-k-1}\sigma_2^k\right] \tag{C.7}$$

$$\frac{\partial\log|\Sigma|}{\partial\sigma_2} = \frac{1}{|\Sigma|}\left[\sum_{k=2}^{N}(-1)^{k-1}k(k-1)C_N^k\sigma_1^{N-k}\sigma_2^{k-1}\right] \tag{C.8}$$

$$\gamma_1(n) = \sum_{i=1}^{N}\left(\hat{B}_i(n) - B_i(n)\right)^2 \tag{C.9}$$

$$\gamma_2(n) = \sum_{\substack{i=1,j=1\\ i\neq j}}^{N}\left(\hat{B}_i(n) - B_i(n)\right)\left(\hat{B}_j(n) - B_j(n)\right) \tag{C.10}$$

$$\frac{\partial\eta_1}{\partial\sigma_1} = \frac{-\sigma_1^2 - 2(N-2)\sigma_1\sigma_2 - (N-1)\sigma_2^2 - (N-2)^2\sigma_2^2}{[\sigma_1^2 + (N-2)\sigma_1\sigma_2 - (N-1)\sigma_2^2]^2} \tag{C.11}$$

$$\frac{\partial\eta_1}{\partial\sigma_2} = \frac{(N-1)(N-2)\sigma_2^2 + 2(N-1)\sigma_1\sigma_2}{[\sigma_1^2 + (N-2)\sigma_1\sigma_2 - (N-1)\sigma_2^2]^2} \tag{C.12}$$

$$\frac{\partial\eta_2}{\partial\sigma_1} = \frac{2\sigma_1\sigma_2 + (N-2)\sigma_2^2}{[\sigma_1^2 + (N-2)\sigma_1\sigma_2 - (N-1)\sigma_2^2]^2} \tag{C.13}$$

$$\frac{\partial\eta_2}{\partial\sigma_2} = \frac{-\sigma_1^2 - (N-1)\sigma_2^2}{[\sigma_1^2 + (N-2)\sigma_1\sigma_2 - (N-1)\sigma_2^2]^2}. \tag{C.14}$$

Thus, $\sigma_i(n) = \arg\max_{\sigma_i} \log f(\hat{\mathbf{B}}(n)|\mathbf{x}(n),\mathbf{V}(n))$ is equivalent to solving the following nonlinear equations:

$$\begin{cases} \frac{\partial \log f\left(\hat{\mathbf{B}}(n)|\mathbf{x}(n),\mathbf{V}(n)\right)}{\partial\sigma_1} = 0 \\ \frac{\partial \log f\left(\hat{\mathbf{B}}(n)|\mathbf{x}(n),\mathbf{V}(n)\right)}{\partial\sigma_2} = 0. \end{cases} \tag{C.15}$$

The numerical method, such as FSOLVE function in MATLAB, can be used to solve the aforementioned nonlinear equations to estimate σ_1 and σ_2.

APPENDIX D

See Tables XIII and XIV.

ACKNOWLEDGMENT

The authors would like to thank the two anonymous referees and the Associate Editor for their constructive comments, which helped to improve the quality of this paper.

REFERENCES

[1] H. Kay, B. Rineer, and B. Kuiper, “Semi-quantitative system identification,” *Artif. Intell.*, vol. 119, no. 1/2, pp. 103–140, May 2000.
[2] I. B. Ozyurt, L. O. Hall, and A. K. Sunol, “SQFDiag: Semi-quantitative model-based fault monitoring and diagnosis via episodic fuzzy rules,” *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 29, no. 3, pp. 294–306, May 1999.
[3] P. J. Brockwell, *Time Series: Theory and Methods,* 2nd ed. New York: Springer-Verlag, 1991.
[4] S. J. Julier and J. K. Uhlmann, “Unscented filter and nonlinear estimation,” *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.
[5] T. Li and L. Guo, “Optimal fault-detection filtering for non-Gaussian systems via output PDFs,” *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 2, pp. 476–481, Mar. 2009.
[6] J. B. Yang, J. Liu, D. L. Xu, J. Wang, and H. W. Wang, “Optimal learning method for training belief rule based systems,” *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 37, no. 4, pp. 569–585, Jul. 2007.
[7] R. Sun, “Robust reasoning: Integrating rule-based and similarity-based reasoning,” *Artif. Intell.*, vol. 75, no. 2, pp. 241–295, Jun. 1995.
[8] Y. Xu, J. Liu, D. Ruan, and W. J. Li, “Fuzzy reasoning based on generalized fuzzy If-Then rules,” *Int. J. Intell. Sys.*, vol. 17, no. 10, pp. 977–1006, Oct. 2002.
[9] W. T. Chen and M. Saif, “A novel fuzzy system with dynamic rule base,” *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 5, pp. 569–582, Oct. 2005.
[10] H. K. Kwan and Y. L. Cai, “A fuzzy neural-network and its application to pattern recognition,” *IEEE Trans. Fuzzy Syst.*, vol. 2, no. 3, pp. 185–193, Aug. 1994.
[11] A. Er and R. Dias, “A rule-based expert system approach to process selection for cast components,” *Knowl.-Based Syst.*, vol. 13, no. 4, pp. 225–234, Jun. 2000.
[12] H. Monsef, A. M. Ranjbar, and S. Jadid, “Fuzzy rule-based expert system for power system fault diagnosis,” *Proc. Inst. Elect. Eng.—Gener. Transm. Distrib.*, vol. 144, no. 2, pp. 186–192, Mar. 1997.
[13] P. Walley, “Measures of uncertainty in expert system,” *Artif. Intell.*, vol. 83, no. 1, pp. 1–58, May 1996.

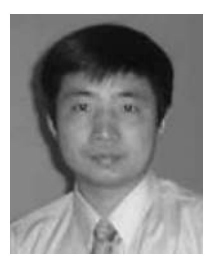
[14] Y. H. Qian, J. Y. Liang, and C. Y. Dang, "Incomplete multigranulation rough set," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 2, pp. 420–431, Mar. 2010.
[15] J. B. Yang, "Rule and utility based evidential reasoning approach for multi-attribute decision analysis under uncertainties," *Eur. J. Oper. Res.*, vol. 131, no. 1, pp. 31–61, May 2001.
[16] J. B. Yang, J. Liu, J. Wang, H. S. Sii, and H. W. Wang, "Belief rule-base inference methodology using the evidential reasoning approach—RIMER," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 2, pp. 266–285, Mar. 2006.
[17] J. B. Yang and P. Sen, "A general multi-level evaluation process for hybrid MADM with uncertainty," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 24, no. 10, pp. 1458–1473, Oct. 1994.
[18] J. B. Yang and M. G. Singh, "An evidential reasoning approach for multiple attribute decision making with uncertainty," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 24, no. 1, pp. 1–18, Jan. 1994.
[19] J. B. Yang and D. L. Xu, "On the evidential reasoning algorithm for multiple attribute decision analysis under uncertainty," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 32, no. 3, pp. 289–304, May 2002.
[20] J. B. Yang and D. L. Xu, "Nonlinear information aggregation via evidential reasoning in multi-attribute decision analysis under uncertainty," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 32, no. 3, pp. 376–393, May 2002.
[21] J. B. Yang, Y. M. Wang, D. L. Xu, and K. S. Chin, "The evidential reasoning approach for MADA under both probabilistic and fuzzy uncertainties," *Eur. J. Oper. Res.*, vol. 171, no. 1, pp. 309–343, May 2006.
[22] J. B. Yang and D. L. Xu, Intelligent Decision System Via Evidential Reasoning, ver. 1.1, IDSL, Cheshire, U.K., 1999.
[23] J. Liu, J. B. Yang, H. S. Sii, and Y. M. Wang, "Fuzzy rule-based evidential reasoning approach for safety analysis," *Int. J. Gen. Syst.*, vol. 33, no. 2/3, pp. 183–204, Apr.–Jun. 2004.
[24] J. Liu, J. B. Yang, and H. S. Sii, "Engineering system safety analysis and synthesis using fuzzy rule based evidential reasoning approach," *Qual. Reliab. Eng. Int.*, vol. 21, no. 4, pp. 387–411, Jun. 2005.
[25] D. L. Xu, J. Liu, J. B. Yang, G. P. Liu, J. Wang, I. Jenkinson, and J. Ren, "Inference and learning methodology of belief-rule-based expert system for pipeline leak detection," *Exp. Syst. Appl.*, vol. 32, no. 1, pp. 103–113, Jan. 2007.
[26] A. P. Dempster, "Upper and lower probabilities induced by a multi-valued mapping," *Ann. Math. Stat.*, vol. 38, no. 2, pp. 325–339, Apr. 1967.
[27] A. P. Dempster, "A generalization of Bayesian inference," *J. R. Stat. Soc. Ser. B*, vol. 30, no. 2, pp. 205–247, 1968.
[28] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, NJ: Princeton Univ. Press, 1976.
[29] J. Pearl, *Probabilistic Reasoning in Intelligence Systems*. San Mateo, CA: Morgan Kaufmann, 1988.
[30] D. J. Spiegelhalter, A. P. Dawld, S. L. Lauritzen, and R. G. Cowell, "Bayesian analysis in expert systems," *Stat. Sci.*, vol. 8, no. 3, pp. 219–247, 1993.
[31] C. Simon, P. Weber, and A. Evsukoff, "Bayesian networks inference algorithm to implement Dempster Shafer theory in reliability analysis," *Reliab. Eng. Syst. Safe.*, vol. 93, no. 7, pp. 950–963, Jul. 2008.
[32] J. Y. Halpern and R. Fagin, "Two views of belief: Belief as generalized probability and belief as evidence," *Artif. Intell.*, vol. 54, no. 3, pp. 275–317, Apr. 1992.
[33] A. P. Dempster, N. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc. Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.
[34] D. M. Titterington, "Recursive parameter estimation using incomplete data," *J. R. Stat. Soc. Ser. B*, vol. 46, no. 2, pp. 257–267, 1984.
[35] P. J. Chung and J. F. Bohme, "Recursive EM and SAGE-inspired algorithms with application to DOA estimation," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2664–2677, Aug. 2005.
[36] H. J. Kushner and M. L. Kelmanson, "Stochastic approximation algorithms of the multiplier type for the sequential Monte Carlo optimization of stochastic systems," *SIAM J. Control Optim.*, vol. 14, no. 5, pp. 827–842, 1976.
[37] H. J. Kushner and S. Lakshmivarahan, "Numerical studies of stochastic approximation procedures for constrained problems," *IEEE Trans. Autom. Control*, vol. AC-22, no. 3, pp. 428–439, Jun. 1977.
[38] Y. M. Wang, J. B. Yang, and D. L. Xu, "Environmental impact assessment using the evidential reasoning approach," *Eur. J. Oper. Res.*, vol. 174, no. 3, pp. 1885–1913, Nov. 2006.
[39] L. Ljung, "Analysis of recursive stochastic algorithm," *IEEE Trans. Autom. Control*, vol. AC-22, no. 4, pp. 551–575, Aug. 1977.
[40] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*. New York: Springer-Verlag, 1997.
[41] C. D. Lewis, *Industrial and Business Forecasting Method*. London, U.K.: Butterworth, 1982.
[42] R. Konrad and U. Rolf, "Extended Kalman filter as an exponential observer for nonlinear systems," *IEEE Trans. Signal Process.*, vol. 47, no. 8, pp. 2324–2328, Aug. 1999.
[43] W. Farrell, "Interacting multiple model filter for tactical ballistic missile tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 2, pp. 418–426, Apr. 2008.
[44] P. D. Hanlon and P. S. Maybeck, "Multiple-model adaptive estimation using a residual correlation Kalman filter bank," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 2, pp. 393–406, Apr. 2000.
[45] S. Kamalasadan and A. A. Ghandakly, "A novel multiple reference model adaptive control approach for multimodal and dynamic systems," *Control Intell. Syst.*, vol. 36, no. 2, pp. 119–128, Mar. 2008.

Zhi-Jie Zhou received the B.Eng. and M.Eng. degrees from the High-Tech Institute of Xi'an, Xi'an, China, in 2001 and 2004, respectively. He is also working toward the Ph.D. degree in Tsinghua University, Beijing, China, and was a Visiting Scholar at the University of Manchester, Manchester, U.K., from March to August 2009.

He is a Lecturer at the High-Tech Institute of Xi'an. His research interests include belief-rule-base, dynamic system modeling, hybrid quantitative and qualitative decision modeling, and fault prognosis and optimal maintenance of dynamic systems. He has published approximately 25 articles to date.

Chang-Hua Hu received the B.Eng. and M.Eng. degrees from the High-Tech Institute of Xi'an, Xi'an, China, in 1987 and 1990, respectively. He received the Ph.D. degree from North Western Polytechnic University, Xi'an, in 1996.

He is a Professor with the High-Tech Institute of Xi'an. He was a Visiting Scholar at the University of Duisburg, Duisburg, Germany, from September to December 2008. His current research has been supported by the National Science Foundation of China. He has published two books and approximately 100 articles. His research interests include fault diagnosis and prediction, life prognosis, and fault-tolerant control.



Jian-Bo Yang received the B.Eng. and M.Eng. degrees from North Western Polytechnic University, Xi'an, China, in 1981 and 1984, respectively, and the Ph.D. degree in decision sciences and systems engineering from Shanghai Jiao Tong University, Shanghai, China, in 1987.

He is a Professor of Decision and System Sciences and the Director of the Decision Sciences Research Center at the Manchester Business School of The University of Manchester, Manchester, U.K. He is also a Specially Appointed Changjian Chair Professor at the School of Management of Hefei University of Technology, Hefei, China. Prior to his current appointment, he was a faculty member of the University of Manchester Institute of Science and Technology, Manchester, U.K. (1998–2004, 1990), the University of Birmingham, Birmingham, U.K. (1995–1997), the University of Newcastle-upon-Tyne, Tyne, U.K. (1991–1995), and Shanghai Jiao Tong University, Shanghai, China, (1987–1989). In the past two decades, he has been conducting research in multiple criteria decision analysis under uncertainty, multiple objective optimization, intelligent decision support systems, hybrid quantitative and qualitative decision modeling using techniques from operational research, artificial intelligence, and systems engineering, and dynamic system modeling, simulation, and control for engineering and management systems. His current applied research is in design decision making, risk modeling and analysis, production planning and scheduling, quality modeling and evaluation, supply chain modeling and supplier assessment, and the integrated evaluation of products, systems, projects, policies, etc. His current research has been supported by EPSRC, EC, DEFRA, SERC, HKRGC, NSFC, and industry. He has published three books, over 140 papers in international journals/book chapters, and a similar number of papers in conferences and developed several software packages in these areas including the Windows-based intelligent decision system via evidential reasoning.

Dong-Ling Xu received the B.Eng. degree in electrical engineering from Hunan University, Changsha, China, in 1983 and the M.Eng. and Ph.D. degrees in system control engineering from Shanghai Jiao Tong University, Shanghai, China, in 1986 and 1988, respectively.

She is a Senior Lecturer in decision and system sciences at the Manchester Business School of The University of Manchester, Manchester, U.K. She was a Senior Software Engineer at an IT company from 1995 to early 2001, a Postdoctoral Research Associate at the University of Newcastle-upon-Tyne, Tyne, U.K., from 1993 to 1995, and a Lecturer and then an Associate Professor at East China University in Shanghai, Shanghai, from 1988 to 1993. She has developed several interactive web-based decision support tools and helped to develop a Windows-based intelligent decision system. These tools are used in a wide range of assessment activities, such as safety and risk assessment, organizational self-assessment in quality management, product evaluation, financial investment policy assessment, and supplier evaluation. In the past 15 years, she has worked in or in close collaboration with industry, applying advanced theory and technology for product quality modeling, fault diagnosis, and risk management. She has published a book and over 50 papers, many in high-quality international journals including IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS, *European Journal of Operational Research*, *Journal of Multiple Criteria Decision Analysis*, and *Decision Support Systems*.

Dong-Hua Zhou (SM'99) received the B.Eng., M.Sci., and Ph.D. degrees from the Department of Automation, Shanghai Jiaotong University, Shanghai, China, in 1985, 1988, and 1990, respectively.

He is currently a Professor in the Department of Automation, Tsinghua University, Beijing, China. He was an Alexander von Humboldt Research Fellow (1995–1996) in the University of Duisburg, Duisburg, Germany, and a Visiting Scholar at Yale University, New Haven, CT, (July 2001–January 2002). He has published over 80 international journal papers and three monographs in the areas of process identification, fault diagnosis, and fault-tolerant control.

Dr. Zhou is a member of the IFAC Technical Committee on Fault Diagnosis and Safety of Technical Processes, a senior member of IEEE, an Associate Editor of the Journal of Process Control, the Deputy General Secretary of the Chinese Association of Automation (CAA), and a council member of CAA. He was also the NOC Chair of the 6th IFAC Symposium on SAFEPROCESS, 2006.