



# Iterative learning belief rule-base inference methodology using evidential reasoning for delayed coking unit

Xiaodong Yu<sup>a,b</sup>, Dexian Huang<sup>a,b,\*</sup>, Yongheng Jiang<sup>a,b</sup>, Yihui Jin<sup>a</sup>

<sup>a</sup> Department of Automation, Tsinghua University, Beijing 100084, China

<sup>b</sup> Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084, China

## ARTICLE INFO

### Article history:

Received 31 August 2011

Accepted 6 February 2012

Available online 22 March 2012

### Keywords:

Belief rule-base

Evidential reasoning

Expert system

Iterative learning

Feedforward compensation

Delayed coking unit

## ABSTRACT

The belief rule-base inference methodology using evidential reasoning (RIMER) approach has been proved to be an effective extension of traditional rule-based expert systems and a powerful tool for representing more complicated causal relationships using different types of information with uncertainties. With a predetermined structure of the initial belief rule-base (BRB), the RIMER approach requires the assignment of some system parameters including rule weights, attribute weights, and belief degrees using experts' knowledge. Although some updating algorithms were proposed to solve this problem, it is still difficult to find an optimal compact BRB. In this paper, a novel updating algorithm is proposed based on iterative learning strategy for delayed coking unit (DCU), which contains both continuous and discrete characteristics. Daily DCU operations under different conditions are modeled by a BRB, which is then updated using iterative learning methodology, based on a novel statistical utility for every belief rule. Compared with the other learning algorithms, our methodology can lead to a more optimal compact final BRB. With the help of this expert system, a feedforward compensation strategy is introduced to eliminate the disturbance caused by the drum-switching operations. The advantages of this approach are demonstrated on the UniSim<sup>TM</sup> Operations Suite platform through the developed DCU operation expert system modeled and optimized from a real oil refinery.

© 2012 Elsevier Ltd. All rights reserved.

## 1. Introduction

Expert systems (ES) are a branch of applied artificial intelligence (AI), and were developed by the AI community in the mid-1960s. The basic idea behind ES is simply that expertise, which is the vast body of task-specific knowledge, is transferred from a human to a computer. This knowledge is then stored in the computer and users call upon the computer for specific advice at a specific conclusion. Then like a human consultant, it gives advices and explains, if necessary, the logic behind the advice (Giarratano & Riley, 1989; Jackson, 1998). In the last five decades, a large number of ES methodologies have been proposed in literatures, and applications implemented in industry fields (Duan, Yang, Li, Gui, & Deng, 2008; Liao, 2005).

Among these, the rule-based ES has been proved to be an effective and quite understandable tool. However, it is inevitable to deal with uncertainty caused by vagueness intrinsic to human knowledge and imprecision or incompleteness resulting from the

limit of human knowledge (Yang, Liu, Wang, Sii, & Wang, 2006). It is therefore necessary to use a scheme for representing and processing the vague, imprecise, and incomplete information in conjunction with precise data. These methods for representing and reasoning with uncertain knowledge, such as Bayesian probability theory (Jensen, 1996), Dempster–Shafer (D–S) theory of evidence (Binaghi & Madella, 1999) and rough set theory (Pawlak, 1991), have attracted much attention in academic research (Yang, Liu, Wang, Sii, & Wang, 2006). Nevertheless, it is impossible for us to use only one of these methods to solve the real problem, which may contain different kinds of uncertainties. In order to develop a generalized knowledge representation scheme and inference methodology to deal with these hybrid uncertainties, a new approach was proposed for building a hybrid rule-base using a belief structure and for inference in the rule-based system using the evidential reasoning theory by Yang et al. (Wang, Yang, Xu, & Chin, 2006; Xu et al., 2007; Yang, Liu, Wang, Sii, & Wang, 2006; Yang, Liu, Xu, Wang, & Wang, 2007). The methodology, based on D–S theory of evidence, decision theory and fuzzy set theory, is referred to as a generic belief rule-base inference methodology using evidential reasoning approach – RIMER (Yang, Liu, Wang, Sii, & Wang, 2006). The RIMER approach provides a more informative and flexible scheme than the traditional IF-THEN rule-base for

\* Corresponding author at: Department of Automation, Tsinghua University, Beijing 100084, China. Tel.: +86 10 62784964; fax: +86 10 62786911.

E-mail address: [huangdx@tsinghua.edu.cn](mailto:huangdx@tsinghua.edu.cn) (D. Huang).

knowledge representation, and is capable of capturing vagueness, incompleteness, and nonlinear causal relationships. In recent years, RIMER has already been applied to the safety analysis of off-shore systems (Liu, Yang, Wang, & Sii, 2005), pipeline leak detection (Xu et al., 2007; Zhou, Hu, Yang, Xu, & Zhou, 2009; Zhou, Hu, Yang, Xu, & Zhou, 2011), clinical decision support systems (Kong, Xu, Liu, & Yang, 2009) and stock trading expert systems (Dymova, Sevastianov, & Bartosiewicz, 2010).

In recent years, delayed coking technology is playing a more and more important role in modern oil refineries (Anthony, Kruse, & Ewy, 1996; Ellis, Paul, & Session, 1998; Friedman, 2005; Haseloff, Friedman, & Goodhart, 2007; Rodríguez-Reinoso, Santana, Palazon, Diez, & Marsh, 1998; Valyavin, Khukhrin, & Valyavin, 2007). It is a thermal cracking process used in petroleum refineries to upgrade and convert petroleum residuum (bottoms from atmospheric and vacuum distillation of crude oil) into liquid and gas product streams leaving behind a solid concentrated carbon material, petroleum coke. With short residence time in the furnace tubes, coking of the feed material is thereby “delayed” until it reaches large coking drums downstream of the heater.

Nevertheless, delayed coking is such a petrochemical process with strong coupling, non-linearity, long time-delay. It is the only main process in a modern petroleum refinery that is a batch-continuous process (Ellis, Paul, & Session, 1998). The flow through the tube furnace is continuous. The feed stream is switched between two drums. One drum is on-line filling with coke while the other one is being steam-stripped, cooled, coke removed, pressure tested, and warmed up. Thus, it is hard to implement effective automatic control to this unit (Friedman, 2005; Haseloff, Friedman & Goodhart, 2007; Zhou, Wang, & Jin, 2009). First, most of operations in drum-switching process are performed manually based on operators' experiences. As a result, the impact on the downstream unit such as the fractionator varies with different operators, fresh feed and also switching time. Second, the delayed coking fractionator is such a complex tower with multi-component and multi-side-draw. On one hand, there are strong non-linearity and large time-delay. On the other hand, it can not be ignored that great disturbance will be brought into the whole process because of the periodic drum-switching operation, which is hard for the traditional PID controller to eject effectively. During the past decade, various advanced process control (APC) technologies have been applied in DCU operations (Elliott, 2003; Haseloff Friedman & Goodhart, 2007). For example, a multivariate model predictive controller was designed and implemented on the fractionator of a DCU in a refinery company in China by Zhao et al. (Zhao, Chu, Su, & Huang, 2010). Whereas, in most APC technologies, to the best of our knowledge, the drum-switching disturbance has not been handled well so far (Yu, et al., 2011). Thus, it is quite important to develop efficient and robust techniques for such complex process. In our previous work (Yu et al., 2011), a rule-based expert system of intelligent switching expert system for DCU operations was established and a feedforward control strategy based on iterative learning was introduced to eliminate disturbances arising from the drum-switching operations. While nevertheless, it is a traditional rule-based expert system, and these simple rules can not represent more complicated causal relationships with uncertainties.

In this paper, a novel iterative learning belief rule-base inference methodology using evidential reasoning (IL-RIMER) is proposed and applied to construct a DCU operation expert system for providing optimal operating information for the field operators. Then a feedforward compensation strategy is incorporated into this expert system and implemented to smooth the operating process while drum-switching. In the following Section 2, the RIMER theory will be reviewed briefly, followed by a detailed

description of the IL-RIMER scheme in Section 3. Then Section 4 shows how a DCU operation expert system can be developed using the IL-RIMER methodology proposed, based on the field data from a real oil refinery. And the effectiveness and efficiency of this expert system is illustrated on the UniSim™ Operations Suite platform subsequently. Finally the paper is concluded in Section 5, followed by some acknowledgments. The basic idea of our algorithm was previously explored by Yu et al. (Yu, Huang, Jiang, & Jin, 2011). This paper represents a significant extension in terms of experimental methodology, parameterization, and analysis.

## 2. The RIMER theory

### 2.1. Belief rule-base

A BRB, which captures the dynamic of a system, consists of a collection of belief rules defined as follows (Yang Liu, Wang, Sii, & Wang, 2006):

$$R_k: \text{IF } x_1 \text{ is } A_1^k \wedge x_2 \text{ is } A_2^k \dots x_{T_k} \text{ is } A_{T_k}^k \text{ THEN} \\ \{(D_1, \beta_{1k}), (D_2, \beta_{2k}), \dots, (D_N, \beta_{Nk})\} \quad (1)$$

with a rule weight  $\theta_k$  and attribute weight  $\delta_{k1}, \delta_{k2}, \dots, \delta_{kT_k}$ , where  $x_1, x_2, \dots, x_{T_k}$  represents the antecedent attributes in the  $k$ th rule  $R_k$ ,  $A_i^k (i=1, 2, \dots, T_k, k=1, 2, \dots, L)$  is the referential value of the  $i$ th antecedent attribute in the  $k$ th rule  $R_k$ ,  $A_i^k \in A_i, A_i = \{A_{ij} | j=1, 2, \dots, J_i\}$  is a set of referential value of the  $i$ th antecedent attribute,  $J_i$  is the number of the referential value,  $\theta_k (\in R^+, k=1, 2, \dots, L)$  is the relative weight of the  $k$ th rule  $R_k$ ,  $\delta_{k1}, \delta_{k2}, \dots, \delta_{kT_k}$  are the relative weights of the  $T_k$  antecedent attributes used in the  $k$ th rule  $R_k$ , and  $\beta_{ik} (i=1, 2, \dots, N, k=1, 2, \dots, L)$  is the belief degree assessed to  $D_j$  which denotes the  $j$ th consequent. If  $\sum_{i=1}^N \beta_{ik} = 1$ , the  $k$ th rule  $R_k$  is said to be complete; otherwise, it is incomplete. Note that “ $\wedge$ ” is a logical connective to represent the “AND” relationship. In addition, suppose that  $T$  is the total number of antecedent attributes used in the rule base.

### 2.2. Belief rule-base inference methodology using evidential reasoning approach

Given an input to the system,  $U(t) = \{U_i(t), i=1, 2, \dots, T_k\}$ , how can the rule-base be used to inference and generate an output? As mentioned earlier,  $T_k$  is the total number of antecedents in the rule-base,  $U_i(t) (i=1, 2, \dots, T_k)$  is the  $i$ th attribute, which can be one of the following types (Yang Liu, Wang, Sii, & Wang, 2006): continuous, discrete, symbolic and ordered symbolic.

Before the start of an inference process, the matching degree of an input to each referential value in the antecedents of a rule needs to be determined so that an activation weight for each rule can be generated. This is equivalent to transforming an input into a distribution on referential values using belief degrees and can be accomplished using different techniques such as the rule or utility-based equivalence transformation techniques (Yang, 2001; Yang, Liu, Xu, Wang & Wang, 2007).

Using the notations provided above, the activation weight of the  $k$ th rule  $R_k$ ,  $w_k$ , is calculated as (Yang Liu, Wang, Sii, & Wang, 2006):

$$\omega_k = \frac{\theta_k a_k}{\sum_{i=1}^L \theta_i a_i} \quad (2)$$

where  $a_k$  is called the normalized combined matching degree, which can be calculated by

$$a_k = \prod_{i=1}^{T_k} (a_i^k)^{\delta_{ki}} \quad (3)$$

with  $\bar{\delta}_{ki} = \delta_{ki} / \max_{i=1, \dots, T_k} \{\delta_{ki}\}$ ,  $a_i^k \in \{a_{ij}, i = 1, 2, \dots, T_k, j = 1, 2, \dots, J_i\}$  is the individual matching degree to which the input  $x_i \in U$  matches the  $i$ th referential value  $A_i^k$  of the packet antecedent  $A^k$  in the  $k$ th rule  $R_k$ , and  $a_i^k \geq 0$  and  $\sum_{i=1}^{T_k} a_i^k \leq 1$ .

In the RIMER,  $a_i^k$  can be generated using various ways depending on the different types of the input information. In Yang's paper (Yang, 2001), an important technique, i.e., rule-based information transformation technique was proposed to deal with the input information that includes qualitative assessment and quantitative data. In this paper, Appendix A gives a brief review of this technique for the quantitative data.

Having determined the activation weight of each rule in the rule-base, the ER approach can be directly applied to combine the rules and generate final conclusions. Suppose the outcome of the combination yields the following:

$$O(U(t)) = \hat{y}(t) = \{(D_j, \beta_j), j = 1, \dots, N\} = \sum_{j=1}^N u(D_j) \beta_j \quad (4)$$

The outcome expressed by Eq. (4) reads that if the input is given by  $U(t) = \{U_i(t), i = 1, 2, \dots, T_k\}$ , then the consequent is  $D_1$  to a degree of  $\beta_1$ ,  $D_2$  to a degree of  $\beta_2, \dots$ , and  $D_N$  to a degree of  $\beta_N$ . Using the analytical format of the ER algorithm (Wang, Yang, Xu, Chin, 2006) the combined belief degree  $\beta_j$  in  $D_j$  can be generated as follows:

$$\beta_j = \frac{\mu^*[\prod_{k=1}^L \omega_k \beta_{jk} + 1 - \omega_k \prod_{i=1}^N \beta_{ik} - \prod_{k=1}^L (1 - \omega_k \sum_{i=1}^N \beta_{ik})]}{1 - \mu^*[\prod_{k=1}^L (1 - \omega_k)]}, \quad j = 1, \dots, N \quad (5)$$

where

$$\mu = \sum_{i=1}^N \prod_{k=1}^L \left( \omega_k \beta_{jk} + 1 - \omega_k \sum_{i=1}^N \beta_{ik} \right) - (N-1) \prod_{k=1}^L \left( 1 - \omega_k \sum_{i=1}^N \beta_{ik} \right) \quad (6)$$

and  $\omega_k$  is as given in Eq. (2).

### 2.3. Optimization methods for training a BRB

As mentioned earlier, a BRB can be constructed by extracting knowledge from experts directly. Still, it is also true that the performance of this kind of expert system can be improved if the rules are well adjusted through learning from new available data (Xu et al., 2007). Furthermore, in practical situations, it is difficult to accurately determine the parameters of a BRB entirely subjectively, particularly, for a large-scale BRB with hundreds or even thousands of rules. In addition, a change in rule weight or attribute weight may lead to changes in the performance of a BRB (Yang et al., 2007). As such, there is a need to develop a supporting mechanism that can be used to train, in a locally optimal way, the BRB that is initially built using expert knowledge.

In Yang's paper (Yang, Liu, Xu, Wang & Wang, 2007), the process of training a BRB was sketched as Fig. 1, where  $U$  is a given input,  $\tilde{O}$  the corresponding observed output either measured using instruments or assessed by experts,  $O$  the simulated output generated by the BRB system,  $\xi(P)$  the difference between

$\tilde{O}$  and  $O$ , and

$$P = (\beta_{ik}, \theta_k, \delta_j; i = 1, 2, \dots, N; k = 1, 2, \dots, L; j = 1, 2, \dots, T_k) \quad (7)$$

are the adjustable parameters. The objective of the training is to minimize the difference  $\xi(P)$  by adjusting the parameters  $P$ . This objective is difficult to achieve manually even by experts, however there are computer algorithms available to solve the problem. Several new optimization models for locally training a BRB were developed by Yang et al. (Yang, Liu, Xu, Wang & Wang, 2007).

### 3. A novel iterative learning algorithm of updating a RIMER model for batch processes

#### 3.1. Preliminaries

**Definition 1.** A BRB is said to be *complete*, if and only if for every set of premises  $\Gamma$ , any formula which semantically follows from  $\Gamma$  is derivable from  $\Gamma$ . That is,

$$\Gamma \models_s \phi \rightarrow \Gamma \vdash_s \phi \quad (8)$$

**Definition 2.** A rule is said to be *redundant*, if and only if any set derived from the original rule-base is neither reduced nor expanded when this rule is removed.

Obviously, repeating rules are *redundant*. Moreover, it should be noted that there are always two different forms of *redundant* rules in an expert system. That is,

**Definition 2.1.** A *redundant* rule is said to be *explicit*, if and only if this rule can never be quoted in any inference procedure. In other words, there is no any benefit to the entire system with the existence of this rule.

**Definition 2.2.** A *redundant* rule is said to be *implicit*, if and only if this rule can sometimes be activated under certain conditions, however, there is little impact to any reasoning results with the existence of this rule.

**Definition 3.** A BRB is said to be *compact*, if and only if the BRB is *complete* without any *redundant* rules. Namely, a compact BRB is a *complete* one with the least non-*redundant* rules.

With a predetermined structure of the initial BRB, the RIMER approach requires the assignment of some system parameters including rule weights, attribute weights, and belief degrees using experts' knowledge, which is hard for engineers to decide objectively and precisely. As such, some optimization models have been proposed to train a BRB (Yang, Liu, Xu, Wang & Wang, 2007). Because these models are off-line trained and in essence are locally optimal, it is very expensive and time consuming to train and re-train them (Zhou et al., 2010). In order to solve these problems, the recursive algorithms for online updating the BRB systems have also been developed and they are fast to converge, which is very important for training systems that have a high level of real-time requirement (Zhou, Hu, Yang, Xu, & Zhou, 2009). However, these optimal algorithms are all based on a predetermined structure of BRB with all the combination of the antecedent attributes (Dymova, Sevastinov, & Bertosiewicz, 2010; Xu et al., 2007; Zhou et al., 2009, 2010). Due to vagueness intrinsic to human knowledge and imprecision or incompleteness resulting from the limit of human knowledge, the prior knowledge for a real complex system may be incomplete, or even inappropriate, which may produce some *redundant* rules, and consequently leads to a *complete*, but not compact structure of BRB. To achieve an global optimal BRB, it is not sufficient to just statistically tune

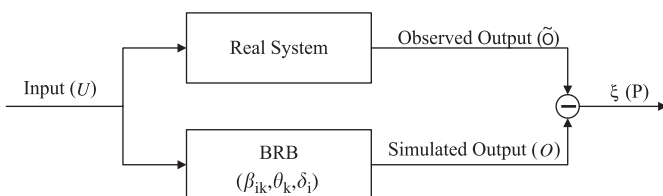


Fig. 1. Illustration of optimal learning process for a BRB.

the parameters for a given BRB, but the structure of a BRB need to be adjusted as well (Zhou et al., 2010). With this end in view, a sequential learning algorithm was proposed for online constructing more compact BRB systems by Zhou et al. (Zhou et al., 2010). Based on the definition of the new concept of statistical utility for a belief rule, a belief rule can be automatically added into a BRB or pruned from the BRB. Unfortunately, if the available training data is not perfect, a possible drawback of this algorithm is that the newly constructed BRB might not provide a representative set of rules for simulating the original system (Zhou et al., 2010). Meanwhile, Zhou's BRB is updated from an initial structure with just two rules, and the newly added ones depend entirely on the training data. As a result of this, the new referential values of both the antecedent attribute and the consequent attribute may not be representative, which may achieve atypical rules for the BRB. Therefore, although compact comparatively, this BRB system is not optimal, even may leads to a non-compact one.

### 3.2. The basic scheme of IL-RIMER algorithm

In this section, a novel iterative learning belief rule-based inference methodology using evidential reasoning for batch processes is proposed. As is shown in Fig. 2, this model mainly consists of two modules: the modified RIMER model and IL training module. In our paper, a statistical utility for each belief rule defined by Zhou et al. (2010) is introduced into the traditional RIMER model. The initial BRB is constructed with all the combinations of the referential values of all the antecedent attributes. Based on each batch, the parameters of the constructed RIMER are updated with iterative learning strategy, including the rule weights, the attribute weights, the belief degrees of the BRB system, and the statistical utilities for all the belief rules. The recursive algorithm based on the recursive expectation maximization (EM) algorithm proposed by Zhou et al. (Zhou, Hu, Yang, Xu, & Zhou, 2009) is introduced into batch processes here and run along the batch direction. Then the pruning operation is implemented in each batch, according to the statistical utilities of the belief rules, which is briefly reviewed in Appendix B. Due to the fact that the initial BRB is constructed with all the possible combinations using prior expert knowledge, rather than only two rules determined by the data constraints in Zhou's paper (Zhou et al., 2010), it is most likely to generate an optimal compact BRB.

As a result of the earlier mentioned discussion, the procedure of the proposed IL-RIMER algorithm can be summarized as the following steps:

Step 1: Determine the antecedent attributes and consequent attributes for a real problem.

Step 2: Set appropriate referential values of both antecedent attributes and consequent attributes.

Step 3: Construct the initial BRB with all the combination of the referential values of all the antecedent attributes, and calculate the belief degrees.

Step 4: For the  $K$ th batch iteration:

Step 4.1: Update the parameters of the constructed RIMER as Eq. (9) with iterative learning strategy, including the rule weights, the attribute weights, the belief degrees of the BRB system, and the statistical utilities for all the belief rules.

$$P = (\beta_{ik}^K, \theta_k^K, \delta_j^K, SU_k^K, \quad i = 1, 2, \dots, N; \quad k = 1, 2, \dots, L; \quad j = 1, 2, \dots, T_k) \quad (9)$$

Step 4.2: If the criteria as Eq. (10) is satisfied, the  $k$ th rule is removed, and the RIMER model is updated. Then the dimensionality of the BRB is reduced. Otherwise, go back to the  $(K+1)$ th iteration.

$$\frac{\sum_{i=K-4}^K SU_k^i}{5} < e_{\text{delete}}, \quad k = 1, 2, \dots, L; \quad (10)$$

## 4. IL-RIMER for DCU operation expert system

### 4.1. Problem description

A simplified diagram of DCU is shown in Fig. 3. The unit takes fresh feed, heats it and injects it into the bottom of the main fractionator, where it is mixed with an internal reflux recycle of heavy cracked material. The total fresh and recycled feed are heated in the DCU furnace to a high cracking temperature. Hot partially cracked feed flows from the DCU furnace into the coke drum, where the reaction continues. Cracked distillate vapor ascends in the coke drum and flows into the fractionator for separation.

Coke remains in the drum and is periodically removed. That is the main reason for DCU process being such a difficult unit to operate. Twice daily filled coke drum are switched off for coke removal and the empty one are connected. The drum that was just filled then goes through a cycle of steaming out, cooling, opening, coke removing, closing, steaming, pressure testing, warming up and finally reconnecting to the furnace and fractionator.

DCU is the most effective process to decarbonize and demetallize heavy petroleum residues. However, inevitable periodic drum-switching operation makes DCU process quite different from other petrochemical processes. All the drum-switching operations are performed based on the field operators' experiences. As a result, the impact on the downstream unit such as the fractionator varies with the different operators. The main objective of DCU system operation is to keep the system stable as best as we can. In more specific terms, how to eliminate the drum-switching disturbance to the main fractionator is a difficult challenge we must handle.

### 4.2. Constructing the IL-RIMER for DCU

Step 1: Set the initial BRB.

The antecedent and consequent attributes:

Table 1 shows the drum-switching operation and the effect to the fractionator summarized from on-site data in an oil refinery. According to the field operators' expert knowledge, when operating on a DCU, some drum-switching data can be obtained. These data include the opening of the feed valve 1 and 2, and the steam flow, denoted by Valve<sub>1</sub>, Valve<sub>2</sub>, and Flow, respectively. They are the three antecedent attributes of the BRB and are defined as follows:

$$U_1 = \text{Valve}_1(t) \quad (11)$$

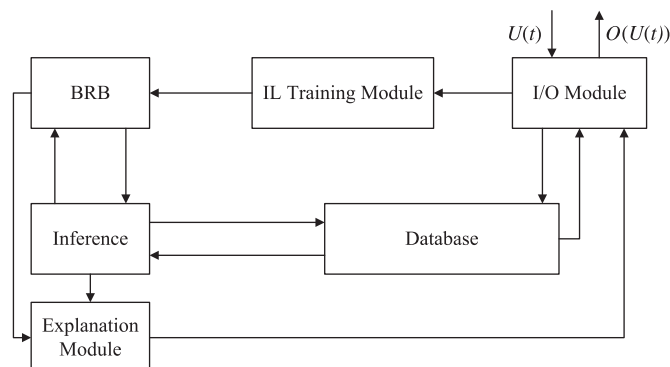


Fig. 2. The structure of IL-RIMER algorithm.



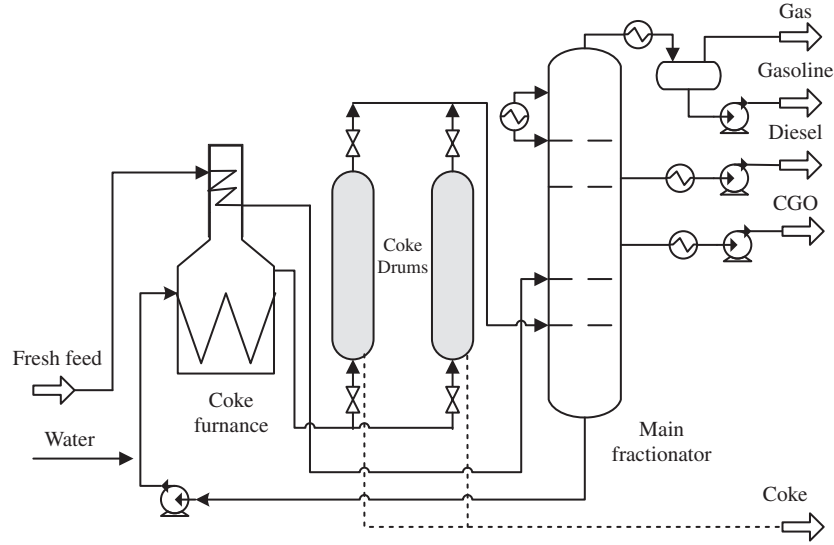


Fig. 3. The technologic flow-chart of the delayed coking unit.

Table 1

The drum-switching operations of delayed coking unit in a refinery.

Drum 1	Drum 2	Opening of the feed valve 1 (%)	Opening of the feed valve 2 (%)	Steam flow (kg/h)	Time (h)	Effect (Y/N)
Steaming	Coking	100	100	6,000	2	Y
Steaming out		0		20,000	2	N
Cooling				0	6.5	
Coke removing					5	
Pressure testing				10,000	0.5	
Warming up 1	30			0	1	Y
Warming up 2	70				1	
Warming up 3	100				6	
Coking	Steaming	100	100	6,000	2	Y
	Steaming out		0	20,000	2	N
	Cooling			0	6.5	
	Coke removing				5	
	Pressure testing			10,000	0.5	
	Warming up 1	30		0	1	Y
	Warming up 2	70			1	
	Warming up 3	100			6	

$$U_2 = \text{Valve}_2(t) \quad (12)$$

$$U_3 = \text{Flow}(t) \quad (13)$$

The two consequent attributes are the percent of the differences during a time interval of the bottom temperature and bottom level of the fractionator, denoted by TempDiff and LevelDiff, which are calculated as follows:

$$D_1 = \text{TempDiff}(t) = \frac{\text{Temp}(t) - \text{Temp}(t-1)}{\text{Temp}(t-1)} \quad (14)$$

$$D_2 = \text{LevelDiff}(t) = \frac{\text{Level}(t) - \text{Level}(t-1)}{\text{Level}(t-1)} \quad (15)$$

The referential points of antecedent and consequent attributes:

The number of referential points used of each antecedent attributes decides the size of the BRB. If the number is too large, there will be too many rules in the BRB, and the subsequent training and inference process will be more demanding. If it is too small, the points may not be able to cover the value range of an antecedent attribute. Normally 5–9 referential points are used for a conventional BRB (Dymova, Sevastinov, & Bertosiwicz, 2010; Xu et al., 2007). The number of referential points for a consequent attribute is also comparable to those of the antecedent attributes. In this paper, we use 3 points for Valve<sub>1</sub> and Valve<sub>2</sub>, which are Zero(Z), Medium(M), Large(L). That is

$$A_1^k \in \{Z, M, L\} \quad (16)$$

$$A_2^k \in \{Z, M, L\} \quad (17)$$

Similarly we use 4 points for Flow and they are Z, M, L, and VL, i.e.,

$$A_3^k \in \{Z, M, L, VL\} \quad (18)$$

For the consequent attributes, TempDiff and LevelDiff, seven referential points are used: Negative Very Large(NVL), Negative Large(NL), Negative Small(NS), Zero(Z), Positive Small(PS), Positive Large(PL), Positive Very Large(PVL), i.e.,

$$D_1 \in \{NVL, NL, NS, Z, PS, PL, PVL\} \quad (19)$$

$$D_2 \in \{NVL, NL, NS, Z, PS, PL, PVL\} \quad (20)$$

In order to use the field data, the referential points defined above for the antecedent and consequent attributes are in linguistic terms and need to be quantified. By examining the calculated TempDiff and LevelDiff, and the recorded Valve<sub>1</sub>, Valve<sub>2</sub>, and Flow values, the following equivalent relationships between the linguistic terms and numerical values are assumed so that the values roughly cover the corresponding attribute value range.

For Valve<sub>1</sub> and Valve<sub>2</sub>, it is assumed that

$$\{Z = 0, \quad M = 50, \quad L = 90\} \quad (21)$$

For Flow, it is assumed that

$$\{Z = 0, \quad M = 5000, \quad L = 8000, \quad VL = 15,000\} \quad (22)$$

For TempDiff, it is assumed that

$$\{NVL = -5\%, NL = -1\%, NS = -0.1\%, \\ Z = 0, PS = 0.1\%, PL = 1\%, PVL = 5\%\} \quad (23)$$

For LevelDiff, it is assumed that

$$\{NVL = -50\%, NL = -20\%, NS = -5\%, Z = 0, \\ PS = 5\%, PL = 20\%, PVL = 50\%\} \quad (24)$$

Rules:

Because Valve<sub>1</sub> and Valve<sub>2</sub> are both divided into 3 terms, respectively and Flow 4 terms, there are 36 combinations of the 3 antecedents leading to 36 rules in total in the initial BRB. In our case, there is no previous rule-base to start with. Rules are extracted by examining the on-site data and using field engineers' experiences, and are used as the starting point for updating online. Table 2 lists the initial 36 belief rules provided by analyzing the field data and operating experiences. For example, Rule 9 in Table 2 is stated as follows:

$$R_9 : \text{IF Valve}_1 \text{ is } Z \wedge \text{Valve}_2 \text{ is } L \wedge \text{Flow is } Z \\ \text{THEN TempDiff is } \{(Z, 0.6890), (PS, 0.3110)\} \\ \wedge \text{LevelDiff is } \{(Z, 0.9729), (PS, 0.0271)\} \quad (25)$$

Step 2: Construct a BRB for DCU online.

Yang's updating strategy for the initial BRB was also implemented in this paper for a comparison. Table 3 shows the updated BRB without batch iterative learning and all the 36 rules' parameters were updated. It should be noted that, as a result of the shortage of structure updating strategy for the entire BRB, some *redundant* rules were also trained. This phenomenon brings us not

only the burden of computation and waste of time, but also the approximation error. Due to the imprecision of the field data, some *redundant* rules which have definite results in essence were trained and updated with an uncertain consequent output. Take Rule 1 for example, the operation condition of this combination of the 3 antecedents never happens in the on-site DCU operating procedure. Obviously, Rule 1 is a redundant one, and there is no need to take into consideration. Unfortunately, belief degrees for the 2 consequents were assigned and updated in Table 3 as follows,

$$\text{TempDiff is } \{(Z, 0.9922), (PS, 0.0078)\} \\ \wedge \text{LevelDiff is } \{(NS, 0.0892), (Z, 0.9108)\} \quad (26)$$

which leads to an unreasonable output of  $\{D_1 = 0.008, D_2 = -0.4462\}$ .

Table 4 shows the updated BRB after 10 batches iterative learning with our strategy proposed in Section 3.2. It is noted that there is a significant reduction of the BRB, from 36 rules to only 11 ones. And the Rule No. in Table 4 is the original Rule No. in Table 2 and Table 3, which was for the sake of correspondence of each rule.

In addition, through a further analysis of the rules remaining in Table 4, it is interesting to note that these rules all describe a specific status of the DCU operation, respectively. Take Rule 21 for an example, the  $\{M, L, Z\}$  combination of the antecedent attributes, which means that there is no steam flow and one of the feed valve is opened moderately while the other is opened totally simultaneously, represents the operation conditions of "warming up". From the technical point of view and a closer look at the field

**Table 2**  
Initial BRB provided by analyzing the field data and operating experiences.

Rule No.	U $\{U_1, U_2, U_3\}$	D Value (%)		$D_1 = \{NVL, NL, NS, Z, PS, PL, PVL\}$ $\{\beta_{11}, \beta_{12}, \beta_{13}, \beta_{14}, \beta_{15}, \beta_{16}, \beta_{17}\}$	$D_2 = \{NVL, NL, NS, Z, PS, PL, PVL\}$ $\{\beta_{21}, \beta_{22}, \beta_{23}, \beta_{24}, \beta_{25}, \beta_{26}, \beta_{27}\}$
		$D_1$	$D_2$		
1	{Z, Z, Z}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
2	{Z, Z, M}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
3	{Z, Z, L}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
4	{Z, Z, VL}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
5	{Z, M, Z}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
6	{Z, M, M}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
7	{Z, M, L}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
8	{Z, M, VL}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
9	{Z, L, Z}	0.0311	0.1356	{0, 0, 0, 0.689, 0.311, 0, 0}	{0, 0, 0, 0.9729, 0.0271, 0, 0}
10	{Z, L, M}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
11	{Z, L, L}	0.5981	0.5216	{0, 0, 0, 0, 0.4466, 0.5534, 0}	{0, 0, 0, 0.8957, 0.1043, 0, 0}
12	{Z, L, VL}	-0.0981	0.1951	{0, 0, 0.981, 0.019, 0, 0, 0}	{0, 0, 0, 0.961, 0.039, 0, 0}
13	{M, Z, Z}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
14	{M, Z, M}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
15	{M, Z, L}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
16	{M, Z, VL}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
17	{M, M, Z}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
18	{M, M, M}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
19	{M, M, L}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
20	{M, M, VL}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
21	{M, L, Z}	-2.3925	-18.6005	{0.3481, 0.6519, 0, 0, 0, 0, 0}	{0, 0.9067, 0.0933, 0, 0, 0, 0}
22	{M, L, M}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
23	{M, L, L}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
24	{M, L, VL}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
25	{L, Z, Z}	0.0109	0.2098	{0, 0, 0, 0.891, 0.109, 0, 0}	{0, 0, 0, 0.958, 0.042, 0, 0}
26	{L, Z, M}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
27	{L, Z, L}	0.6251	0.4306	{0, 0, 0, 0, 0.4166, 0.5834, 0}	{0, 0, 0, 0.9139, 0.0861, 0, 0}
28	{L, Z, VL}	-0.1108	0.2156	{0, 0.012, 0.988, 0, 0, 0, 0}	{0, 0, 0, 0.9569, 0.0431, 0, 0}
29	{L, M, Z}	-2.5009	-22.0130	{0.3752, 0.6248, 0, 0, 0, 0, 0}	{0.0671, 0.9329, 0, 0, 0, 0, 0}
30	{L, M, M}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
31	{L, M, L}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
32	{L, M, VL}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
33	{L, L, Z}	-0.5816	-0.5483	{0, 0.5351, 0.4649, 0, 0, 0, 0}	{0, 0, 0.1097, 0.8903, 0, 0, 0}
34	{L, L, M}	1.9325	0.2569	{0, 0, 0, 0, 0.7669, 0.2331}	{0, 0, 0, 0.9486, 0.0514, 0, 0}
35	{L, L, L}	0	0	{0, 0, 0, 1, 0, 0, 0}	{0, 0, 0, 1, 0, 0, 0}
36	{L, L, VL}	-0.1097	0.2129	{0, 0.0108, 0.9892, 0, 0, 0, 0}	{0, 0, 0, 0.9574, 0.0426, 0, 0}

**Table 3**  
Updated BRB with training data without batch iterative learning.

Rule No.	$U$ $\{U_1, U_2, U_3\}$	D Value (%)		$D_1 = \{NVL, NL, NS, Z, PS, PL, PVL\}$	$D_2 = \{NVL, NL, NS, Z, PS, PL, PVL\}$
		$D_1$	$D_2$	$\{\beta_{11}, \beta_{12}, \beta_{13}, \beta_{14}, \beta_{15}, \beta_{16}, \beta_{17}\}$	$\{\beta_{21}, \beta_{22}, \beta_{23}, \beta_{24}, \beta_{25}, \beta_{26}, \beta_{27}\}$
1	{Z, Z, Z}	0.0008	−0.4462	{0, 0, 0, 0.9922, 0.0078, 0, 0}	{0, 0, 0.0892, 0.9108, 0, 0, 0}
2	{Z, Z, M}	−0.0019	−0.3697	{0, 0, 0.0187, 0.9813, 0, 0, 0}	{0, 0, 0.0739, 0.9261, 0, 0, 0}
3	{Z, Z, L}	0.0010	0.0690	{0, 0, 0, 0.9896, 0.0104, 0, 0}	{0, 0, 0, 0.9862, 0.0138, 0, 0}
4	{Z, Z, VL}	0.0077	−0.3775	{0, 0, 0, 0.9226, 0.0774, 0, 0}	{0, 0, 0.0755, 0.9245, 0, 0, 0}
5	{Z, M, Z}	0.0025	−0.2129	{0, 0, 0, 0.9746, 0.0254, 0, 0}	{0, 0, 0.0426, 0.9574, 0, 0, 0}
6	{Z, M, M}	−0.0091	0.3404	{0, 0, 0.0914, 0.9086, 0, 0, 0}	{0, 0, 0, 0.9319, 0.0681, 0, 0}
7	{Z, M, L}	0.0051	0.2691	{0, 0, 0, 0.9492, 0.0508, 0, 0}	{0, 0, 0, 0.9462, 0.0538, 0, 0}
8	{Z, M, VL}	0.0050	0.2039	{0, 0, 0, 0.9504, 0.0496, 0, 0}	{0, 0, 0, 0.9592, 0.0408, 0, 0}
9	{Z, L, Z}	0.0311	0.1356	{0, 0, 0, 0.689, 0.311, 0, 0}	{0, 0, 0, 0.9729, 0.0271, 0, 0}
10	{Z, L, M}	0.0031	0.1035	{0, 0, 0, 0.9687, 0.0313, 0, 0}	{0, 0, 0, 0.9793, 0.0207, 0, 0}
11	{Z, L, L}	0.5981	0.5216	{0, 0, 0, 0.4466, 0.5534, 0}	{0, 0, 0, 0.8957, 0.1043, 0, 0}
12	{Z, L, VL}	−0.0981	0.1951	{0, 0, 0.981, 0.019, 0, 0, 0}	{0, 0, 0, 0.961, 0.039, 0, 0}
13	{M, Z, Z}	0.0017	−0.2358	{0, 0, 0, 0.9827, 0.0173, 0, 0}	{0, 0, 0.0472, 0.9528, 0, 0, 0}
14	{M, Z, M}	−0.0039	−0.3718	{0, 0, 0.039, 0.961, 0, 0, 0}	{0, 0, 0.0744, 0.9256, 0, 0, 0}
15	{M, Z, L}	−0.0006	0.1579	{0, 0, 0.0063, 0.9937, 0, 0, 0}	{0, 0, 0, 0.9684, 0.0316, 0, 0}
16	{M, Z, VL}	−0.0060	−0.0842	{0, 0, 0.0597, 0.9403, 0, 0, 0}	{0, 0, 0.0168, 0.9832, 0, 0, 0}
17	{M, M, Z}	−0.0022	−0.1000	{0, 0, 0.0217, 0.9783, 0, 0, 0}	{0, 0, 0.02, 0.98, 0, 0, 0}
18	{M, M, M}	0.0036	−0.4873	{0, 0, 0, 0.964, 0.036, 0, 0}	{0, 0, 0.0975, 0.9025, 0, 0, 0}
19	{M, M, L}	−0.0057	0.4210	{0, 0, 0.0573, 0.9427, 0, 0, 0}	{0, 0, 0, 0.9158, 0.0842, 0, 0}
20	{M, M, VL}	−0.0070	0.0139	{0, 0, 0.0699, 0.9301, 0, 0, 0}	{0, 0, 0, 0.9972, 0.0028, 0, 0}
21	{M, L, Z}	−2.3925	−18.6005	{0.3481, 0.6519, 0, 0, 0, 0, 0}	{0, 0.9067, 0.0933, 0, 0, 0, 0, 0}
22	{M, L, M}	−0.0017	−0.2350	{0, 0, 0.017, 0.983, 0, 0, 0}	{0, 0, 0.047, 0.953, 0, 0, 0, 0}
23	{M, L, L}	−0.0031	−0.2123	{0, 0, 0.0312, 0.9688, 0, 0, 0}	{0, 0, 0.0425, 0.9575, 0, 0, 0, 0}
24	{M, L, VL}	−0.0092	0.4147	{0, 0, 0.0918, 0.9082, 0, 0, 0}	{0, 0, 0, 0.9171, 0.0829, 0, 0}
25	{L, Z, Z}	0.0109	0.2098	{0, 0, 0, 0.891, 0.109, 0, 0}	{0, 0, 0, 0.958, 0.042, 0, 0}
26	{L, Z, M}	0.0091	0.2529	{0, 0, 0, 0.9091, 0.0909, 0, 0}	{0, 0, 0, 0.9494, 0.0506, 0, 0}
27	{L, Z, L}	0.6251	0.4306	{0, 0, 0, 0.4166, 0.5834, 0}	{0, 0, 0, 0.9139, 0.0861, 0, 0}
28	{L, Z, VL}	−0.1108	0.2156	{0, 0.012, 0.988, 0, 0, 0, 0}	{0, 0, 0, 0.9569, 0.0431, 0, 0}
29	{L, M, Z}	−2.5009	−22.0130	{0.3752, 0.6248, 0, 0, 0, 0, 0}	{0.0671, 0.9329, 0, 0, 0, 0, 0}
30	{L, M, M}	0.0002	0.1564	{0, 0, 0, 0.9985, 0.0015, 0, 0}	{0, 0, 0, 0.9687, 0.0313, 0, 0}
31	{L, M, L}	−0.0047	0.0433	{0, 0, 0.047, 0.953, 0, 0, 0}	{0, 0, 0, 0.9913, 0.0087, 0, 0}
32	{L, M, VL}	−0.0079	−0.0444	{0, 0, 0.0792, 0.9208, 0, 0, 0}	{0, 0, 0.0089, 0.9911, 0, 0, 0}
33	{L, L, Z}	−0.5816	−0.5483	{0, 0.5351, 0.4649, 0, 0, 0, 0}	{0, 0, 0.1097, 0.8903, 0, 0, 0}
34	{L, L, M}	1.9325	0.2569	{0, 0, 0, 0, 0.7669, 0.2331}	{0, 0, 0, 0.9486, 0.0514, 0, 0}
35	{L, L, L}	0.0056	0.4822	{0, 0, 0, 0.9441, 0.0559, 0, 0}	{0, 0, 0, 0.9036, 0.0964, 0, 0}
36	{L, L, VL}	−0.1097	0.2129	{0, 0.0108, 0.9892, 0, 0, 0, 0}	{0, 0, 0, 0.9574, 0.0426, 0, 0}

**Table 4**  
Updated BRB after 10 batches iterative learning.

Rule No.	$U$ $\{U_1, U_2, U_3\}$	D Value (%)		$D_1 = \{NVL, NL, NS, Z, PS, PL, PVL\}$	$D_2 = \{NVL, NL, NS, Z, PS, PL, PVL\}$
		$D_1$	$D_2$	$\{\beta_{11}, \beta_{12}, \beta_{13}, \beta_{14}, \beta_{15}, \beta_{16}, \beta_{17}\}$	$\{\beta_{21}, \beta_{22}, \beta_{23}, \beta_{24}, \beta_{25}, \beta_{26}, \beta_{27}\}$
9	{Z, L, Z}	0.0311	0.1356	{0, 0, 0, 0.689, 0.311, 0, 0}	{0, 0, 0, 0.9729, 0.0271, 0, 0}
11	{Z, L, L}	0.5981	0.5216	{0, 0, 0, 0.4466, 0.5534, 0}	{0, 0, 0, 0.8957, 0.1043, 0, 0}
12	{Z, L, VL}	−0.0981	0.1951	{0, 0, 0.981, 0.019, 0, 0, 0}	{0, 0, 0, 0.961, 0.039, 0, 0}
21	{M, L, Z}	−2.3925	−18.6005	{0.3481, 0.6519, 0, 0, 0, 0, 0}	{0, 0.9067, 0.0933, 0, 0, 0, 0, 0}
25	{L, Z, Z}	0.0109	0.2098	{0, 0, 0, 0.891, 0.109, 0, 0}	{0, 0, 0, 0.958, 0.042, 0, 0}
27	{L, Z, L}	0.6251	0.4306	{0, 0, 0, 0.4166, 0.5834, 0}	{0, 0, 0, 0.9139, 0.0861, 0, 0}
28	{L, Z, VL}	−0.1108	0.2156	{0, 0.012, 0.988, 0, 0, 0, 0}	{0, 0, 0, 0.9569, 0.0431, 0, 0}
29	{L, M, Z}	−2.5009	−22.0130	{0.3752, 0.6248, 0, 0, 0, 0, 0}	{0.0671, 0.9329, 0, 0, 0, 0, 0}
33	{L, L, Z}	−0.5816	−0.5483	{0, 0.5351, 0.4649, 0, 0, 0, 0}	{0, 0, 0.1097, 0.8903, 0, 0, 0}
34	{L, L, M}	1.9325	0.2569	{0, 0, 0, 0, 0.7669, 0.2331}	{0, 0, 0, 0.9486, 0.0514, 0, 0}
36	{L, L, VL}	−0.1097	0.2129	{0, 0.0108, 0.9892, 0, 0, 0, 0}	{0, 0, 0, 0.9574, 0.0426, 0, 0}

data, this period will bring out significant negative disturbance to the consequent attributes, the temperature and level at the bottom of the fractionator, which is verified by the belief degrees assigned to the consequent attributes as shown in Table 4.

Step 3: Test the newly constructed BRB with a feedforward compensation strategy.

In order to verify this expert system built above, a simulated DCU was constructed on the UniSim<sup>TM</sup> Operations Suite platform, which is customized to exactly replicate the actual plant and operate effectively in real-time.

Fig. 4 shows the different operation statuses of this DCU identified by the proposed BRB. The inflection points of the bold

blue polyline in the second sub-figure indicate the change of different operation statuses, which are coincident with the real operations.

For testing the newly constructed BRB with a feedforward compensation strategy, operation data in 5 day of this simulated DCU were used to complete the iterative learning strategy. With the help of the BRB obtained here, the relationship between different operation statuses and the temperature or level at the bottom of the fractionator can be easily found out. Then taking the upper fresh feed in Fig. 3 as the manipulated variable, the different operation statuses obtained in the BRB proposed above as the disturbance variables, the temperature and level at the

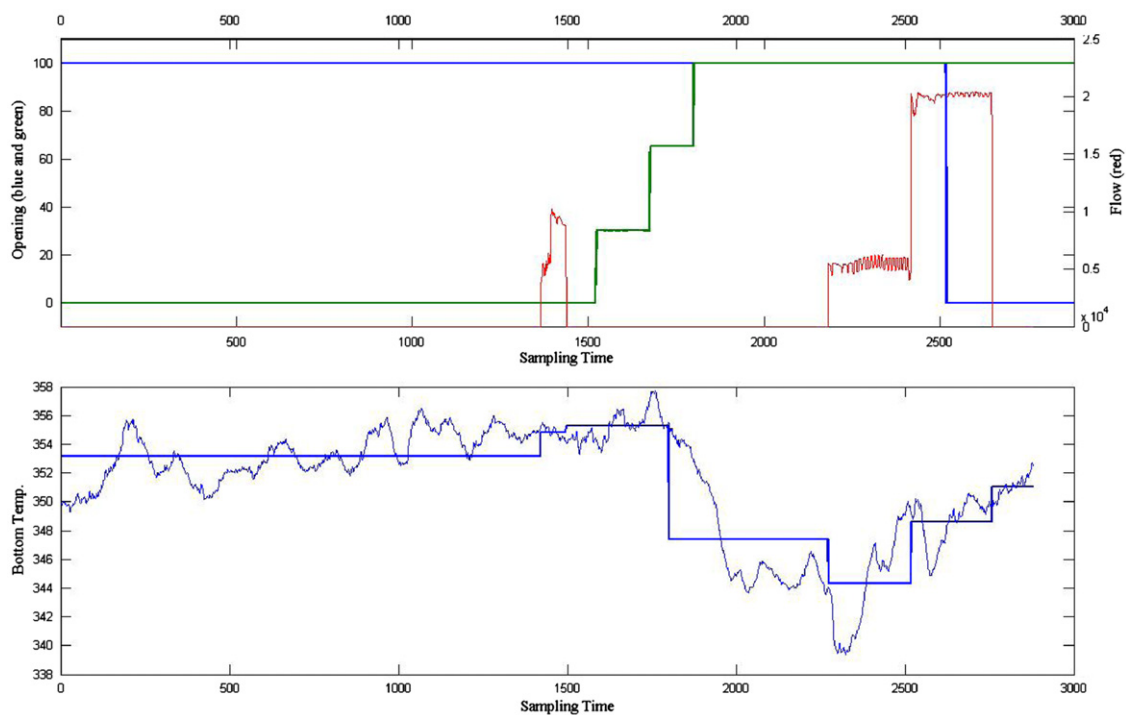


Fig. 4. The operation statuses of DCU identified by the proposed BRB.

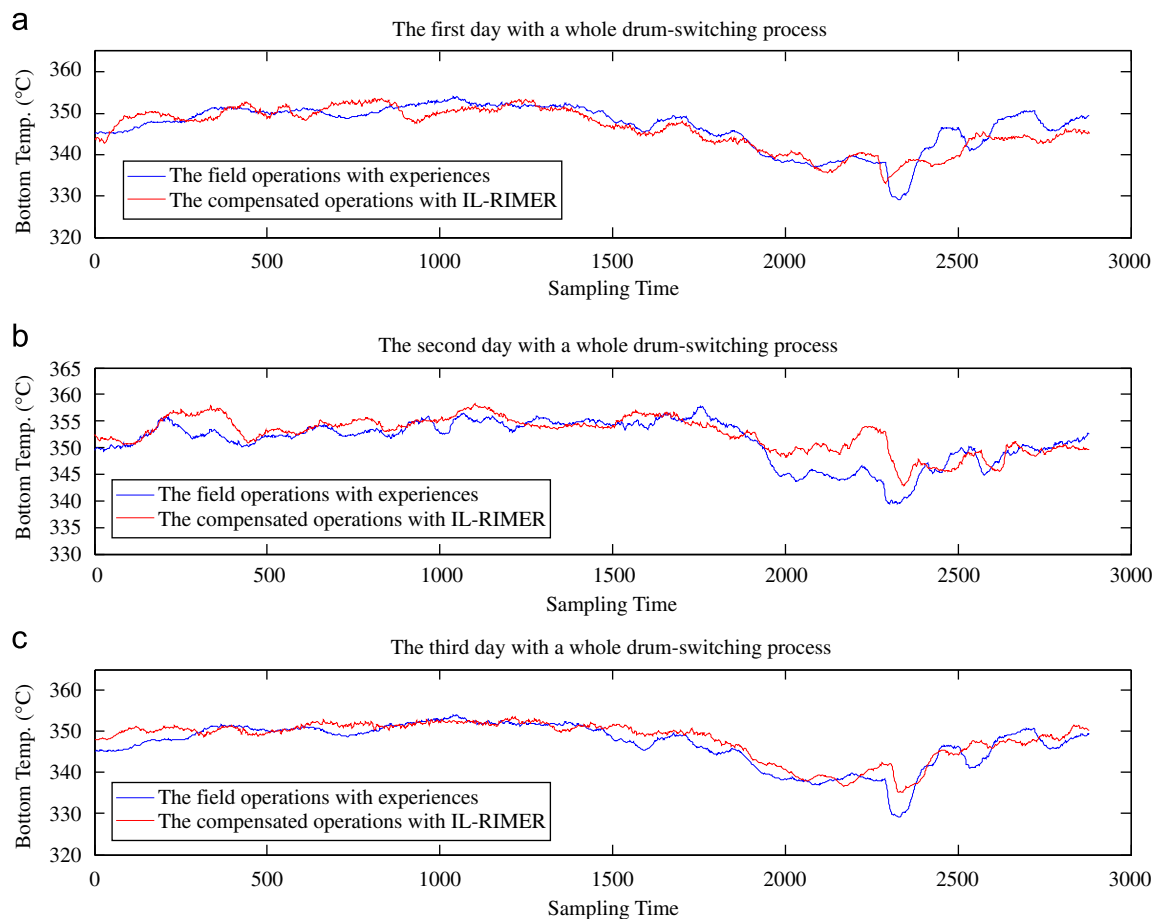


Fig. 5. The bottom temperature of the fractionator with and without the compensated operations using IL-RIMER methodology.



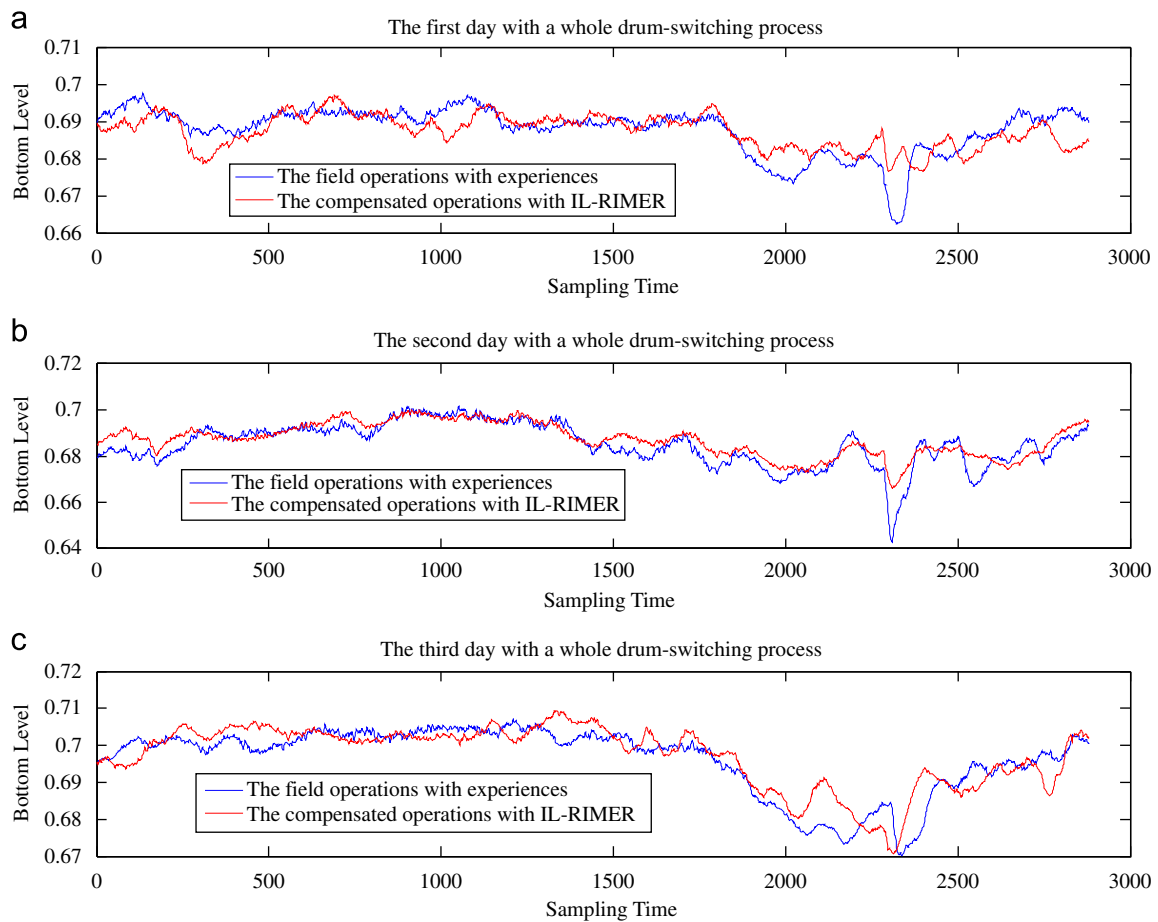


Fig. 6. The bottom level of the fractionator with and without the compensated operations using IL-RIMER methodology.

Table 5

The STD of one day's bottom temperature of the fractionators.

	<i>a</i>	<i>b</i>	<i>c</i>
The experienced operation	5.2061	4.0144	5.2061
The IL-RIMER strategy	4.9707	3.2140	4.5311

bottom of the fractionator as the controlled variables, a feedforward compensation strategy is incorporated into this expert system and implemented to smooth the operating process while the drum-switching.

Fig. 5 shows the temperature at the bottom of the fractionator in 3 day. It is observed from Fig. 5 that the performance with this feedforward compensation of the IL-RIMER proposed above is much better than the one without it. When the operation status of DCU changes, a certain compensation strategy is implemented immediately according to the relationship modeled by the newly constructed BRB. As a result, the great disturbance to the material and heat of the fractionator brought about by the drum-switching operation is eliminated by about 4 °C. Fig. 6 shows that it also works well in the performance of the bottom level of the fractionator. It should also be noticed that the outputs in Fig. 5 do not use the same units with the ones in the BRB dealt with above (Fig. 5 use the real units of the temperature and level at the bottom of the fractionator, while the BRB use the percent of the differences). So does Fig. 6.

The standard deviation (STD) of one day's bottom temperature of the fractionator with and without the compensated operation using IL-RIMER methodology proposed in our paper is shown in Table 5. A closer look at the standard deviation of the temperature

Table 6

The STD of one day's bottom temperature of the fractionators just during the drum-switching period.

	<i>a</i>	<i>b</i>	<i>c</i>
The experienced operation	4.1916	2.2485	4.1916
The IL-RIMER strategy	1.7648	2.9128	2.7108

Table 7

The STD of one day's bottom temperature of the fractionators.

	<i>a</i>	<i>b</i>	<i>c</i>
The experienced operation	0.0062	0.0092	0.0091
The IL-RIMER strategy	0.0045	0.0072	0.0083

Table 8

The STD of one day's bottom temperature of the fractionators just during the drum-switching period.

	<i>a</i>	<i>b</i>	<i>c</i>
The experienced operation	0.0054	0.0100	0.0054
The IL-RIMER strategy	0.0026	0.0049	0.0061

just during the drum-switching period in Table 6 reveals that the expert system developed in this paper is much capable of eliminating the great disturbance brought about by the drum-switching operations.

Tables 7 and 8 show the statistical characteristics of the bottom level control performance of the the fractionator. It is consistent with the good behavior in bottom temperature control described above.

## 5. Conclusions

In this paper, a novel updating algorithm for RIMER model is proposed based on iterative learning strategy for DCU. Daily DCU operations under different conditions are modeled by a BRB, which is then updated using iterative learning methodology, based on a novel statistical utility for every belief rule. Compared with the other learning algorithms, our methodology can lead to a more optimal compact final BRB. Obtaining the congruent relationship between the different operation statuses and the disturbance to the fractionator modeled by the optimized BRB, a feedforward compensation strategy is introduced to eliminate the disturbance caused by the drum-switching operations. A DCU operation expert system is also developed using the methodology proposed above based on the field data from a real oil refinery. The simulation results with a better performance on the UniSim<sup>TM</sup> Operations Suite platform demonstrate the effectiveness and efficiency of this approach proposed in our paper.

## Acknowledgments

Financial support from the National Basic Research Program of China (No. 2012CB720500) and National Natural Science Foundation of China (No. 60974008) is gratefully acknowledged.

## Appendix A

In most cases, the input of a quantitative antecedent attribute is given by numerical values. Then an input value  $x_i(t)$  is represented using the following equivalent expectation (Zhou et al. 2010):

$$S(x_i(t)) = \{(A_{ij}, a_{ij}(x_i(t))), \quad i = 1, 2, \dots, T_k, \quad j = 1, 2, \dots, J_i\} \quad (A.1)$$

where  $a_{ij}(x_i(t))$  can be calculated by

$$a_{ij}(x_i(t)) = \frac{A_{ij+1} - x_i(t)}{A_{ij+1} - A_{ij}} \quad \text{if } A_{ij} \leq x_i(t) \leq A_{ij+1}, \quad j = 1, 2, \dots, J_i - 1 \quad (A.2)$$

$$a_{ij+1}(x_i(t)) = 1 - a_{ij}(x_i(t)) \quad \text{if } A_{ij} \leq x_i(t) \leq A_{ij+1}, \quad j = 1, 2, \dots, J_i - 1 \quad (A.3)$$

$$a_{i,s}(x_i(t)) = 0, \quad s = 1, 2, \dots, \quad J_i, s \neq j+1 \quad (A.4)$$

It should be noted that there are some remarks as follows:

- (1) A large value  $A_{ij+1}$  is supposed here to be preferred over a small value  $A_{ij}$ .  $A_{ij}$  and  $A_{i,1}$  are assumed to be the largest and smallest feasible values, respectively.
- (2) When the quantitative antecedent attribute,  $x_i(t)$ , comes as a random variable with some probabilities, the corresponding rule-based information transformation technique has also been studied in Yang's paper (Yang, 2001).

## Appendix B

Based on the definition of utility and the neuron's "significance" concept of RBF, Zhou et al. (2010) gave a statistical utility's definition for a belief rule.

Suppose that the quantitative antecedent attribute,  $x_i(t)$  ( $i = 1, 2, \dots, T_k$ ), changes in the interval  $[a_i, b_i]$  and obeys some distribution with the sampling density function  $p_i(x_i(t))$ . Define

$$E_k \approx \theta_k \prod_{i=1}^{T_k} I_i^k(A_i^k) \quad (B.1)$$

where  $I_i^k(A_i^k) = \int_{a_i}^{b_i} (a_i^k(x_i(t)))^{\bar{\sigma}_{ki}} p_i(x_i(t)) dx_i(t)$ .  $A_i^k \in \{A_{ij}, i = 1, 2, \dots, T_k, j = 1, 2, \dots, J_i\}$  denotes the referential value of the  $i$ th antecedent attribute in the  $k$ th belief rule. Then with the definition of  $E_k(A_k) \triangleq E_k$ , where  $A_k = [A_1^k, \dots, A_{T_k}^k]^T$  represents the referential vector of the antecedent attributes in the  $k$ th belief rule, we get

$$\lim_{n \rightarrow +\infty} U(k) \approx \frac{E_k(A_k) \sum_{j=1}^N u(D_j) \beta_j}{\sum_{l=1}^L E_l(A_l)} \quad (B.2)$$

where, Eq. (B.2) is considered as the statistical utility of the  $k$ th belief rule. It was also investigated under the assumption of the uniform distribution in Zhou's paper (Zhou et al., 2010).

## References

- Anthony, D. B., Kruse, C. J., & Ewy, G. L. (1996). Advances in coking technology. In Proc. AIChE 1996 Spring National Meeting, New Orleans.
- Binaghi, E., & Madella, P. (1999). Fuzzy Dempster-Shafer reasoning for rule-based classifiers. *International Journal of Intelligent Systems*, 14(6), 559–583.
- Duan, X.-G., Yang, C.-H., Li, H.-X., Gui, W.-H., & Deng, H. (2008). Hybrid expert system for raw materials blending. *Control Engineering Practice*, 16(11), 1364–1371.
- Dymova, L., Sevastianov, P., & Bartosiewicz, P. (2010). A new approach to the rule-base evidential reasoning: Stock trading expert system application. *Expert Systems with Applications*, 37(8), 5564–5576.
- Elliott, J. D. (2003). Optimize coker operations. *Hydrocarbon Processing*, 82(9), 85–92.
- Ellis, P. J., Paul, C. A., Session, T. (1998). Tutorial: delayed coking fundamentals. In Proc. AIChE 1998 Spring National Meeting, New Orleans.
- Friedman, Y. Z. (2005). Why coker APC applications are tough. *Hydrocarbon Processing*, 84(12), 93–98.
- Giarratano, J. C., & Riley, G. (1989). *Expert Systems: Principles and Programming*. Pacific Grove, CA, USA: Brooks/Cole Publishing Co.
- Haseloff, V., Friedman, Y. Z., & Goodhart, S. G. (2007). Implementing coker advanced process control. *Hydrocarbon Processing*, 86(6), 99–103.
- Jackson, P. (1998). *Introduction to Expert Systems*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Jensen, F. V. (1996). *An Introduction to Bayesian Networks*. London: UCL Press Limited.
- Kong, G., Xu, D. L., Liu, X., & Yang, J. B. (2009). Applying a belief rule-base inference methodology to a guideline-based clinical decision support system. *Expert Systems*, 26(5), 391–408.
- Liao, S. H. (2005). Expert system methodologies and applications—A decade review from 1995 to 2004. *Expert Systems with Applications*, 28(1), 93–103.
- Liu, J., Yang, J. B., Wang, J., & Sii, H. S. (2005). Engineering system safety analysis and synthesis using the fuzzy rule-based evidential reasoning approach. *Quality and Reliability Engineering International*, 21(4), 387–411.
- Pawlak, Z. (1991). *Rough Sets: Theoretical Aspects of Reasoning about Data*. Springer.
- Rodríguez-Reinoso, F., Santana, P., Palazon, E. R., Diez, M. A., & Marsh, H. (1998). Delayed coking: industrial and laboratory aspects. *Carbon*, 36(1–2), 105–116.
- Valyavin, G. G., Khukhrin, E. A., & Valyavin, K. G. (2007). The place of delayed coking in modern oil refineries. *Chemistry and Technology of Fuels and Oils*, 43(3), 191–196.
- Wang, Y.-M., Yang, J.-B., Xu, D.-L., & Chin, K.-S. (2006). The evidential reasoning approach for multiple attribute decision analysis using interval belief degrees. *European Journal of Operational Research*, 175(1), 35–66.
- Xu, D. L., Liu, J., Yang, J. B., Liu, G. P., Wang, J., Jenkinson, I., et al. (2007). Inference and learning methodology of belief-rule-based expert system for pipeline leak detection. *Expert Systems with Applications*, 32(1), 103–113.
- Yang, J. B. (2001). Rule and utility based evidential reasoning approach for multiattribute decision analysis under uncertainties. *European Journal of Operational Research*, 131(1), 31–61.
- Yang, J. B., Liu, J., Wang, J., Sii, H. S., & Wang, H. W. (2006). Belief rule-base inference methodology using the evidential reasoning approach-RIMER. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, 36(2), 266–285.
- Yang, J. B., Liu, J., Xu, D. L., Wang, J., & Wang, H. (2007). Optimization models for training belief-rule-based systems. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, 37(4), 569–585.
- Yu, X., Huang, D., Jiang, Y., & Jin, Y. (2011). Iterative learning belief rule-base inference methodology using evidential reasoning for delayed coking unit. In Proc. 2011 International Symposium on Advanced Control of Industrial Processes (ADCONIP). May 23–26, Hangzhou, China.
- Yu, X., Wei, Y., Huang, D., Jiang, Y., Liu, B., & Jin, Y. (2011). Intelligent switching expert system for delayed coking unit based on iterative learning strategy. *Expert Systems with Applications*, 38(7), 9023–9029.

- Zhao, Y., Chu, J., Su, H., & Huang, B. (2010). Multi-step prediction error approach for controller performance monitoring. *Control Engineering Practice*, 18(1), 1–12.
- Zhou, M., Wang, S., & Jin, X. (2009). Application of advanced process control in delayed coking units. *Control and Instruments in Chemical Industry (Chinese)*, 36(1), 79–82.
- Zhou, Z. J., Hu, C. H., Yang, J. B., Xu, D. L., Chen, M. Y., & Zhou, D. H. (2010). A sequential learning algorithm for online constructing belief-rule-based systems. *Expert Systems with Applications*, 37(2), 1790–1799.
- Zhou, Z. J., Hu, C. H., Yang, J. B., Xu, D. L., & Zhou, D. H. (2009). Online updating belief rule based system for pipeline leak detection under expert intervention. *Expert Systems with Applications*, 36(4), 7700–7709.
- Zhou, Z. J., Hu, C. H., Yang, J. B., Xu, D. L., & Zhou, D. H. (2011). Online updating belief-rule-base using the RIMER approach. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, 41(6), 1225–1243.